UNITED STATES DISTRICT COURT
DISTRICT OF NEW JERSEY

OANDA Corporation, *Plaintiff* v. GAIN Capital Holdings, Inc.; GAIN Capital Group, LLC. *Defendants.*	Civil Action No. 2:20-cv-5784 **JURY TRIAL DEMANDED**

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff OANDA Corporation ("OANDA" or "Plaintiff") complains and alleges as follows against defendants GAIN Capital Holdings, Inc. and GAIN Capital Group, LLC (dba FOREX.com) (collectively, "GAIN" or "Defendants"). These allegations are made based on personal knowledge as to OANDA with respect to its own actions, and upon information and belief as to all other matters.

THE PARTIES

1. Plaintiff OANDA is a Delaware corporation, having offices at 1441 Broadway 6th Floor, Suite 6027, New York, New York 10018.

2. OANDA is a global leader in online multi-asset trading services and currency data and analytics.

3. OANDA is the owner, by assignment, of U.S. Patents No. 7,146,336 (the ’336 Patent) and 8,392,311 (the ’311 Patent), attached as Exhibit A and Exhibit B, respectively.

4. On information and belief, Defendant GAIN Capital Holdings, Inc. is a Delaware corporation, with its global headquarters at 135 U.S. Highway 202/206, Bedminster, New Jersey 07921.

5. On information and belief, Defendant GAIN Capital Group, LLC is a Delaware limited liability company, with its global headquarters at 135 U.S. Highway 202/206, Bedminster, New Jersey 07921.

6. On information and belief, GAIN Capital Group, LLC owns and operates the website https://forex.com, among others, which provides foreign exchange (also known as “forex” or “FX”) trading and brokerage services, including an online trading platform, and which infringes OANDA’s patent rights as described herein.

7. On information and belief, GAIN Capital Holdings, Inc. owns and operates the website https://www.gaincapital.com and uses the services of GAIN Capital Group, LLC, including the application programming interfaces (APIs) provided by https://forex.com, to operate automated trading platform(s).

JURISDICTION AND VENUE

8. This is an action for patent infringement arising under 35 U.S.C. §1, *et seq*.

9. This Court has subject matter jurisdiction over this action pursuant to 28 U.S.C. §1331.

10. This Court has both general and specific personal jurisdiction over Defendants. Each of the Defendants has sufficient minimum contacts within the State of New Jersey (including via Defendants locating their worldwide headquarters here, as well as sales of Defendants' products and services in New Jersey), pursuant to due process and/or the New Jersey Long Arm Statute, because Defendants purposefully availed themselves of the privileges of conducting business in New Jersey, because Defendants regularly conduct and solicit business within New Jersey, and because Plaintiff's causes of action arise directly from Defendants' business contacts and other activities in the State of New Jersey.

11. Venue is proper in this District pursuant to 28 U.S.C. §1400(b) because Defendants have committed acts of infringement in this District, including at least those acts complained of herein, and have regular and established places of business in New Jersey.

//

//

THE SCHOLARSHIP & INVENTIONS OF OANDA

12. OANDA, a market leader in currency data and currency trading, was founded in 1996 by Dr. Michael Stumm and Dr. Richard Olsen.

13. Dr. Michael Stumm is a teacher, researcher, entrepreneur, and executive. As a professor in the University of Toronto's Department of Electrical and Computer Engineering, he has published over 100 papers in top-tier conference proceedings and scientific journals. Dr. Stumm is the inventor or co-inventor on fifteen U.S. patents related to market and currency trading and telecommunications networks.

14. Dr. Richard Olsen is an academic, entrepreneur, and founder of Olsen Ltd., a leading econometric research and development firm. Dr. Olsen is the lead author of the textbook, *An Introduction to High-Frequency Finance* (Academic Press, 2001), which provides the first and only source of unified information about high-frequency data, with a particular emphasis on foreign exchange markets. Dr. Olsen is the inventor or co-inventor on nine U.S. patents related to market and currency trading.

15. Dr. Stumm and Dr. Olsen's vision in founding OANDA was to make currency exchange rate information more accessible to a broader audience. By the mid-1990s, even with the advent of the internet, there were no centralized, transparent exchanges for currencies that retail investors could access, as there were

for stocks. That lack of transparency allowed large banks and currency dealers to maintain large "spreads" (the price difference between where a trader may purchase or sell an underlying asset) for retail customers.

16. In 1996, OANDA launched the world's largest and most accurate database of currency prices, employing Dr. Stumm's technological expertise and Dr. Olsen's expertise in currency markets. OANDA soon became the gold standard for forex prices and interbank exchange rates online, relied upon by major corporations, auditing firms, and individual traders alike.

17. In 2000, Dr. Stumm and Dr. Olsen had the idea to create an online automated trading platform, through which they could offer individual investors the more favorable rates banks used to trade currency among themselves. Prior to that, while OANDA had made accurate exchange rates more available to the public, banks and currency dealers continued to charge consumers large spreads when trading currency. While some online trading platforms existed at that time, they suffered from a number of deficiencies. In the then-existing online currency market, for example, a trade went through three steps from initiation to execution: (1) the trader specified to a dealer the "currency pair" (a price quote of the exchange rate for two different currencies traded in forex markets) and the amount that the trader would want to trade (without specifying whether he or she would like to buy or sell); (2) the dealer specified to the trader both a bid and an ask price and gave the trader

several seconds to respond, in order to protect against price fluctuations (the dealer not knowing whether the trader would buy, sell, or reject the offer); and (3) the trader either rejected the offer or specified whether the trader was buying or selling (with his or her response having to occur within a timeframe of a few seconds).

18. This "three-way handshake" created problems, including that potential internet delays might not allow the trader to respond within the few-seconds window, and that corporate firewalls restricted the flow of information outside the corporate network.

19. Dr. Stumm and Dr. Olsen used their combined expertise to invent systems and methods for online currency trading that overcame these and other deficiencies of then-existing online currency trading. Their inventions, for example, allowed for execution of online currency transactions with only two communications, instead of three, eliminated the previous problems with timing lags, and built in automated protections against price fluctuations. Dr. Stumm and Dr. Olsen were granted patent protection on these novel systems and methods, including the '336 and '311 Patents, among others.

20. These inventions were embodied in OANDA's pioneering currency trading platform, fxTrade, which launched in 2001. The first fully automated online currency trading platform, fxTrade, among other features, monitored market exchange rates, offered immediate price quotes (with a much smaller spread than

offered by banks), executed trades instantaneously, and prevented clients from risking too much money through automatic stop-loss orders. It also allowed customers to trade with deposits as small as one dollar, while charging interest on leveraged trades on a second-by-second basis.

21. As stated in the provisional patent application that forms the basis for the ’336 and ’311 Patents, and which is also the manuscript for the textbook *An Introduction to High-Frequency Finance*:

> As the archetype of financial markets, the foreign exchange market is the largest financial market worldwide. It involves dealers in different geographic locations, time zones, and working hours who have different time horizons, home currencies, information access, transaction costs, and other institutional constraints.

Ex. C, US Provisional Patent App. 60/274,174, p.15; *see also* Olsen, *et al*., *An Introduction to High-Frequency Finance*, Preface, p. xxi.

22. Additionally, the provisional patent discussed the skepticism amongst academics and others at the time to the innovations and inventions of the patents-in-suit:

> Recently, the skepticism among academics to the possibility of developing profitable trading models has decreased with the publication of many papers that document profitable trading strategies in financial markets, even when including transaction costs.
>
> …

> The purpose of this chapter is not to provide *ready-to-use* trading strategies, but to give a description of the main ingredients needed for any real-time trading model to be usable for actual trading on financial markets. Any reasonable trading strategy is composed of a set of tools that provides trading recommendations within a capital management system.
>
> …
>
> To construct successful trading strategies is not an easy task and many possible mistakes must be avoided during the different development phases of new models. We shall describe here some of the main traps in which new system designers generally fall and provide some ideas as to how to construct more robust trading strategies.

Ex. C, US Provisional Patent App. 60/274,174, pp. 315-317; *see also* Olsen, *et al.*, *An Introduction to High-Frequency Finance*, Ch. 11 – Trading Models, pp. 295-297 (Academic Press, 2001) (emphasis in original).

THE '336 PATENT

23. On December 5, 2006, the United States Patent and Trademark Office duly and legally issued United States Patent No. 7,146,336, entitled "Currency Trading System, Methods, and Software." A true and correct copy of the '336 Patent is attached as Exhibit A.

24. The '336 Patent and its claims are entitled to, at least, the benefit of the filing date of its provisional patent application, 60/274,174, which was filed on March 8, 2001. A true and correct copy of the provisional patent application 60/274,174 is attached as Exhibit C.

25. OANDA is the owner, by assignment, of the ’336 Patent.

26. The ’336 Patent teaches, among other things:

> In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. In another aspect, the present invention comprises methods for trading currency over a computer network. In another aspect, the present invention comprises software for currency trading over a computer network.
>
> U.S. Patent 7,146,336, Abstract.

27. The claims of the ’336 Patent are, and are presumed to be, valid, patent-eligible and enforceable.

28. The claims of the ’336 Patent are not directed to an abstract idea or concept. Rather, they are directed to specific implementations of computerized trading systems and interfaces for trading currencies (e.g., foreign exchange or “forex”).

//

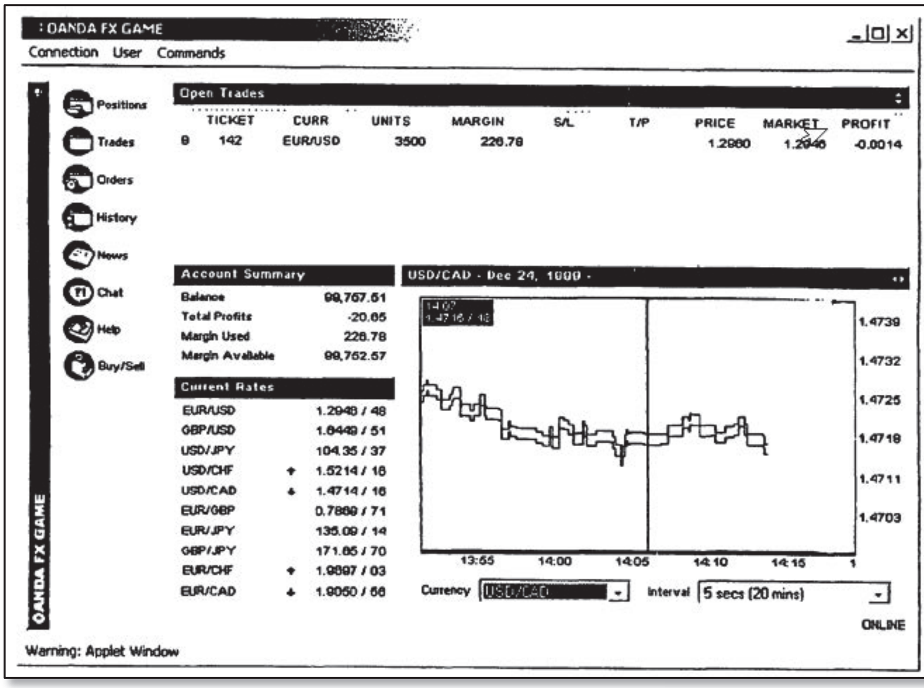
//

//

//

//

//



('336 Patent, Fig. 2)

29. Each of the claims of the '336 Patent is inventive over the prior art, including but not limited to independent claims 1-5, 7, and 11, and dependent claims 6 and 8-10. Specifically, the claims are non-abstract and embody an inventive concept at least because their claimed elements, combinations of elements, and the interactions between those elements was not well-understood, routine, and conventional at the time of the application.

30. The claims of the '336 Patent claim technological improvements on the prior art, including but not limited to the Piskiel, Heinzle, Usher, Selleck, Szoc, Tsagarakis, Lange, and Rossman references listed on the face of the patent.

31. The '336 Patent's improvements over traditional on-line currency trading are discussed within the specification of the patent itself, including the

limitations of the prior art's traditional systems. *See* '336 Patent, Col. 1:18-43. These improvements and the claims of the '336 Patent, as well as the use of the claim elements to accomplish the goals of the invention, were inventive, unconventional, and not well known as of the priority date of the '336 Patent.

32. The '336 Patent's benefits include, but are not limited to, that its claimed teachings lessened or eliminated the problems of, among other things, paying and collecting interest, and executing stored orders, as discussed in the specification. The claimed inventions are able to overcome these at least because, by having a computerized interest rate manager calculate, pay out, and collect interest on a tick-by-tick basis, or by having a trade manager check stored orders, more accurate and comprehensive trades and payments of interest (e.g., "rollover") can be accomplished than is possible with human beings involved. In this respect, the claimed systems make it both quantitatively and qualitatively different from what can be accomplished by humans, teams of humans, or the prior art systems.

33. For example, the computerized interest rate managers claimed by the '336 Patent enable accurate payment and collection of interest, even on small positions, which would have been impossible and unprofitable to calculate using humans. Similarly, the computerized trade manager claimed by the '336 Patent makes tracking and execution of varied and complex stored orders (e.g., stop loss, take profit, and limit) possible, in real time, with an accuracy that would be

impossible for humans to attain, not least because – due to the extremely high frequency of the price movements – it would be physically impossible for humans to simultaneously monitor the price movements, receive and input orders from traders, compare the current prices against the stored orders, and report the execution back to the traders.

34. Moreover, as discussed and claimed by the ’336 Patent in Claim 11, by implementing the disclosed trading system, brokers can automatically catch traders or accounts who are operating outside their margin limits and automatically liquidate their holdings in real time. This protects the broker from the credit risk of positions held in accounts on margin that are overrunning their margin limits.

35. As discussed in the specification of the ’336 Patent, this approach to trading currencies is unconventional and a sharp departure from the traditional methods described in the specification and in the prior art.

36. Each of the claims of the ’336 Patent are patentably distinct from each other and offer individualized technological improvements and differing inventive concepts, which vary each claim from the other claims. None of the claims of the ’336 Patent are duplicative or representative of the other claims because the technological limitations of each claim differ from the others.

37. For example, Claim 6, which depends from Claim 5, adds the limitation:

> *wherein said pricing engine is further operable to compute currency exchange rates based on positions held by said system.*

This limitation is not present in Claim 5 and adds the specific technological requirements recited therein, which improve on the prior art systems and methods; in this case, a way for the system operator to take into account its own positions to advantageously set exchange rates.

38. For a further example, Claim 10, which depends from Claim 8, adds the limitation:

> *at least one of said one or more trading models comprises: (a) a price collector component; (b) a price filter component; (c) a price database component; (d) a gearing calculator component; (e) a deal acceptor component; (f) an opportunity catcher component; and (g) a book-keeper component.*

This limitation is not present in Claim 8 and adds the specific technological requirements recited therein, which improve on the prior art systems and methods; in this case, that the system include subsystems for, among other things, monitoring and acting on prices, gearing, deals, and opportunities.

39. Regarding independent claims 1-5, 7, and 11, it was not well-understood, routine, and conventional at the time of the application to trade currencies over a computer network using a trading client system, as specified in those claims, at least because of the deficiencies of the prior art systems described in the specification and in the prosecution history.

40. Regarding dependent claims 6 and 8-10, each of these dependent claims add the additional specific limitations recited therein to the claims from which they depend, further reinforcing and adding to the specific technological requirements of the claimed methods.

41. Further, the claims of the ’336 Patent claim specific technological improvements on pre-existing technological systems and methods, including the traditional on-line currency markets discussed in the patent, as well as the prior art patents and other references cited on the face of the patent (e.g., Piskiel, Heinzle, Usher, Selleck, Szoc, Tsagarakis, Lange, and Rossman.)

42. Regarding each of the independent claims, it was further not well understood, routine, and conventional to combine the elements of these independent claims with the elements of their respective allowed dependent claims.

THE ’311 PATENT

43. On March 5, 2013, the United States Patent and Trademark Office duly and legally issued United States Patent No. 8,392,311, entitled “Currency Trading System, Methods, and Software.” A true and correct copy of U.S. Patent No ’311 is attached as Exhibit B.

44. The ’311 Patent and its claims are entitled to, at least, the benefit of the filing date of its provisional patent application, 60/274,174, which was filed on March 8, 2001. Exhibit C.

45. OANDA is the owner, by assignment, of the '311 Patent.

46. The '311 Patent teaches, among other things:

> In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. In another aspect, the present invention comprises methods for trading currency over a computer network. In another aspect, the present invention comprises software for currency trading over a computer network.

U.S. Patent 8,392,311, Abstract.

47. The claims of the '311 Patent are, and are presumed to be, valid, patent-eligible and enforceable.

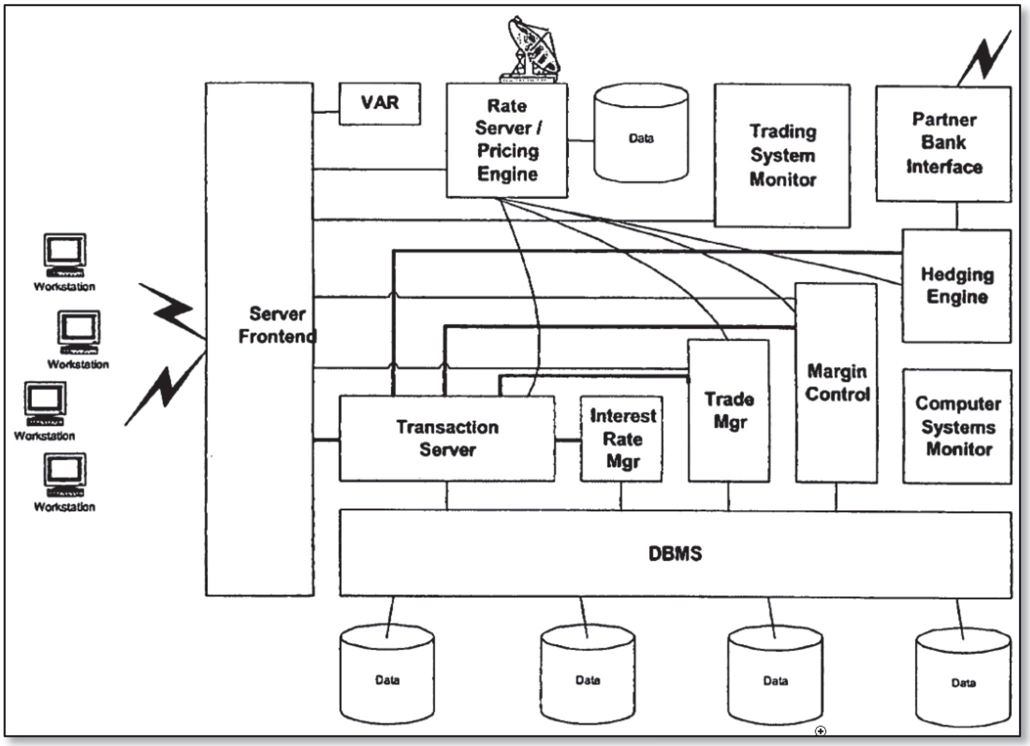
48. The claims of the '311 Patent are not directed to an abstract idea or concept. Rather, they are directed to specific implementations of computerized trading systems and interfaces for trading currencies (e.g., foreign exchange or "FX").

49. Each of the claims of the '311 Patent is inventive over the prior art, including but not limited to independent claims 1 and 7 and dependent claims 2-6. Specifically, the claims are non-abstract and embody an inventive concept at least because their claimed elements, combinations of elements, and the interactions between those elements, was not well-understood, routine, and conventional at the time of the application.

50. The claims of the ’311 Patent claim technological improvements on the prior art, including but not limited to the Potter, Buchalter, Selleck, Szoc, Cagan, McDermott, and Turton references listed on the face of the patent.

51. The ’311 Patent’s improvements over traditional on-line currency trading are discussed within the specification of the patent itself, including the limitations of the prior art’s traditional methods. *See* ’311 Patent, Col. 1:20-39. These improvements and the claims of the ’311 Patent, as well as the use of the elements of the claims to accomplish the goals of the invention, were inventive, unconventional, and not well known as of the priority date of the ’311 Patent.

52. The ’311 Patent’s benefits include, but are not limited to, that its claimed teachings lessened or eliminated the problems of the three-way handshake, as discussed in the specification. The claimed inventions were able to overcome the problems of the three-way handshake at least because, by cutting out human beings from certain parts of the process and having a client-server computer system transmit constantly updated exchange rates and prices, and constantly receive orders, the customers (traders) using the system will have more accurate pricing data due to decreases in the latency of the system. Additionally, by having the system constantly receive orders from traders, the incidence of orders refused because the price had changed while the trader was waiting for a response from a human is greatly reduced.



('311 Patent, Fig. 3)

53. As discussed in the specification, this approach to trading currencies is unconventional and a sharp departure from the traditional method described as the three-way handshake in the specification.

54. Each of the claims of the '311 Patent are patentably distinct from each other and offer individualized technological improvements, and differing inventive concepts, which vary each claim from the other claims. None of the claims of the '311 Patent are duplicative or representative of the other claims because the technological limitations of each claim differ from the others.

55. For example, Claim 2, which depends from Claim 1, adds the limitation:

> *wherein the requested trade price is derived from a respective one of the first price or second price of the received current exchange rate and a user input limit value defining a maximum acceptable difference between the respective one of the first price or second price of the received current exchange rate received at the trading client system and the respective one of the first price or second price of the corresponding current exchange rate determined at the trading client system at which the trade can be effected.*

This limitation is not present in Claim 1 and adds the specific technological requirements recited therein, which improve on the prior art systems and methods; in this case, a specific method for deriving the requested trade price at the trading client system.

56. For a further example, Claim 4, which depends from Claim 2, adds the limitation:

> *displaying to the user a set of input fields to define a desired trade, the input fields including an identification of the pair of currencies the user desires to trade, the amount of the currencies desired to be traded, the selected first price or second price of the current exchange rate received at the trading client system and a limit value, and where the input fields to identify the pair of currencies and the first price or second price are populated with appropriate values determined from the user's selection of the one of the first price or second price;.*

This limitation is not present in Claim 1 or Claim 2 and adds the specific technological requirements recited therein, which improve on the prior art systems and methods; in this case, a particular user interface that includes input fields to define a desired trade, including an identification of the pair of currencies the user desires to trade.

57. Regarding independent claims 1 and 7, it was not well-understood, routine, and conventional at the time of the application to trade currencies over a computer network using a trading client system, as specified in claims 1 and 7 of the ’311 Patent, at least because of the deficiencies of the prior art systems described in the specification and in the prosecution history.

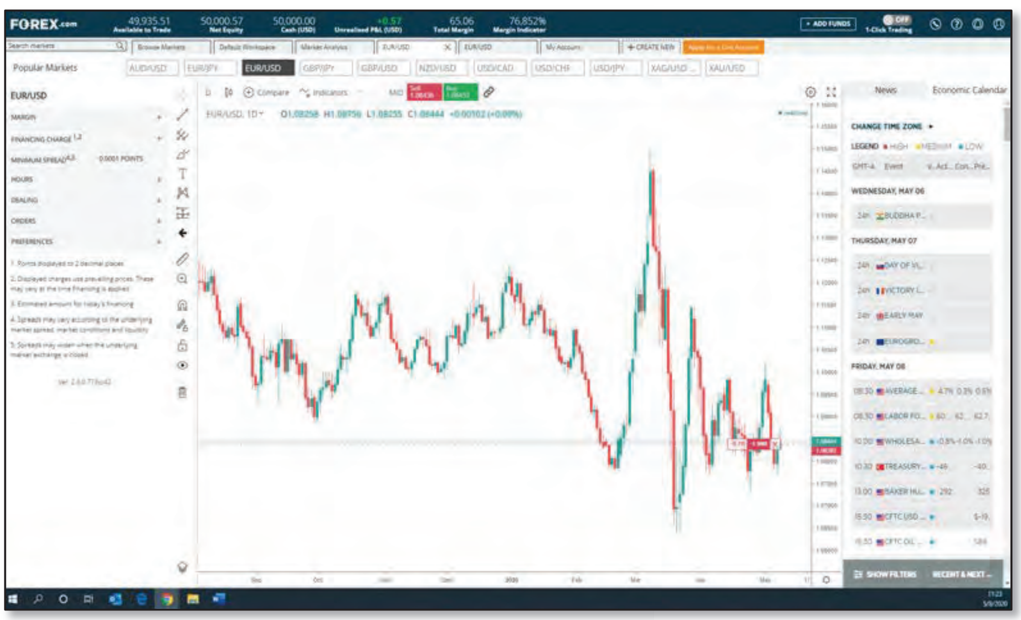
58. Regarding dependent claims 2-6, each of these dependent claims add the additional specific limitations recited therein to the claims from which they depend, further reinforcing and adding to the specific technological requirements of the claimed methods.

59. Further, the claims of the ’311 Patent claim specific technological improvements on pre-existing technological systems and methods, including the traditional on-line currency markets discussed in the patent, as well as the prior art patents and other references cited on the face of the patent (e.g., Potter et al., Buchalter, Selleck, Szoc et al., Cagan, McDermott, and Turton.)

60. Regarding each of the independent claims, it was further not well understood, routine, and conventional to combine the elements of said independent claims with the elements of their respective allowed dependent claims.

DEFENDANTS' INFRINGEMENT

61. Defendant Gain Capital Group, LLC operates the website https://forex.com, where it, as shown in the screenshot below, provides an online currency trading platform. Defendant also offers desktop and mobile apps, API access, white-label systems (Defendants' systems being offered under another company's label), and other technological mechanisms for foreign exchange trading. Defendant GAIN Capital Holdings, Inc. uses GAIN Capital Group, LLC's platform, including the APIs, to operate automated infringing trading systems.



62. Defendants' platforms and systems practice each and every limitation of claims 1-11 of the '336 Patent and claims 1-7 of the '311 Patent.

63. Plaintiff identifies, on the claim charts attached, those elements of Defendants' platforms and systems that Plaintiff believes at this time practice each of the individual limitations of the '336 and '311 Patents.

64. Plaintiff's identification of Defendants' infringement is preliminary, and Plaintiff expects to identify additional acts of infringement, and additional mechanisms by which Defendants platforms and systems infringe, upon a reasonable opportunity for discovery.

COUNT I –INFRINGEMENT OF '336 PATENT

65. The foregoing numbered paragraphs are incorporated by reference into this section.

66. Defendants have infringed one or more claims of the '336 Patent by making, using, selling, offering for sale, or selling products and/or services that meet each of the limitations of one or more claims of the '336 Patent. More specifically, Defendant GAIN Capital Group, LLC has made, used, sold, and offered for sale infringing instrumentalities at https://forex.com, and GAIN Capital Holdings, Inc. has used those infringing instrumentalities, including the application programming interfaces ("APIs"), to operate automated infringing trading systems.

67. Defendants have continued to operate their online trading platforms and systems in an infringing manner, despite being notified of its infringement by Plaintiff on at least one occasion by letter specifically referencing the '336 Patent.

68. Defendants infringe each and every limitation of at least Claim 1 of the '336 Patent. *See* Claim Chart for '336 attached as Exhibit D.

COUNT II –INFRINGEMENT OF ’311 PATENT

69. The foregoing numbered paragraphs are incorporated by reference into this section.

70. Defendants have infringed one or more claims of the ’311 Patent by making, using, selling, offering for sale, or selling products and/or services that meet each of the limitations of one or more claims of the ’311 Patent. More specifically, Defendant GAIN Capital Group, LLC has made, used, sold, and offered for sale infringing instrumentalities at https://forex.com, and GAIN Capital Holdings, Inc. has used those infringing instrumentalities, including the APIs, to operate automated infringing trading systems.

71. Defendants have continued to operate their online trading systems in an infringing manner, despite being notified of their infringement by Plaintiff on at least one occasion by letter specifically referencing the ’311 Patent.

72. Defendants infringe each and every limitation of at least Claim 1 of the ’311 Patent. *See* Claim Chart for ’311 attached as Exhibit E.

COUNT III – CONTRIBUTORY OR INDUCED INFRINGEMENT

73. The foregoing numbered paragraphs are incorporated by reference into this section.

74. On information and belief, Defendants use subcontractors, managers, agents, or other third parties (“Third-Party Infringers”) to operate or assist in the

management of its online trading systems, or to provide additional services in connection with its services, including by use of its platform's APIs. These subcontractors or other third parties infringe one or more of the claims of the patent in suit.

75. Defendants were aware of their infringement of the '336 and '311 Patents at least as early as October 25, 2018, when Plaintiff notified Defendants, by letter, of its infringement and yet Defendants continue to cause its Third-Party Infringers to operate or assist in the management of its online trading platforms on its behalf.

76. Defendants continued use of Third-Party Infringers to operate its online trading platforms on its behalf constitutes contributory and/or induced infringement.

COUNT IV – WILLFUL INFRINGEMENT

77. The foregoing numbered paragraphs are incorporated by reference into this section.

78. At least as early as October 25, 2018, Defendants have been aware of its infringement of OANDA's patents. On that date, OANDA notified Defendants, by letter, of its infringement and demanded that Defendants take a license or cease its infringement. Defendants declined to do so.

79. Defendants' continued infringement of the patent is willful given its knowledge of the '336 Patent and '311 Patent.

PRAYER FOR RELIEF

Plaintiff prays for entry of judgment against Defendants, jointly and severally, granting relief as follows:

A. judgment that Defendants have infringed and continue to infringe one or more claims of the ’336 and ’311 Patents, directly and/or indirectly, literally and/or under the doctrine of equivalents;

B. judgment that Defendants contribute to and induce the infringement of one or more claims of the ’311 and ’336 Patents, literally and/or under the doctrine of equivalents;

C. an Order for an accounting;

D. an award of damages pursuant to 35 U.S.C. §284 sufficient to compensate Plaintiff for Defendants’ past infringements, and any continuing or future infringement, up until the date that Plaintiff’s patent expires;

E. a determination of a reasonable royalty for any future infringement by Defendants, and an Order directing Defendants to pay such royalty on future infringement;

F. as assessment of pre-judgment and post-judgment interest and costs against Defendants, and an Order awarding such interest and costs, in accordance with 35 U.S.C. §284;

G. that Defendants be directed to pay enhanced damages, including Plaintiff's attorneys' fees, incurred in connection with this lawsuit pursuant to 35 U.S.C. §285;

H. an injunction against continued infringement, including but not limited to an injunction against Defendants and/or their agents; and

I. such other and further relief as this Court may deem just and proper.

JURY DEMAND

Plaintiff demands a trial by jury on all issues.

Date: May 11, 2020

Respectfully Submitted,

By: /Erik Dykema/

Erik Dykema
(NJ Bar No. 021652010)
erik@kzllp.com
Koning Zollar LLP
4 Manheim Road
Essex Fells, New Jersey 07021
T: 858.252.3234
F: 858.252.3238

Drew Koning
(pro-hac vice application to be filed)
drew@kzllp.com
Blake Zollar
blake@kzllp.com
(pro-hac vice application to be filed)
Koning Zollar LLP
169 Saxony Road, STE 115

Encinitas, CA 92024
T: 858.252.3234
F: 858.252.3238

Shaun Paisley
(pro-hac vice application to be filed)
shaun@kzllp.com
Koning Zollar LLP
470 James Street, Suite 007
New Haven, CT 06513
T: 203.951.1213
F: 858.252.3238

ATTORNEYS FOR PLAINTIFF
OANDA Corporation

EXHIBIT A

US007146336B2

(12) **United States Patent**
Olsen et al.

(10) **Patent No.:** **US 7,146,336 B2**
(45) **Date of Patent:** **Dec. 5, 2006**

(54) **CURRENCY TRADING SYSTEM, METHODS, AND SOFTWARE**

(75) Inventors: **Richard B. Olsen**, Zurich (CH); **Michael Stumm**, Toronto (CA)

(73) Assignee: **Oanda Corporation**, Toronto (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 673 days.

(21) Appl. No.: **09/858,610**

(22) Filed: **May 16, 2001**

(65) **Prior Publication Data**

US 2002/0156718 A1 Oct. 24, 2002

Related U.S. Application Data

(60) Provisional application No. 60/274,174, filed on Mar. 8, 2001.

(51) **Int. Cl.**
G06Q 40/00 (2006.01)

(52) **U.S. Cl.** **705/37**; 705/35

(58) **Field of Classification Search** 705/35, 705/37, 1

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,787,402 A * 7/1998 Potter et al. 705/37
5,916,307 A * 6/1999 Piskiel et al. 709/314
6,199,046 B1 * 3/2001 Heinzle et al. 705/1
2001/0044771 A1 * 11/2001 Usher et al. 705/37
2001/0049651 A1 * 12/2001 Selleck 705/37
2001/0056398 A1 * 12/2001 Scheirer 705/38
2002/0023053 A1 * 2/2002 Szoc et al. 705/39
2002/0087455 A1 * 7/2002 Tsagarakis et al. 705/37
2003/0115128 A1 * 6/2003 Lange et al. 705/37
2003/0159109 A1 * 8/2003 Rossmann et al. 715/513

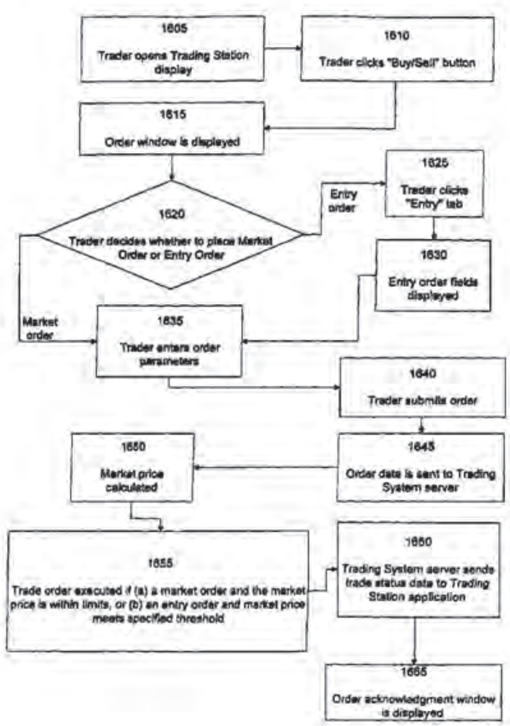
* cited by examiner

Primary Examiner—Alain L. Bashore
(74) *Attorney, Agent, or Firm*—Morgan, Lewis & Bockius LLP

(57) **ABSTRACT**

In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. In another aspect, the present invention comprises methods for trading currency over a computer network. In another aspect, the present invention comprises software for currency trading over a computer network.

11 Claims, 14 Drawing Sheets



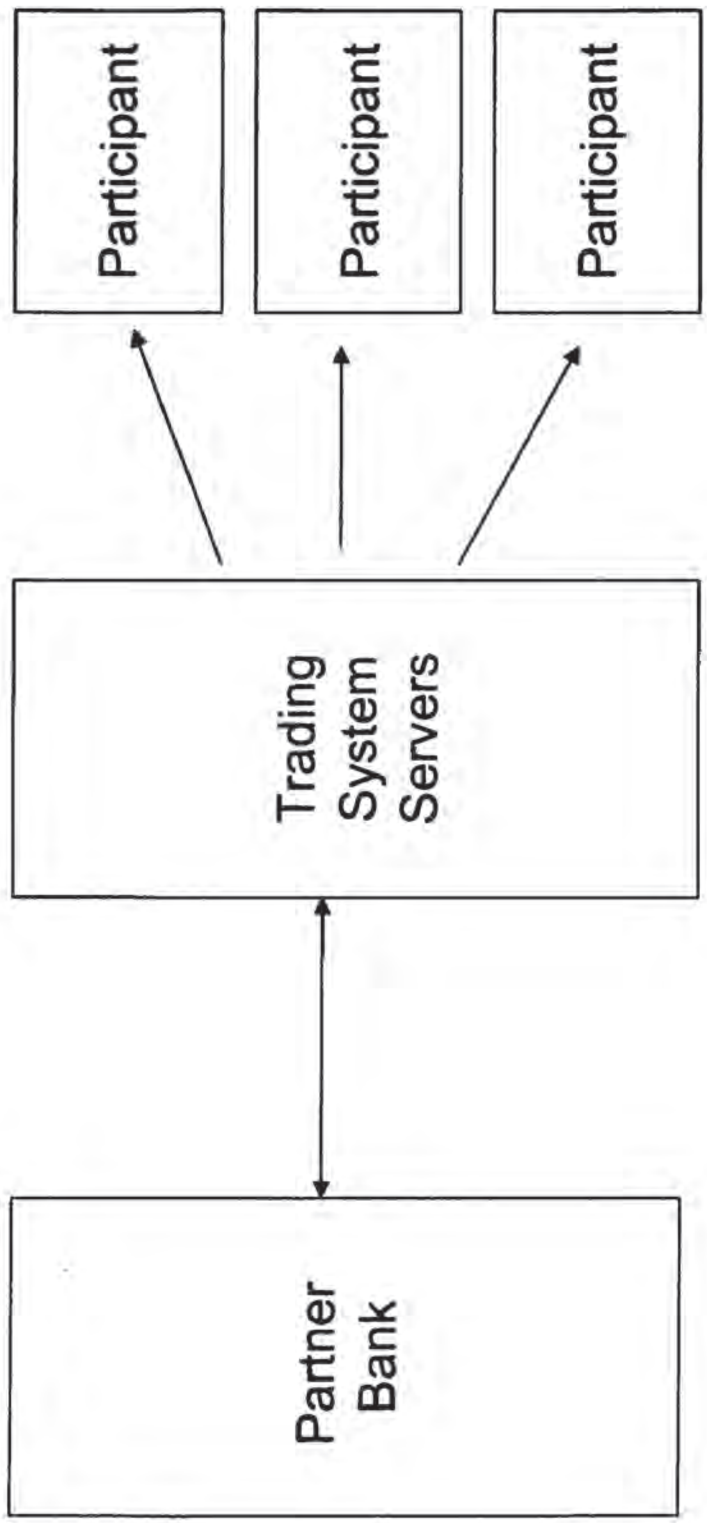

Participant
Participant
Participant
Trading System Servers
Partner Bank

Fig. 1

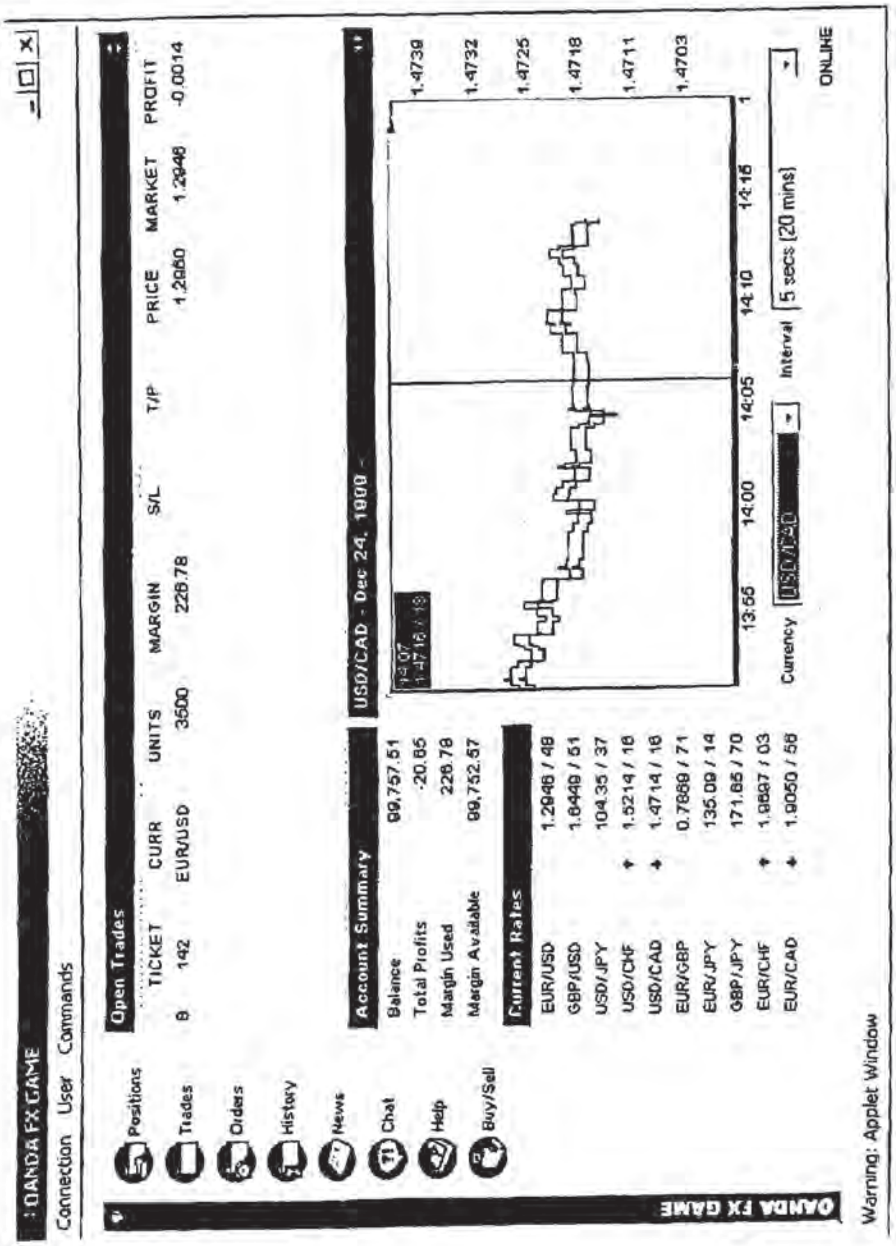

OANDA FX GAME
Connection User Commands
OANDA FX GAME
Positions
Trades
Orders
History
News
Chat
Help
Buy/Sell
Open Trades
TICKET CURR UNITS MARGIN S/L T/P PRICE MARKET PROFIT
142 EUR/USD 3500 228.78 1.2960 1.2948 -0.0014
Account Summary
Balance 99,757.51
Total Profits -20.65
Margin Used 228.78
Margin Available 99,752.57
Current Rates
EUR/USD 1.2946 / 49
GBP/USD 1.6449 / 51
USD/JPY 104.35 / 37
USD/CHF 1.5214 / 18
USD/CAD 1.4714 / 16
EUR/GBP 0.7869 / 71
EUR/JPY 135.09 / 14
GBP/JPY 171.65 / 70
EUR/CHF 1.9697 / 03
EUR/CAD 1.9050 / 56
USD/CAD - Dec 24, 1999
1.4739
1.4732
1.4725
1.4718
1.4711
1.4703
13:55
14:00
14:05
14:10
14:15
Currency
USD/CAD
Interval
5 secs (20 mins)
ONLINE
Warning: Applet Window

Fig. 2

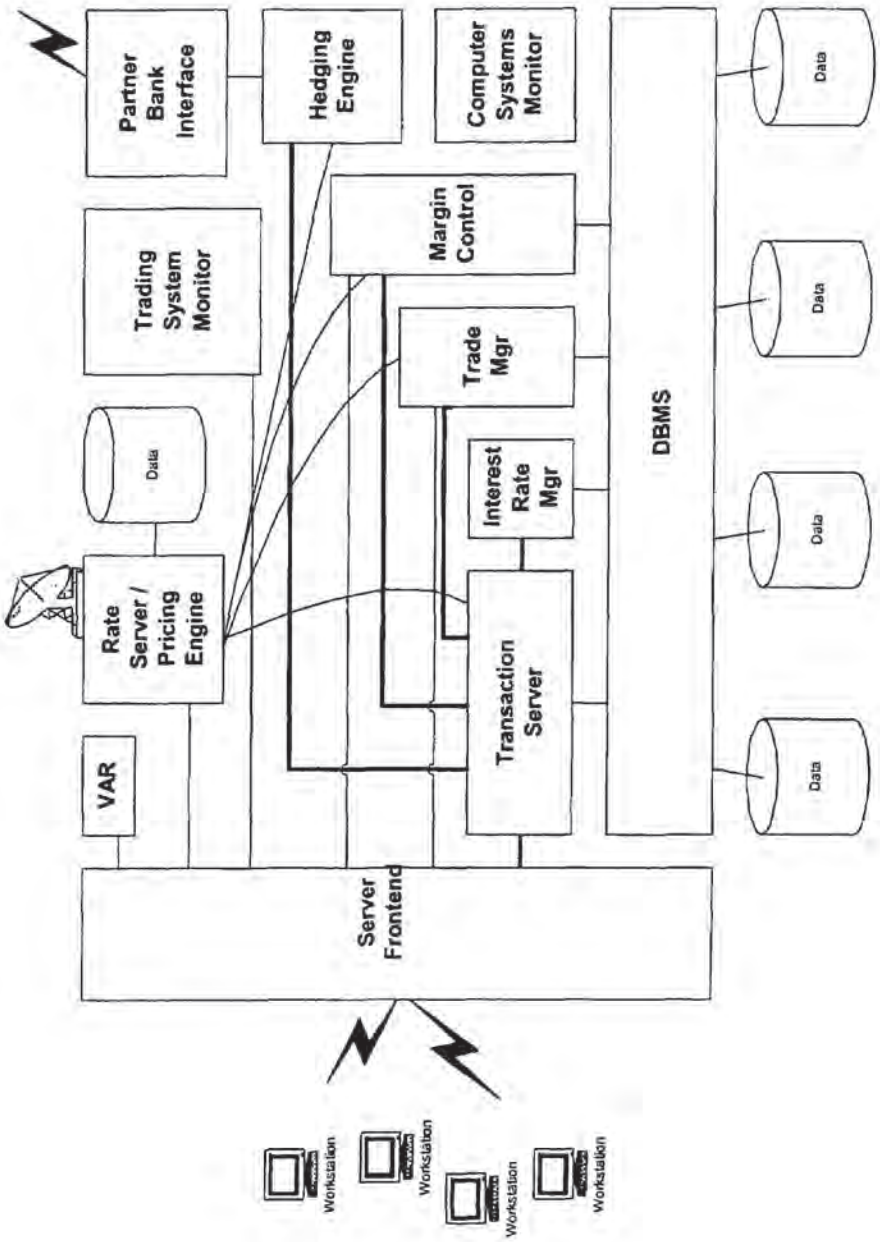

Partner Bank Interface
Hedging Engine
Computer Systems Monitor
Trading System Monitor
Margin Control
Trade Mgr
Interest Rate Mgr
DBMS
Data
Data
Data
Data
Data
Rate Server / Pricing Engine
VAR
Transaction Server
Server Frontend
Workstation
Workstation
Workstation
Workstation

Fig. 3

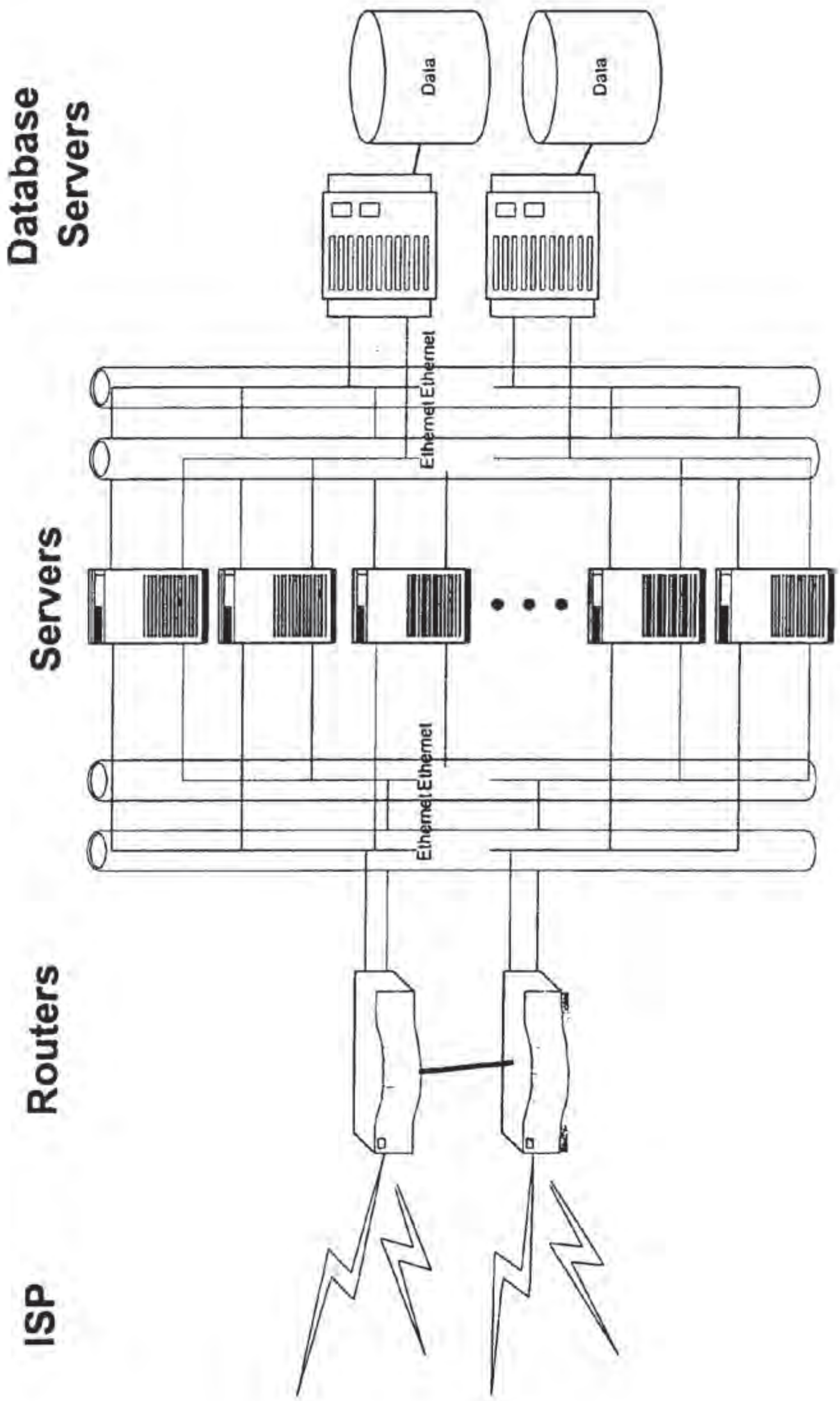
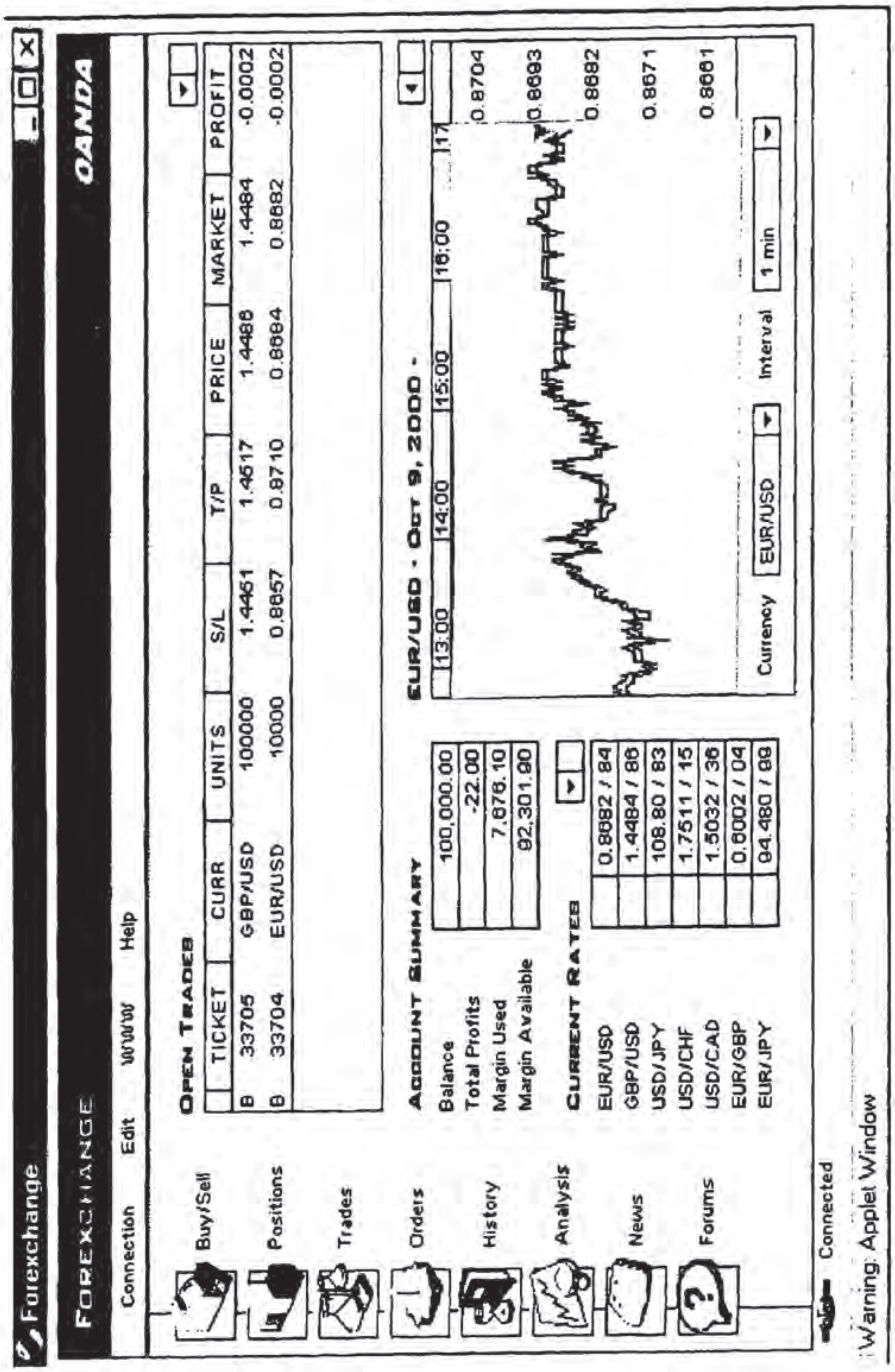

Database Servers
Data
Data
Ethernet Ethernet
Servers
Ethernet Ethernet
Routers
ISP

Fig. 4



Forexchange
FOREXCHANGE
OANDA
Connection
Edit
WWW
Help
Buy/Sell
Positions
Trades
Orders
History
Analysis
News
Forums
OPEN TRADES
TICKET
CURR
UNITS
S/L
T/P
PRICE
MARKET
PROFIT
B 33705 GBP/USD 100000 1.4461 1.4617 1.4488 1.4484 -0.0002
B 33704 EUR/USD 10000 0.8857 0.8710 0.8884 0.8882 -0.0002
ACCOUNT SUMMARY
Balance 100,000.00
Total Profits -22.00
Margin Used 7,676.10
Margin Available 92,301.90
CURRENT RATES
EUR/USD 0.8882 / 84
GBP/USD 1.4484 / 86
USD/JPY 108.80 / 83
USD/CHF 1.7511 / 15
USD/CAD 1.5032 / 36
EUR/GBP 0.6002 / 04
EUR/JPY 94.480 / 89
EUR/USD - OCT 9, 2000 -
13:00
14:00
15:00
16:00
0.8704
0.8693
0.8682
0.8671
0.8661
Currency
EUR/USD
Interval
1 min
Connected
Warning: Applet Window

Fig. 5

ACCOUNT SUMMARY

Balance	100,000.00
Total Profits	-22.00
Margin Used	7,676.10
Margin Available	92,301.90

Fig. 6

OPEN TRADES

	TICKET	CURR	UNITS	S/L	T/P	PRICE	MARKET	PROFIT
S	33760	USD/CHF	100000			1.7415	1.7413	0.0002
S	33759	EUR/USD	65000	0.8715	0.8668	0.8689	0.8692	-0.0003

Fig. 7

OPEN POSITIONS

	CURR	UNITS	AVG. PRICE	MARKET	PROFIT
S	USD/CHF	100000	1.7415	1.7407	0.0008
S	EUR/USD	65000	0.8689	0.8697	-0.0008

Fig. 8

OPEN ORDERS

	ORDER	CURR	UNITS	S/L	T/P	PRICE	MARKET	DURATION
B	245	EUR/JPY	200000	93.203	93.789	93.512	93.509	Undefined

Fig. 9

RECENT HISTORY

TRANSACTION	TYPE	CURR	UNITS	PRICE	DATE/TIME
56958	Buy Market	EUR/USD	10000	0.8684	Oct 9, 2000, 16:42
56959	Buy Market	GBP/USD	100000	1.4486	Oct 9, 2000, 16:43
57001	Take Profit	GBP/USD	100000	1.4604	Oct 11, 2000, 12:38
57021	Buy Market	EUR/USD	25000	0.8688	Oct 11, 2000, 15:04
57022	Sell Market	EUR/USD	25000	0.8689	Oct 11, 2000, 15:07

Fig. 10

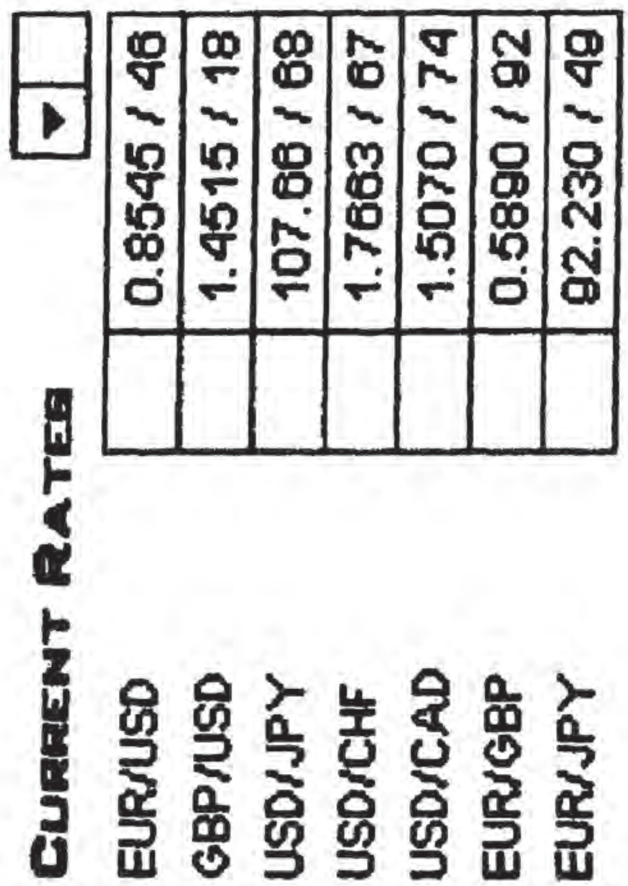

CURRENT RATES
EUR/USD
GBP/USD
USD/JPY
USD/CHF
USD/CAD
EUR/GBP
EUR/JPY
0.8545 / 46
1.4515 / 18
107.66 / 68
1.7663 / 67
1.5070 / 74
0.5890 / 92
92.230 / 49

Fig. 11

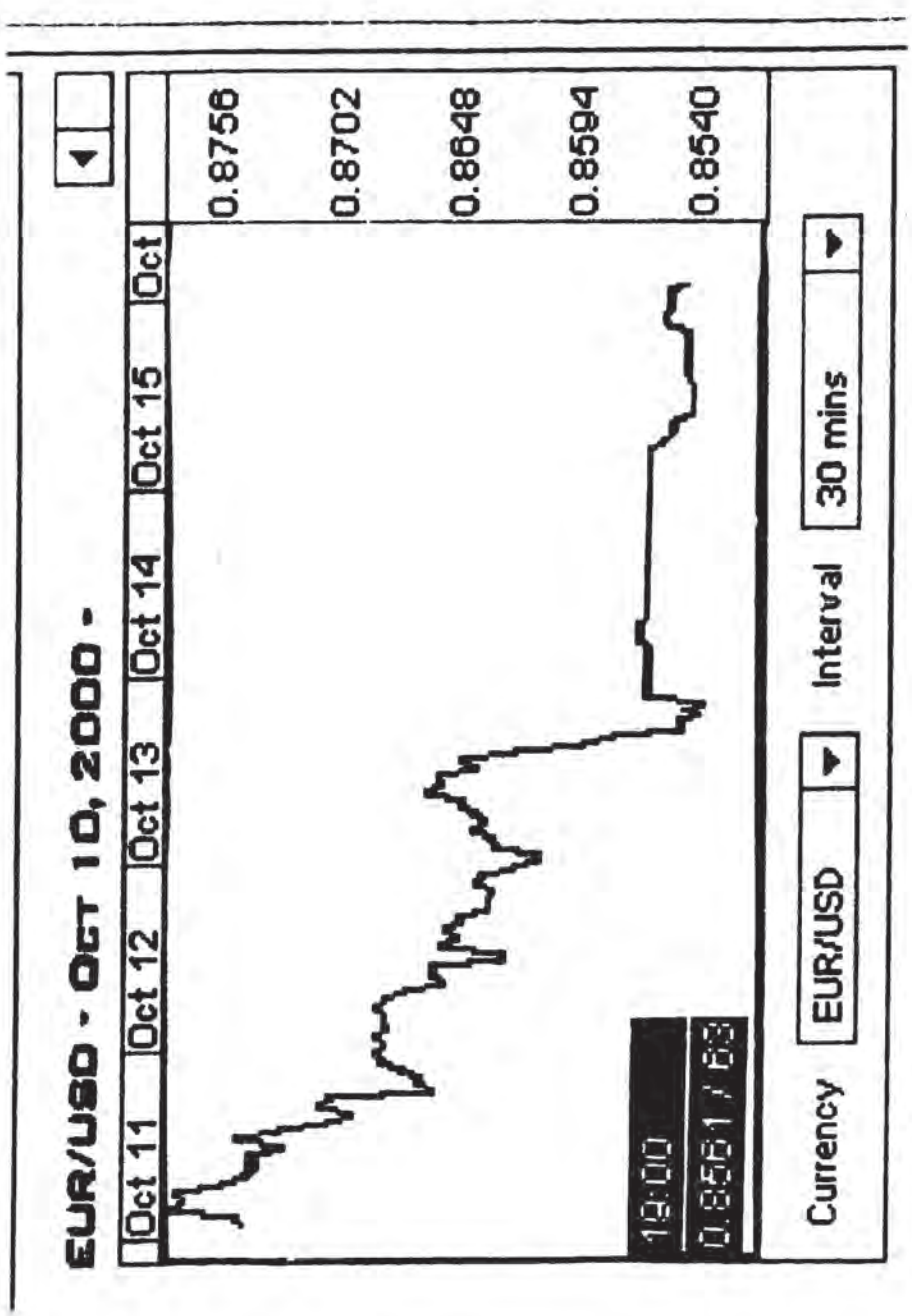

EUR/USD - OCT 10, 2000 -
Oct 11
Oct 12
Oct 13
Oct 14
Oct 15
Oct
0.8756
0.8702
0.8648
0.8594
0.8540
19:00
0.8561 / 63
Currency
EUR/USD
Interval
30 mins

Fig. 12

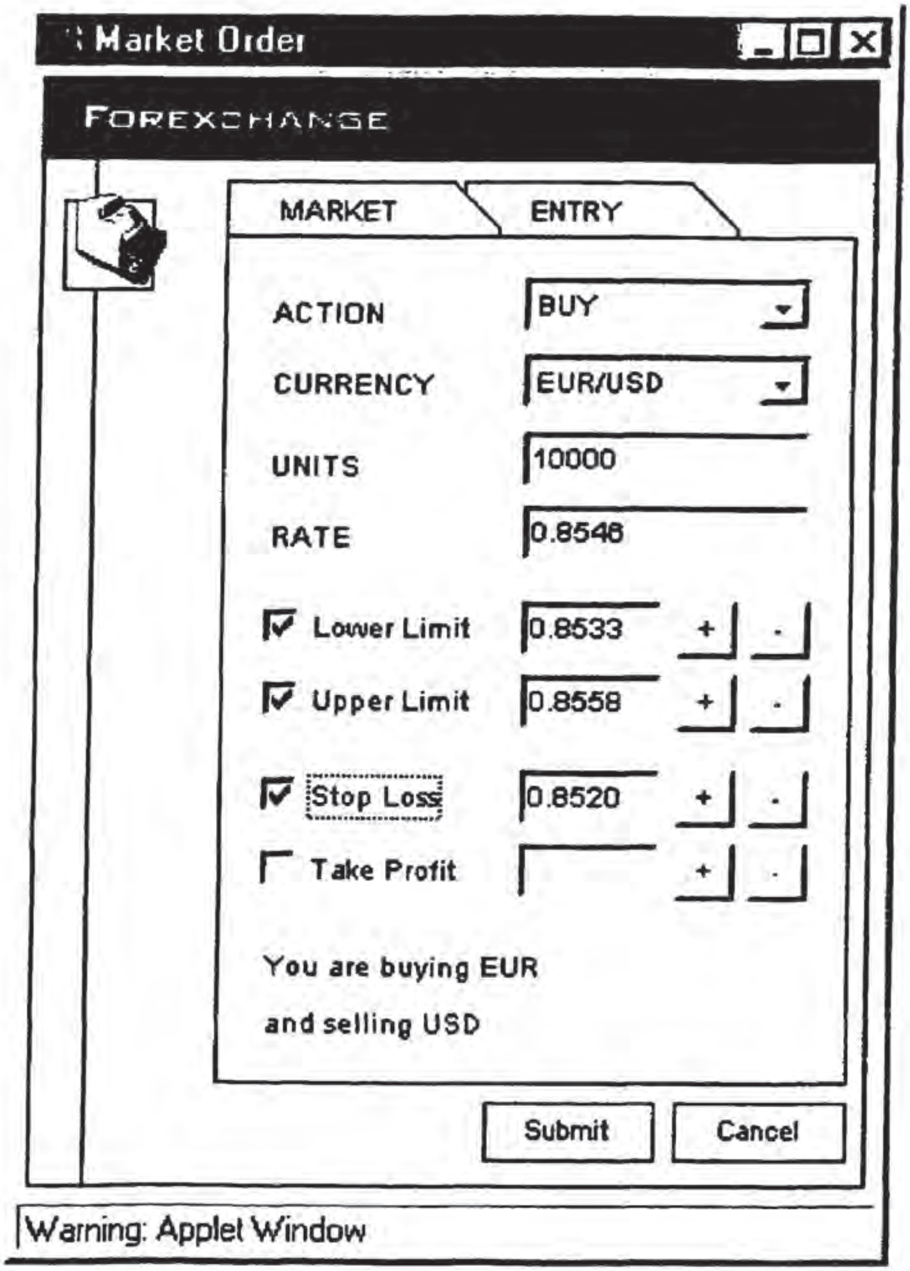

Market Order
FOREXCHANGE
MARKET
ENTRY
ACTION
BUY
CURRENCY
EUR/USD
UNITS
10000
RATE
0.8548
Lower Limit
0.8533
Upper Limit
0.8558
Stop Loss
0.8520
Take Profit
You are buying EUR
and selling USD
Submit
Cancel
Warning: Applet Window

Fig. 13

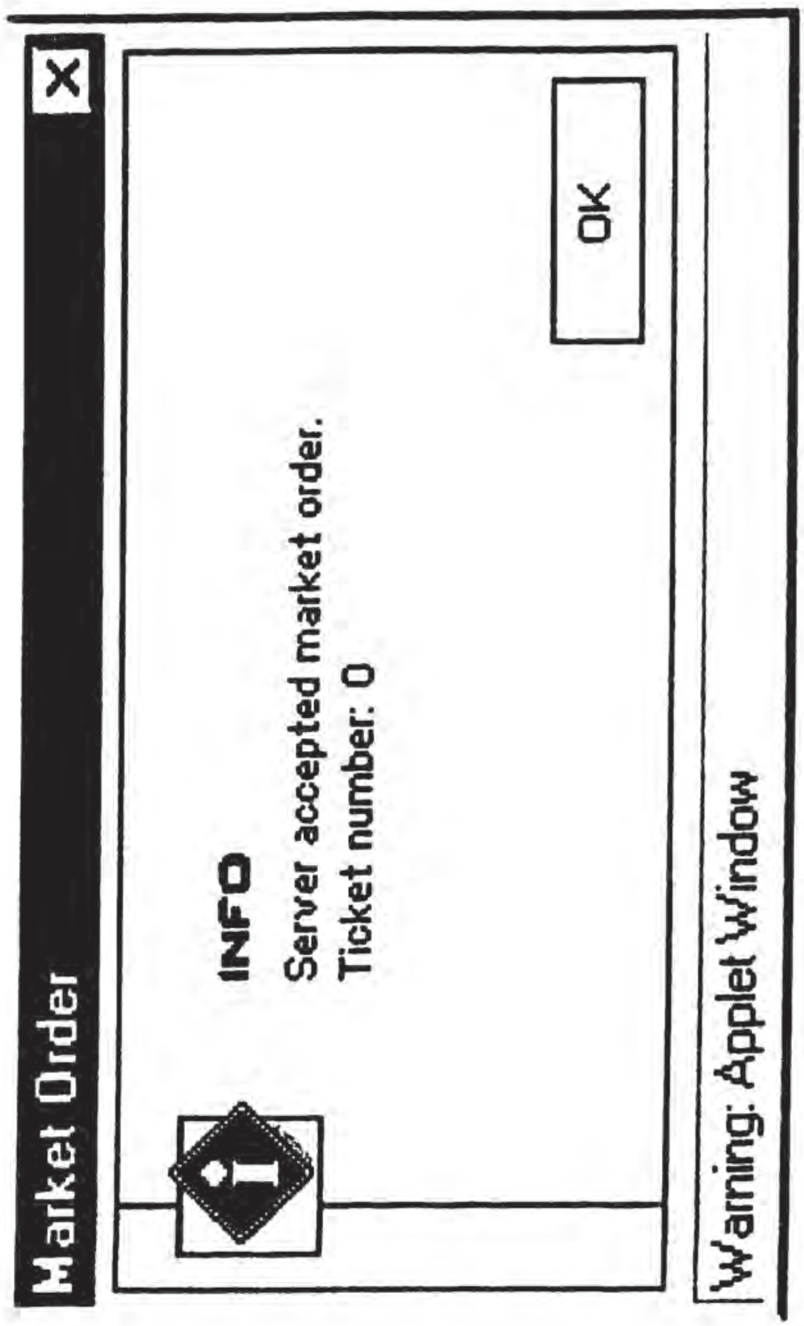

Market Order
INFO
Server accepted market order.
Ticket number: 0
OK
Warning: Applet Window

Fig. 14

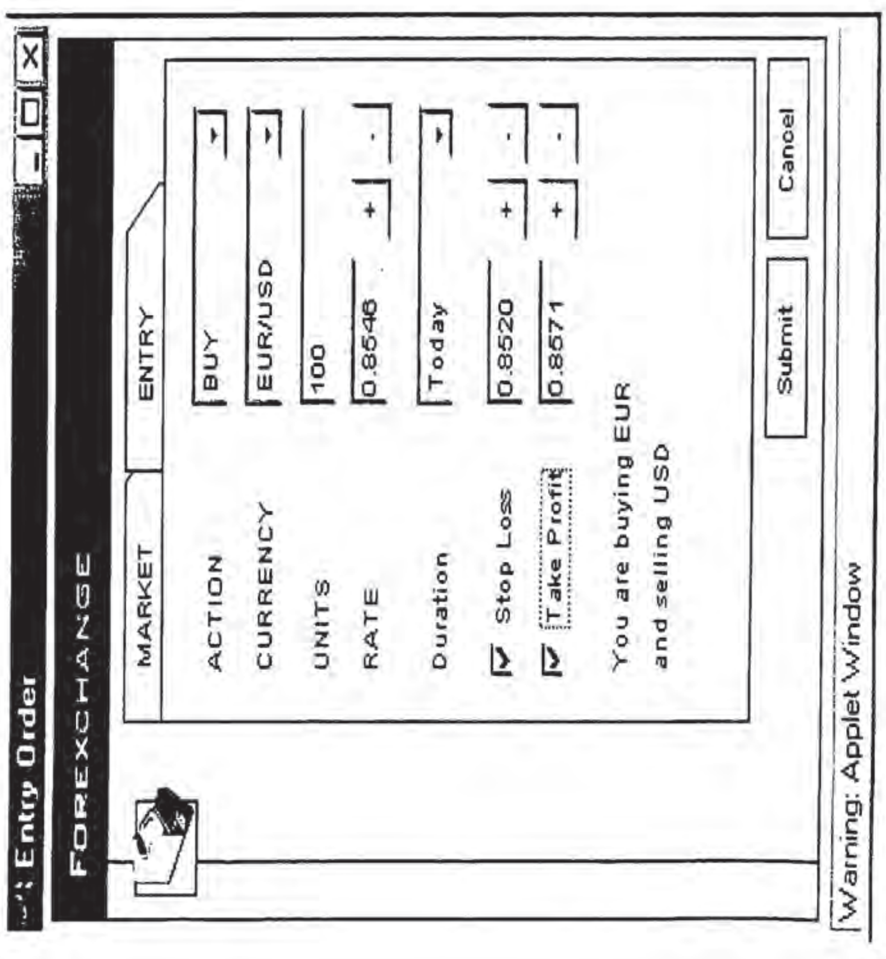

Entry Order
FOREXCHANGE
MARKET
ENTRY
ACTION
BUY
CURRENCY
EUR/USD
UNITS
100
RATE
0.8546
Duration
Today
Stop Loss
0.8520
Take Profit
0.8571
You are buying EUR
and selling USD
Submit
Cancel
Warning: Applet Window

Fig. 15

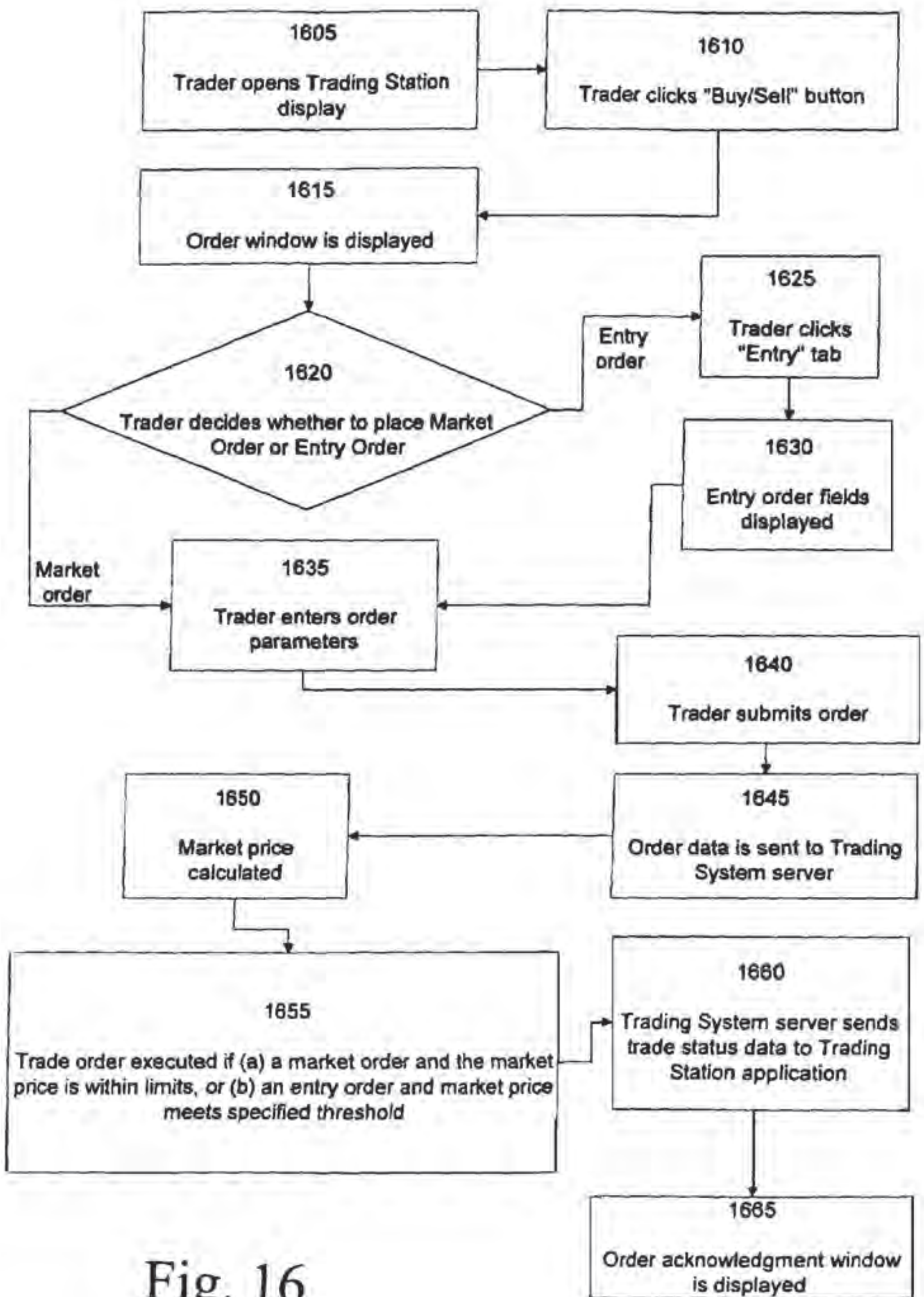

1605
Trader opens Trading Station display
1610
Trader clicks "Buy/Sell" button
1615
Order window is displayed
1620
Trader decides whether to place Market Order or Entry Order
Entry order
1625
Trader clicks "Entry" tab
1630
Entry order fields displayed
Market order
1635
Trader enters order parameters
1640
Trader submits order
1645
Order data is sent to Trading System server
1650
Market price calculated
1655
Trade order executed if (a) a market order and the market price is within limits, or (b) an entry order and market price meets specified threshold
1660
Trading System server sends trade status data to Trading Station application
1665
Order acknowledgment window is displayed

Fig. 16

CURRENCY TRADING SYSTEM, METHODS, AND SOFTWARE

CROSS-REFERENCE TO RELATED APPLICATION

This application claims priority to U.S. provisional application No. 60/274,174, filed Mar. 8, 2001, and incorporates the entire contents thereof herein by reference.

FIELD OF THE INVENTION

The present invention is related to currency trading; more particularly, the invention is related to trading currency over a computer network.

BACKGROUND

In a traditional on-line currency market, a trade occurs through three steps: (1) the trader specifies to a dealer the currency pair and the amount that he would to trade (but does not specify whether he would like to buy or sell); (2) the dealer specifies to the trader both a bid and an ask price and gives the trader several seconds to respond (the dealer not knowing whether the trader will buy, sell, or reject the offer); and (3) the trader either rejects the offer or specifies whether he is buying or selling (his response must occur within a time frame of a few seconds).

But performing such a three-way handshake over the Internet is somewhat impractical because of Internet delays: the trader might not actually have a few seconds to respond before the dealer withdraws the offer. Thus, there is a need for a system and method of on-line currency trading that is based on a trading model that is superior to the three-way handshake described above.

Another problem is that many corporations have firewalls that restrict access to the corporate network, and that typically restrict access to the Internet (and to well-known services such as email, the World Wide Web, etc.) from within the corporation. This inhibits the ability of on-line trading systems to access information from and transfer information to users behind corporate firewalls.

SUMMARY

The present invention overcomes the above-described and other disadvantages of previous currency trading systems and methods. In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. Each of these components is described in detail below in the Detailed Description section.

In another aspect, the present invention comprises methods for trading currency over a computer network. In one embodiment, a preferred method comprises: (a) transmitting currency market information over a computer network to an end user; (b) receiving a currency trade order from the end user, wherein the currency trade order comprises limits within which the currency trade will be acceptable to the end user; (c) calculating a market exchange rate for the currency trade order; and (d) executing the order if the market exchange rate is within the specified limits.

In another embodiment, a preferred method comprises: (a) transmitting currency market information over a computer network to an end user; (b) receiving a currency trade order from the end user, wherein the currency trade order comprises a threshold exchange rate; (c) calculating a market exchange rate for the received currency trade order; and (d) executing the order (1) if the market exchange rate is or goes above the threshold exchange rate and the order is a sell order, and (2) if the market exchange rate is or goes below the threshold exchange rate and the order is a buy order.

In a further embodiment, a preferred method comprises: (a) receiving currency market information over a computer network from a trading system server; (b) transmitting a currency trade order to the trading system server, wherein the currency trade order comprises limits within which the currency trade will be acceptable; and (c) if a market exchange rate is within the specified limits, receiving information from the trading system server indicating that the currency trade order was executed.

In another embodiment, a preferred method comprises: (a) receiving currency market information over a computer network from a trading system server; (b) transmitting a currency trade order to the trading system server, wherein the currency trade order comprises a threshold exchange rate; and (c) if (1) the applicable market exchange rate is or goes above the threshold exchange rate and the order is a sell order, or (2) the applicable market exchange rate is or becomes below the threshold exchange rate and the order is a buy order, receiving information from the trading system server indicating that the currency trade order was executed.

In another aspect, the present invention comprises software for currency trading over a computer network. In one embodiment, preferred software comprises: (a) software for receiving data over a computer network from a trading system server; (b) software for displaying a first graphical user interface display that: (i) displays continuously updated currency exchange rates in real-time based on data received from the trading system server; and (ii) displays action buttons, including a buy/sell button; (c) software for displaying, in response to a user clicking the buy/sell action button, a buy/sell window display that: (i) comprises trade order parameter fields; and (ii) accepts trade order data entered into the trade order parameter fields by a user; and (d) software for transmitting said trade order data to said trading system server over said computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** depicts parties involved in a preferred embodiment.

FIG. **2** depicts a graphical user interface of a preferred embodiment.

FIG. **3** depicts modules of a preferred trading system server.

FIG. **4** depicts hardware components of a preferred embodiment.

FIG. **5** depicts a graphical user interface of a preferred embodiment.

FIG. **6** depicts an account summary table display.

FIG. **7** depicts an open trades table display.

FIG. **8** depicts an open positions table display.

FIG. **9** depicts an open orders table display.

FIG. **10** depicts a transactions table display.

FIG. **11** depicts a currency rates table display.

FIG. **12** depicts a currency exchange rate graph display.

FIG. **13** depicts a buy/sell pop-up window display.

FIG. **14** depicts an acknowledgment window display.

FIG. **15** depicts an entry order display.

FIG. **16** depicts steps of a method of a preferred embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The Currency Trading System of a preferred embodiment (hereinafter "Trading System") of the present invention allows traders to trade currencies over a computer network. Preferably, this computer network is the Internet, and the subsequent description herein is primarily in terms of the Internet. However, those skilled in the art will recognize that the following description also applies to other computer networks. Traders interface to the system using ordinary Web browsers running feature-rich Java applets; they obtain real-time data feeds of current exchange rates, they can analyze past exchange rates using graphical tools, they can review their current portfolio and past trades, and they can place buy and sell orders in the real-time market. Businesses interface to the system using an API. Innovative features that set the Trading System apart from the competition are: (i) extremely low spreads on the order of a few basis points, (ii) the ability to trade very small amounts as low as $1, and (iii) 24 hour a day, 7 days a week availability. This system has the potential to revolutionize the way currency trading is done and to open up currency trading to a new, large market segment of investors and speculators for whom currency trading is not feasible today. Moreover, it allows businesses to address their currency exchange requirements in the most cost-effective and efficient way.

A description of the preferred server infrastructure used in the Trading System follows. We first give a brief introduction of the system as a whole.

The Trading System involves three components (see FIG. **1**): (1) traders that are distributed around the world; (2) Trading System servers; and (3) "Partners" consisting of the financial institution(s) through which real currency exchange trades are executed, and from which real-time data feeds are obtained.

Traders communicate with Trading System servers over a secure, encrypted Internet connection to review their accounts, to monitor currency exchange market conditions, or to initiate currency exchange trades. The Trading System servers are preferably connected to the partners' back-office systems, using direct, private lines.

A trader trades with the Trading System similar to the way she currently trades with a broker, except that the trading is over the Web, occurs 24 hours a day, 7 days a week, and allows very small trades with very low spreads. Moreover, an initial deposit, which may be as low as $20, can be charged to a credit card to get started. Alternatively, the trader can transfer initial funds directly to the Partner bank to be credited to her Trading System account.

The end user interface to the Trading System is a Web page that can be displayed on any standard Java-enabled browser. The Web page (one version is shown in FIG. **2**; a second, preferred version is shown in FIG. **5**) depicts a summary of the trader's current position, recent trading activities, profit/loss performance of the portfolio, and a graphical display of recent past performance of the currencies the trader has positions in, as well as real-time exchange rates.

As discussed above, in a traditional on-line currency market a trade occurs through three steps: (1) a trader specifies a currency pair and an amount he would like to trade (and does not specify whether he would like to buy or sell); (2) a dealer provides both a bid and an ask price to the trader and gives the trader several seconds to respond (the dealer does not know whether the trader will buy, sell, or reject the offer); and (3) the trader either rejects the offer or accepts the offer and specifies whether he is buying or selling, within the response period set by the dealer.

In the Internet domain, this type of three-way handshake is problematic. The timing constraints are difficult to implement because of frequent delays in transmission over the Internet. To overcome this problem, the present invention uses a "two-way handshake," in which: (1) a trader specifies in her trade order: (a) a currency pair; (b) a desired amount to trade; (c) whether she wishes to buy or sell; and (optionally) (d) upper and lower limits on an acceptable exchange rate; and (2) a dealer (in this case, a preferred Trading System) executes the trade using the most current "market rates" (as calculated by the system). However, the system only executes the order if the calculated market rate lies above any specified lower limit and below any specified upper limit. Note that this method does not require the use of timing constraints, and thus avoids the Internet-implementation disadvantage of previous methods.

In the present invention, trades can be initiated by the trader at the push of a button. A trading request form pops up with fields properly initialized so as to minimize the number of keystrokes required. A trader may elect to execute a trade right away, in which case the buyer of a currency will buy at the current exchange rate market offer price. Conversely, a trader can sell a currency at the current bid price. A range of automatic trading options is available, including setting bid/offer prices with a certain duration and "all-or-nothing" rules. Furthermore, the trader can limit her risks by placing stop-loss orders that are executed automatically. Similarly, she can lock in profits, by issuing take-profit orders.

All communication between the trader's browser and the Trading System server occurs through the Internet, preferably using the strongest available encryption (e.g., 128 bit keys). Moreover, the trader must authenticate herself using private passwords or certification keys obtained from certification authorities, such as Verisign or Entrust.

A request for a market trade preferably proceeds as follows: the trader, at a push of a button, obtains a trade order ticket in a popup window on the browser with key fields pre-initialized (see FIG. **13**). When the trade order is issued, again by the push of a button, a message is sent to a Trading System server, where the market price is calculated based on such factors as market data, size of the transaction, time of day, the Trading System's current exposure, and predictions on market direction. The trade order is executed using this market price. (The trader can specify limits, so that the trade occurs only if the price falls within these limits.) As such, the Trading System operates as a market maker. A message is then sent back to the trader with specific trade details, which is displayed in a popup window (see FIG. **14**) on the trader's browser together with a transaction id (for future reference). Moreover, an open orders table (see FIG. **9**) and current portfolio summary table (not shown) is updated to reflect the change.

Alternatively, the trader can issue in a similar manner an entry order (see FIG. **15**) that requests a trade be made when the currency exchange rate reaches a specified threshold. The trader may specify how long the entry order will be valid.

Referring to the attached figures, a preferred embodiment comprises a method of trading that in turn comprises the following steps (see FIG. **16**): At step **1605**, a trader desiring to trade opens a Trading Station display, and at step **1610** clicks a "Buy/Sell" button **510** on the Trading Station display (see FIG. **5**). At step **1615** an order window is displayed (see FIG. **13**). At step **1620** the trader decides whether to place a market order or an entry order. If a market order, then the trader proceeds to step **1635** and enters desired order parameters (as shown in FIG. **13**). If an entry order, then the trader proceeds to step **1625** and clicks an "Entry" tab **1320** (see FIG. **13**). At step **1630** entry order fields are displayed (see FIG. **15**). Then the trader proceeds to step **1635** and enters desired order parameters (as shown in FIG. **15**).

Once order parameters are entered at step **1635**, the trader submits the order by clicking a "Submit" button **1310** (see FIG. **13**) if the order is a market order, or clicking a "Submit" button **1510** (see FIG. **15**) if the order is an entry order. At step **1645** data describing the order is sent by the Trading Station application to a Trading System server, where the data is stored. At step **1650** a current market price for the currency the trader desires to purchase is calculated. At step **1655** the trader's order is executed if (a) the trader's order is a market order and the calculated market price is within the limits set by the trader in the market order form at step **1635**; or (b) order is an entry order and the calculated market price meets the threshold(s) specified in the Entry order form at step **1635**.

At step **1660** the Trading System server sends trade status data to the trader's Trading Station application. This data includes an indication that the order has been executed, if that is the case, and at any rate includes an indication that the order has been received. At step **1665**, the Trading Station application displays an order acknowledgment window (see FIG. **14**) that displays order status information.

Over time, the Trading System will accumulate an imbalance in its currency portfolio and, at times, it will need to neutralize its risk exposure to adverse currency fluctuations. The Trading System Pricing Engine can influence its exposure by setting its price quotes accordingly. Moreover, it can close out its positions periodically or take hedging positions by executing larger trades through its Partners. Preferably the Trading System's positions are managed based on state-of-the-art trading models. Preferred trading models are described in U.S. patent application Ser. No. 09/855,633, filed May 14, 2001, the contents of which are incorporated herein by reference, as well as U.S. provisional application No. 60/274,174, filed Mar. 8, 2001.

The Trading System servers preferably operate 24 hours a day, 7 days a week. These servers interface with the traders over the Internet on the one hand and on the other hand with the Partner's back-office operations. Using standard, state-of-the-art database technology, it maintains the accounts of all traders and executes trades issued by the traders. The Trading System thus plays the role of a market maker in that it internally aggregates all trades and only occasionally balances its internal positions by trading larger sums through the Partner. These larger trades are issued to the Partner in an automated way. The Trading System also takes hedging positions so as to minimize risks on the unbalanced portions of the traders' account aggregates.

Partner's Role

The Partner maintains all actual funds. It is the source and target of all fund transfers to and from customers; it maintains the aggregate accounts; and it executes all trades issued automatically by the Trading System servers. From a legal point of view, all funds must be maintained in money market instruments. Hence, the Partner will maintain a Long and a Short money market fund for each currency supported.

Overview of Currency Trading System Server Internal Design

The Trading System server architecture is designed to be:

(1) modular, in order to: (a) simplify development (time-to-market); (b) easily allow extensions and modifications; (c) ensure correctness and robustness, and (d) keep it maintainable;

(2) efficient, in order to provide fast response times to online users and minimize the computational and networking resources required to support the service;

(3) scalable, in order to support (with suitable computing infrastructure) a large number of online users and high transaction volumes; and

(4) fault resilient, so that any individual failure of a computing node or network connection does not interrupt service.

The Trading System server software preferably runs exclusively on Unix platforms, and is composed of the following modules, each with a distinct set of responsibilities (see FIG. **3**):

(1) Database (DBMS) **310**. This is the heart of the server. It keeps track of customer profile information, all customer accounts and all transactions, and ensures that atomicity, consistency, independence, and durability ("ACID") properties are maintained. The database is the reference point for all information kept by the system.

The database is preferably a standard commercially-available SQL database, configured for full replication for reliability, availability, and improved performance. The preferred embodiment is based on IBM's DB-2 product line, but Oracle, for example, could also easily be used.

(2) Server Front-end **315**. The server front-end **315** is responsible for all communication with the Web-based clients. It supports both persistent and non-persistent connections to the traders. The persistent connects are used primarily for periodic (i.e., every few seconds) transmission of the latest currency rates so that the traders can update the currency graphs and tables in real-time. Using persistent connections significantly reduces protocol processing overhead and reduces network bandwidth requirements. Non-persistent connections are used for all transaction-oriented requests, such as orders, transaction history requests, logins, etc. All transaction-oriented communication between the trader browsers and the Server Front-end occurs fully encrypted, while rate information is transmitted in unencrypted form for efficiency reasons.

Traders preferably communicate with the server using a request-response type of protocol. The Server Front-end **315** interprets each request it receives and, for each, takes appropriate action. For login requests, it sets up appropriate data structures so that all future requests can be serviced in the most efficient way. It also sets up encryption keys for secure communication, and logs the start of a new session with the Transaction Server. For rate requests, it returns the requested rates it obtains from the rate server. For orders, it executes the orders by issuing appropriate requests to the transaction server after checking the margin requirements, the availability of funds, and using rates as determined by the pricing engine. For stop-loss/take-profit and fixed-price orders (that may get executed in the future), the Trade Manager **365** is also informed. For each trade that gets executed, the Hedging Engine **340** and Margin Control **350** modules are informed, so that they always have an up-to-date snapshot of the state. For transaction history, the appropriate information is returned to the client after obtaining it from the Transaction Server **355**.

The Server Front-end **315** also encapsulates a standard Web server (a la Apache), that services other trader requests that entail formatted text; this includes all of the Help pages, large transaction history requests, and server monitoring information. The Server Front-end **315** also acts as a Firewall.

Internally, the Server Front-end **315** is structured as a set of threads that service one request after another. The threads allow concurrent request servicing so that many requests can be serviced in parallel.

(3) Rate Server/Pricing Engine **325**. The Rate Server obtains currency exchange rate information from a variety of external rate sources and stores it locally. The Pricing Engine computes the currency exchange rates that the traders see and that are used for trading. These are computed from the currency exchange rates obtained from the external rate sources, the directional movement and volatility of the market, the current Trading System exposure and a number of other parameters. The computed rates are made available to the other modules of the system, and are also stored on persistent media. Various methods of calculating such rates are known to those skilled in the art. A preferred method is described in U.S. patent application Ser. No. 09/764,366, filed Jan. 18, 2001, to Müller et al.

Traders can request historical rate data so that they can graphically display the movements of any pair of currencies. The Rate Server serves such requests and preferably has several years of currency exchange rates available for this purpose.

For fast response time, the Rate Server caches in memory all of the frequently and recently requested rates so as to minimize the number of disk accesses required.

(4) VAR (Value at Risk) Server **320**. This server obtains and serves Value at Risk information. Various methods of calculating VAR are known in the art. A preferred method is disclosed in U.S. Provisional Patent Application No. 60/200,742, filed May 1, 2000, to Müller.

(5) Transaction Server **355**. This server encapsulates all transaction functionality and communicates the transactions to the Database **310** server (which runs on a separate host) after validating the transactions. The Transaction Server **355** also updates all other modules that need to be informed of new transactions. Finally, the Transaction Server **355** informs the currently online traders when a transaction (that may have been issued by a stop-loss, take-profit, or limit order daemon or by the Margin Control Manager) takes place.

(6) Interest Rate Manager **360**. The Interest Rate Manager **360** periodically (for example, every few minutes, every few seconds, or tick-by-tick) goes through all trader accounts to compute the interest rate due or owed based on the instruments currently in the portfolio, each resulting in a transaction of the trader account. The portfolio information is obtained through the Transaction Server **355**. The interest rates used are obtained from external sources, and the history of interest rates are stored on persistent storage. Because real-time (or near real-time) information is used, the Interest Rate Manager is capable of calculating, paying out, and collecting interest by the second. Interest calculation formulas are known to those skilled in the art, and any appropriate formula can be used in the Interest Rate Manager without departing from the scope of the invention. An example is the formula

$$A = P\left(1 + \frac{r}{m}\right)^{mt}$$

where P is the principal, r is the annual interest rate, t is the time (in years) over which interest is earned, m is the number of times per year that interest is compounded, and A is the amount owed (principal plus interest). The interest earned during time t is A–P. Thus, for example, if the annual interest rate is 3%, and the interest is compounded daily, then the interest I_k earned over each time period $T_k=t_k-t_{k-1}$, where each t_k is a clock-time (i.e., a particular day-hour-minute-second-fraction-of-a-second) to the nearest second (thus T_k is in seconds), is calculated according to the formula $I_k=A_k-P_k$, where P_k is the principal (the amount earning interest, not the "original" principal) at time t_{k-1}, and

$$A_k = P_k\left(1 + \frac{0.03}{365}\right)^{\frac{365 T_k}{31,536,000}}.$$

Since there are 31,536,000 seconds per year,

$$\frac{T_k}{31,536,000} \text{ is}$$

the time in years over which the interest is being calculated. Similar formulas can be used when t_k is given to the nearest tenth, hundredth, or other fraction of a second. If interest is compounded continuously, those skilled in the art will recognize how to apply the well-known formula $A=Pe^{rt}$ appropriately. Thus, to calculate interest on a tick-by-tick basis, the above formulas can be used, with T_k representing time between ticks.

(7) Trade Manager **365**. The Trade Manager **365** continuously checks whether a trade should be executed on behalf of a trader, and if so executes the trade by interacting with the transaction server. The Trade Manager **365** consists of multiple subcomponents: (a) a stop-loss daemon continuously checks to see whether stop-loss orders should be executed and, if so, executes them through the Transaction Server **355**; (b) a take-profit daemon continuously checks to see whether take-profit orders should be executed and, if so, executes them through the Transaction Server **355**; and (c) a limit-order daemon continuously checks to see whether a limit order should be executed and, if so, executes it through the Transaction Server **355**.

The daemons continuously monitor the current rates to determine whether action is required. Moreover, each of the daemons caches in memory all of the orders that it may need to execute. They keep the orders suitably sorted so that they can take fast action when necessary; for example, the stop-loss daemon sorts the orders in descending order of stop-loss price, the take-profit in ascending order of take-profit price.

(8) Margin Control Manager **350**. This module continuously monitors the margin requirements of all trader accounts. When necessary, the Margin Control Manager **350** will liquidate some (or all) of a trader's holdings. It caches in memory all of the information necessary for this computation, sorted in decreasing order of risk, so that it can take

swift action when necessary. Holdings are liquidated through the Transaction Server **355**, when necessary.

(9) Trading System Monitor **330**. This module continuously monitors the current state of the Trading System. Among others, state parameters include: (a) current Trading System currency positions; (b) current margin situation; (c) a summary of stop-loss, take-profit, and limit orders that exist; (d) the number of currently online users; (e) the number, size and type of trades executed per second; and (f) a summary of the account positions held by the users.

This information is made available (a) to the Pricing Engine **325** (where it is used to set the currency exchange rates made available to the traders), (b) to the Hedging Engine **340** so that it can determine when to issue trades with the Partner Bankend Bank, and (c) to system operators and Trading System financial engineers in real time via a feature-rich Web interface. Moreover, this information is logged on persistent storage for later, off-line analysis.

(10) Hedging Engine **340**. This module continuously monitors current Trading System currency positions, the positions held in the trader accounts, recent trading activity, and the market direction and volatility to determine when to issue a trade with the backend Partner Bank. Various methods of performing such calculations are known to those skilled in the art. The Hedging Engine **340** preferably uses the hedging tool described in U.S. patent application Ser. No. 09/764,366, filed Jan. 18, 2001, to Müller et al., the contents of which are incorporated herein by reference.

(11) Partner Bank Interface **335**. This module communicates directly with the backend Partner Bank to issue trades and obtain account information.

(12) Computer Systems Monitor **345**. This module continuously monitors the operation and state of the computer systems on which the Trading System is running. Besides error conditions, such metrics as memory, processor, disk, and network utilization; paging activity; the number of packets sent over the various networks; the number of transactions; and the number of processes and threads are of interest. This information is made available to system operators in real time via a feature-rich Web interface and local consoles. In addition, it is stored on persistent storage for later, off-line analysis.

The Server modules described above are structured so that they can run independently as separate processes that can be independently mapped onto an arbitrary computer within a cluster. Moreover, each of the modules can run in replicated form, providing both fault tolerance and increased throughput.

Preferred Physical Organization of the Trading System Server

A Trading System Server of a preferred embodiment runs on a hardware base consisting of a cluster of hosts and disk farms connected by networks. All of the hardware components are preferably replicated for fault tolerance, as depicted in FIG. **4**.

The cluster is connected to multiple ISPs so that if one ISP goes down, traders can still communicate with the server. The ISPs are connected to the Server through a pair of routers **410** that monitor each other; if one of them goes down, then the other will automatically take over.

For security reasons, the Database **310** is on a separate back-end network; this way, it is not connected directly to the Internet and can only be accessed by the Transaction Server **355**. The Database **310** is setup in a dual configuration, so that the system can continue operating with a single database failure. All disks are mirrored, again, so that any single disk failure will not result in a loss of data.

All of the other server processes run on a cluster of servers **420**, connected to the Internet routers **410** on the one side, and connected to the backend database **430** on the other side. A virtually unlimited number of servers can be used in the cluster, allowing the system to scale up to support a large number of users. The servers can be partitioned by functionality, allowing specialized servers to be used, optimized for the particular functionality. For example, the Rate Servers **325** need minimal CPU power, and only a limited amount of memory. They also can be replicated easily without the introduction of any complexity or overhead. Hence, smaller, less costly hardware can be used for this purpose.

After login, traders typically communicate with a particular server in "sessions" for performance reasons. Using sessions improves cache locality, resulting in far fewer database accesses, and it allows the cost of creating session encryption keys to be amortized over many communication actions. For load balancing purposes, the trader software is directed to henceforth communicate with the least loaded server at the time when the trader first logs in. In case of severe load imbalances, individual traders are redirected to new, less loaded servers. If any of the servers crashes, then the client software that was communicating with the crashed server will detect the failure and automatically (transparently to the user) go through a new login procedure.

User Interface Description

Overview

The following is a description of a preferred user interface of a preferred Trading System. The main user interface display is called a "Trading Station," and it is used for all interactions with the trading system by a trader, such as analyzing changes in currency exchange rates, reviewing the trader's current currency positions, reviewing the trader's past transactions, or issuing buy and sell requests. The key features of the Trading Station are that: (1) it runs on any of the popular Web browsers connected to the Internet; (2) it displays continuously updated currency exchange rates in real-time; (3) it displays all pertinent information in one window; and (4) all interaction with the server occurs over fully-encrypted Internet connections.

System Requirements

The User Interface is preferably implemented in Java so as to run on any browser with JDK1.2 support, which includes all Netscape Navigators versions 4.2 and up as well as Microsoft's Internet Explorer versions 5.0 and up. The Trading Station is preferably supported for Windows 95, Windows 98, Windows 2000, Windows NT, Linux, Sun Solaris, and other Unix-based operating systems.

If operated from behind a firewall, then the Trading Station may operate properly only if the firewall allows HTTP requests to Port **90**. Many corporations have firewalls that restrict access to the corporate network to well-known services such as email. Typically this restriction is accomplished by restricting the ports that may be used. For example, Port **80** is typically used for http (Web-based) traffic. Some firewalls inspect traffic going through Port **80** to ensure that the port is being used only for Web-based traffic. This is problematic for trading systems that do not use http messages—it causes users behind corporate firewalls to be inaccessible. However, a preferred embodiment of the present invention overcomes this obstacle by prefacing Trading System messages with standard http headers to make them appear to be http requests and responses, even though they are not.

Log In Procedure

In order to log in, a trader must be a registered user. Registering is preferably free and can be accomplished by clicking on a "new users" link on a login page. Logging in requires a user to provide a user-ID and password. If a trader forgets her password, she can click on the "forgot your password" link and fill in the information requested; her password will be then be emailed to her.

If user-ID and password are entered correctly, a small window appears indicating that the Trading Station is being loaded. After a short time, a larger window appears with the Trading Station Graphical User Interface shown in FIG. **5**. Once the Trading Station is properly loaded, the contents of the small window is changed to include a number of useful links. It is important that this small window not be closed while the Trading Station is to remain in operation, although it may be minimized so as not to be in the trader's way. (This small window is necessary due to the limitations of typical Java implementations on most browsers that would otherwise not allow a trader to continue browsing the Web while the User Interface is active.)

Main Window of Trading Station

The Trading Station user interface is shown in FIG. **5**. It can be resized to a convenient size, by using the standard resizing mechanisms supported by the trader's operating system's windowing system.

The Trading Station is preferably partitioned into a number of components that each serve a different purpose:

(A) Action buttons: a vertical panel located on the left hand side of the Trading Station contains a set of action buttons that allow a user to perform the most common operations.

(B) Menus: a set of pull-down menus across the top allows a user to invoke additional operations.

(C) Account summary: an area in the middle of the Trading Station that gives a summary of the user's account.

(D) Table: an area located across the top of the Trading Station that is used to display various information in tabular format. The information displayed depends on the most recently clicked Action Button. It might display currently held instruments, current open positions, or a history of recent transactions.

(E) Currency rates: an area at the bottom left that displays various currency rates. These rates are continuously updated in real time.

(F) Graph: located at the bottom-right corner, graphs display currency rates over time. The graphs are also updated in real-time as new rates become available.

Subsequent sections describe each of these components in detail.

Action Buttons

The Trading Station preferably has the following action buttons in a panel on the left side. Clicking the appropriate button will invoke the described operation:

Buy/Sell: Pops up a Buy/Sell window, from which a trader can issue trade requests. (See the description of the buy/sell window (FIG. **13**) for more information.)

Positions: Displays the currently open positions in a table. (See the description of the Open Positions Table (FIG. **8**) for the contents of the table.)

Trades: Displays the currently open trades in a table. (See the description of the Open Trades Table (FIG. **7**) for the contents of the table.)

Orders: Displays the open orders (that may be executed some time in the future) in a table. (See the description of the Open Orders Table (FIG. **9**) for the contents of the table.)

History: Displays a recent history of the trader's transactions in the table. (See the description of the History Table (FIG. **10**) for contents of the table.)

Analysis: Pops up a new browser window with access to a number of analysis tools that might help in making trading decisions.

News: Pops up a new browser window with the latest currency news.

Forums: Pops up a new browser window with access to a number of forums (sometimes known as newsgroups) that allow a trader to participate in discussions with other traders and currency trading experts.

Pull-down Menus

There are preferably 5 pull-down menus (not shown), each offering different operations or services:

Connection Menu: (1) Disconnect: disconnects the Trading Station from the Trading System server. The Trading Station will remain open, but currency rates will no longer be updated, and transactions will not be possible. (2) Connect: connect the Trading Station to the server, so the trader is back on line. (3) Quit: quit and exit this application.

Account Menu: (1) Transaction history: pop up a new browser window to display an extensive list of all transactions that occurred on a trader's account. See the Transaction History section (relating to FIG. **10**) for a description of what is displayed. (2) Clear account balance and P/L: for those who have incurred large losses on their account, this operation allows a trader to start over again with a cleared P/L and new funds in the account. This feature is primarily useful when an account on the Trading System is used as a game—i.e., no real money changes hands. (3) Add funds to account: for a game account, add funds to the account for a real money-account, transfer money into the account from the trader's credit card or obtain instructions on how to wire transfer money into the trader's account. (4) Buy/Sell: issue a trade or market order (see FIG. **13**). (5) Open positions: display the open positions in a table (see FIG. **8**). (6) Open trades: display all open trades in a table (see FIG. **7**). (7) Open orders: display all open orders in a table (see FIG. **9**). (8) Recent transaction history: display the most recent transactions in a table (see FIG. **10**).

Commands Menu: (1) Change passwords. (2) Graph: specify the currency pair to be displayed in the graph.

Information Menu: (1) Interest rates: display interest rate information in a separate browser window. (2) Market News: display up-to-date currency market news in a separate browser window. (3) Analysis tools: use an analysis tool in a separate window. (4) Forum: participate in various forums related to currency trading. (5) Rankings: see a list of the most successful currency traders using the Trading System.

Help Menu: (1) Documentation: links to descriptive documents. (2) About: display software version number and credits. (3) Debug: display debugging information in a new window.

Account Summary

The account summary display (see FIG. **6** for an example) is a small table that provides a summary of the trader's account status. It preferably shows: (1) Account Balance: the amount of the trader's cash holding in the trader's account. (2) Realized P/L: the amount of profit or loss the trader has incurred with the trader's trading activity to date. (3) Unrealized P/L: the amount of profit or loss that the trader holds with the trader's current open positions. If the trader clears all of his open positions, then this amount would be added to the Realized P/L amount. (4) Margin Used: the amount of the trader's account balance and unrealized P/L tied up for margin purposes. (5) Margin Available: the amount of the trader's account balance and unrealized P/L available as margin for new trading transactions.

This information is preferably continuously updated in real-time to take current market conditions into account. Moreover, the information is always shown in the trader's home currency.

Tables

The table area of the Trading Station shows different information depending on the last Action Button-selected. It can include: (1) open trades; (2) open positions; (3) open orders; and (4) transaction history. The default is open positions.

How the information in the table is displayed can be controlled in two ways: (1) Scroll bars are used to scroll the table up or down, allowing a trader to see information that is hidden from view. (2) Sorting can be achieved by clicking on a column header, which causes the table to be sorted so that the column is in increasing or decreasing order. Clicking once sorts the column in increasing order; clicking again sorts it in decreasing order.

For all tables except Transaction history, clicking on a row of the table will cause a pop-up window to appear, offering further actions for that open trade, position, or order.

Open Trades Table

The open trades table (see FIG. **7**) shows a list of the trader's currently open trades. The table preferably has 9 columns, described from left to right (not all are depicted in FIG. **7**):

(1) Short/Long: Indicates whether the position is short or long.

(2) Ticket Number: a number that uniquely identifies an open trade. A trader can use this number as a reference for inquiries to the Trading System or its operators, or to search for particular entries in the transaction history table.

(3) Currency pair: the pair of currencies involved in this trade. The first currency of the pair is referred to as the base currency, while the second one is referred to as the quote currency.

(4) Units: the number of transacted units for this trade, expressed in the base currency.

(5) S/L: the trader's stop-loss for this trade. This trade will be closed automatically as soon as the currency exchange rate for this currency pair crosses the S/L value. A stop-loss limit is used to limit the loss a trader may incur with this trade.

(6) T/P: the trader's take-profit for this trade. This trade will be closed automatically as soon as the currency exchange rate for this currency pair crosses the T/P value. A take-profit limit is used to realize the trader's profit as soon as it reaches the T/P value.

(7) Rate: the exchange rate obtained when the trade got executed.

(8) Market: the current exchange rate for this currency pair.

(9) Profit: the unrealized profit (when positive) or loss (when negative) expressed in base currency and on a per unit basis.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order. Clicking on a row with an open trade will cause a pop-up window to appear offering two operations: (1) Close trade. (2) Modify trade. This is used to modify the S/L or the T/P limits.

Open Positions Table

The Open Positions Table (see FIG. **8**) displays a list of the trader's open positions. It is similar to the Open Trades table, except that all trades of the same currency pair are aggregated into one line.

The table preferably has 6 columns, described from left to right (not all are shown in FIG. **8**):

(1) Short/Long: Indicates whether the position is short or long.

(2) Currency pair: the pair of currencies the position refers to. The first currency of the pair is referred to as the base currency, while the second one is referred to as the quote currency.

(3) Units: the number of units held in this position, expressed in the base currency.

(4) Rate: the average exchange rate obtained for the trades in this position.

(5) Market: the current exchange rate for this currency pair.

(6) Profit: the unrealized profit (when positive) or loss (when negative) expressed in base currency and on a per unit basis.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column displayed in increasing or decreasing order. Clicking on a row with an open position will cause a pop-up window to appear offering the option to close the position.

Open Orders Table

The Open Orders Table (see FIG. **9**) shows a list of the trader's currently open orders. An open order is a request that a particular trade should be made automatically when the exchange rate of the specified currency pair crosses a specified threshold.

The table preferably has 9 columns, described from left to right (not all are shown in FIG. **9**).

(1) Short/Long: indicates whether the position is short or long.

(2) Order ID: a number that uniquely identifies the order. A trader can use this number as a reference for inquiries to the Trading System.

(3) Currency pair: the pair of currencies to be traded.

(4) Units: the number of units to be traded, expressed in the base currency.

(5) S/L: the stop-loss for this trade. This trade, once executed, will be closed automatically as soon as the currency exchange rate for this currency pair crosses the S/L value. A stop-loss limit is used to limit the loss a trader may incur with this trade.

(6) T/P: the trader's take-profit for this trade. This trade, once executed, will be closed automatically as soon as the currency exchange rate for this currency pair crosses the T/P value. A take-profit limit is used to realize the trader's profit as soon as it reaches the T/P value.

(7) Rate: specifies that the trade should be executed as soon as the exchange rate for the specified currency pair crosses this value.

(8) Market: the current exchange rate for this currency pair.

(9) Duration: specifies the amount of time an order should stand, until it is automatically canceled.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order. Clicking on a row with an order will cause a pop-up window to appear offering two operations: (1) Cancel order. (2) Modify order. This is used to modify the exchange rate threshold at which the trade is to be executed, or the S/L or T/P limits.

Transactions Table

The Transactions (or Transaction History) Table (see FIG. **10**) shows a list of the most recent transactions on the account. For access to a full list of past transactions, a user selects the Information pull-down menu and then selects Transaction History.

The Transaction History Table preferably has 6 columns, described from left to right:

(1) Transaction ID: uniquely identifies the transaction.

(2) Type: identifies the type of transaction.

(3) Currency pair: the pair of currencies associated with the transaction.

(4) Units: the number of units to traded in the transaction, expressed in the base currency.

(5) Price: the currency exchange rate applied when buying or selling a currency pair.

(6) Date/Time: the date and time of the transaction.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order.

Currency Rates

The Currency Rates Table (see FIG. **11**) shows the current exchange rate for the currency pairs supported by the Trading System. They are preferably updated in real time, approximately every 5 seconds. When there is a significant exchange rate movement for a currency pair, up/down indicators show the direction of the rate change in order to alert a trader, should a trader not currently have the currency pair displayed in the graph.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a currency pair's ask price will pop up a buy window for that currency pair. Clicking on a currency pair's bid price will pop up a sell window for that currency pair.

Graphs

Graphs (see FIG. **12**) show how currency exchange rates change over a period of time, ranging from minutes to months. All graphs are updated in real-time, as new currency rates arrive.

At any given time, the difference between the lower boundary and the upper boundary of the curve represents the difference between the bid and the ask price, and the difference may vary over time depending on market conditions. Thus, the top part of the curve indicates the ask price, and the bottom of the curve indicates the bid price.

As a mouse cursor **1220** is moved over the graph, a sub-area **1230** in the graph shows precise exchange rate information for the target currency pair corresponding to the time instance represented by the position of the mouse cursor.

The graph may also display Buy or Sell widgets that indicate at which point in time a trader bought or sold a currency pair. Downward pointing red arrows indicate a sold currency pair (where a trader is hoping the rates will go down after that point), and upward pointing green arrows

indicate a bought currency pair (where a trader is hoping the rates will go up after that point).

A trader can adjust what is shown in the graph: (1) The currency pair displayed is selected using the pull-down menu **1240** at the bottom left. (2) The granularity of the graph is selected using the pull-down menu **1250** at the bottom right of the graph. Selecting a fine granularity, such as 5 seconds (where each point on the horizontal axis represents 5 seconds of time), will display a relatively short time interval (less than an hour, in this case). Selecting a larger granularity, such as one day, will display longer-tern trends (9 months of exchange rate information in this case).

Scroll buttons **1260** at the top right of the graph area allow a trader to shift the time interval shown to the left or to the right (backward or forward in time). Clicking on the graph with the mouse will hide the Buy/Sell widgets. Clicking again will cause them to reappear.

Buy/Sell Window

A Buy/Sell pop-up window (see FIG. **13**) allows a trader to issue buy or sell orders. The window can be caused to pop up either by: (1) clicking on the Buy/Sell action button (see FIG. **5**); (2) clicking on the bid or ask price in the Currency Rates Table (see FIG. **11**); or (3) clicking on an existing trade, position, or order in the Table area of the Trading Station display (see FIG. **5**).

Two types of orders are supported: (1) Market Orders are orders that are transacted immediately based on market exchange rates. (2) Entry Orders are orders that are executed when the exchange rate crosses a certain threshold.

The type of order can be selected by clicking on the appropriate tab in the Buy/Sell Window (see FIG. **13**). Market order comes up as the default order.

Issuing a Market Order. To issue a market order with the Buy/Sell Window and the Market Tab selected, a number of fields must be filled out (although most of the fields are pre-initialized with reasonable values):

(1) ACTION: choose between buy and sell.

(2) CURRENCY: choose the currency pair the trader wishes to buy or sell. By default, this field will be initialized as follows: (A) If the Buy/Sell button was used to obtain the window, the currency pair currently shown in the graph. (B) If the bid or ask price was clicked to obtain the window, the currency pair for which the price was clicked. (C) If a trade or position was clicked in the Table area, the currency pair corresponding to the trade or position. The pull-down menu can be used to select another currency pair.

(3) UNITS: the number of units of the currency pair the trader wishes to buy or sell, with units expressed in terms of the base currency.

(4) Lower Limit: the order will result in a trade only if a price is obtained that does not lie below this limit. By default, no limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however the value can be changed either by modifying the number directly or by using the +/– buttons to increase or decrease the value, respectively.

(5) Upper Limit: the order will result in a trade only if a price is obtained that does not lie above this limit. By default, no limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/– buttons to increase or decrease the value, respectively.

(6) Stop Loss: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no stop-loss limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(7) Take Profit: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no take profit limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

The RATE field is set by the Trading Station and corresponds to the most recent exchange rate for the selected currency pair.

To issue the order, a Submit button **1310** must be selected. If the order is successful, and a trade occurs, then an acknowledgment window (see FIG. **14**) pops up with a Ticket number that can be used for future reference. Moreover, the Open Trades Table (see FIG. **7**) will be updated to reflect the new trade, as will the Open Positions Table (see FIG. **8**) and the Transaction History Table (see FIG. **10**).

Several issues are important to note:

(1) If an order is successful and a trade occurs, then the exchange rate obtained for the trade will correspond to the most current exchange rate maintained at the Trading System servers and not necessarily the rate displayed in the Buy/Sell window.

(2) An order without Lower and Upper Limits will always result in a trade.

(3) An order with both Lower and Upper Limits will result in a trade if and only if the exchange rate for the potential trade lies between the two limits.

Issuing an Entry Order. To issue an entry order with the Buy/Sell Window and the Entry Tab selected (see FIG. **15**), a number of fields must be filled out (although most of the fields are pre-initialized with reasonable values):

(1) ACTION: choose between buy and sell.

(2) CURRENCY: choose the currency pair the trader wishes to buy or sell. By default, this field will be initialized as follows: (A) if the Buy/Sell button was used to obtain the window, the currency pair currently shown in the graph; (B) if the bid or ask price was clicked to obtain the window, the currency pair for which the price was clicked. The pull-down menu can be used to select another currency pair.

(3) UNITS: the number of units of the currency pair the trader wishes to buy or sell, with units expressed in terms of the base currency.

(4) RATE: the order will result in a trade as soon as the exchange rate for the selected currency pair crosses the given value; that is, for buy orders, if the rate goes below this value, and for sell orders if the rate goes above the given value

(5) Duration: this value is used to limit the amount of time an outstanding order will remain effective. By default, the order remains effective indefinitely. However, the duration can be set to the end of the day or for an hour.

(6) Stop Loss: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no stop-loss limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(7) Take Profit: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no take profit limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

To issue the order, a Submit button **1510** must be selected. This results in an acknowledgment window (see FIG. **14**) popping up with a Ticket number that can be used for future reference. Moreover, the Open Orders Table (see FIG. **9**) will be updated to reflect the new order. Note that a trader can modify the parameters of an open order (including the rate representing the trade threshold, or the S/L and T/P) by clicking on the order in the Open Orders Table.

What is claimed is:

1. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; and further comprising an interest rate manager in communication with said transaction server and said database, wherein said interest rate manager is operative to calculate, pay out, and collect interest on a tick-by-tick basis.

2. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; and further comprising a trade manager in communication with said transaction server and said database, wherein said trade manager comprises a stop-loss daemon that (a) continuously checks whether stop-loss orders should be executed, and (b) if a stop-loss order should be executed, executes it through said transaction server.

3. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; and further comprising a trade manager in communication with said transaction server and said database, wherein said trade manager comprises a take-profit daemon that (a) continuously checks whether take-profit orders should be executed, and (b) if a take-profit order should be executed, executes it through said transaction server.

4. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; and further comprising a trade manager in communication with said transaction server and said database, wherein said trade manager comprises a limit-order daemon that (a) continuously checks whether limit orders should be executed, and (b) if a limit order should be executed, executes it through said transaction server.

5. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server, wherein said pricing engine is operable to compute currency exchange rates based on: (a) data obtained from external rate sources; and (b) market directional movement and volatility.

6. A system as in claim 5, wherein said pricing engine is further operable to compute currency exchange rates based on positions held by said system.

7. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; further comprising a hedging engine in communication with said transaction server, wherein said hedging engine is operable to perform at least two of the following calculations: (a) calculate a total amount of home currency appearing in all open positions; (b) calculate an out-of-equilibrium exposure; (c) calculate a new potential net exposure; (d) calculate an equilibrium position; (e) calculate boundaries of possible exposures; (f) calculate values for a pair of quoting functions; and (g) calculate an average price and an average spread.

8. A system as in claim 6, wherein said positions are managed based on one or more trading models.

9. A system as in claim 8, wherein at least one of said one or more trading models comprises: (a) a price collector component; (b) a price filter component; (c) a price database component; (d) a gearing calculator component; (e) a deal acceptor component; and (f) a book-keeper component.

10. A system as in claim 8, wherein at least one of said one or more trading models comprises: (a) a price collector component; (b) a price filter component; (c) a price database component; (d) a gearing calculator component; (e) a deal acceptor component; (f) an opportunity catcher component; and (g) a book-keeper component.

11. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; further comprising a margin control manager in communication with said transaction server and said database, wherein said margin control manager is operable to monitor on a tick-by-tick basis margin requirements of accounts and on said tick-by-tick basis liquidate holdings as needed to maintain specified margins.

* * * * *

EXHIBIT B

US008392311B2

(12) **United States Patent**
Olsen et al.

(10) **Patent No.: US 8,392,311 B2**
(45) **Date of Patent: *Mar. 5, 2013**

(54) **CURRENCY TRADING SYSTEM, METHODS, AND SOFTWARE**

(75) Inventors: **Richard B. Olsen**, Zurich (CH); **Michael Stumm**, Toronto (CA)

(73) Assignee: **Oanda Corporation** (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1305 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/634,014**

(22) Filed: **Dec. 4, 2006**

(65) **Prior Publication Data**

US 2007/0078755 A1 Apr. 5, 2007

Related U.S. Application Data

(63) Continuation of application No. 09/858,610, filed on May 16, 2001, now Pat. No. 7,146,336.

(60) Provisional application No. 60/274,174, filed on Mar. 8, 2001.

(51) **Int. Cl.**
G06Q 40/00 (2012.01)

(52) **U.S. Cl.** **705/37**

(58) **Field of Classification Search** 705/37
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,787,402 A *	7/1998	Potter et al.	.....	705/37
H2064 H *	5/2003	Buchalter	.....	705/37
H0002064 H *	5/2003	Buchalter	.....	705/37
2001/0049651 A1*	12/2001	Selleck	.....	705/37
2002/0023053 A1*	2/2002	Szoc et al.	.....	705/39

OTHER PUBLICATIONS

Cagan, Penny. Financial risk management sources: A walk on the wild side; EContent 22.6 (Dec. 1999): 16-25.*

By Darren McDermott; Corporate Performance 1999 Review: 1st quarter—Latest Pro; Wall Street Journal [New York, N.Y] May 3, 1999: C1.*

Turton, Jonathon. It is time rethink your netting; Corporate Finance 175 (Jun. 1999): 32-34.*

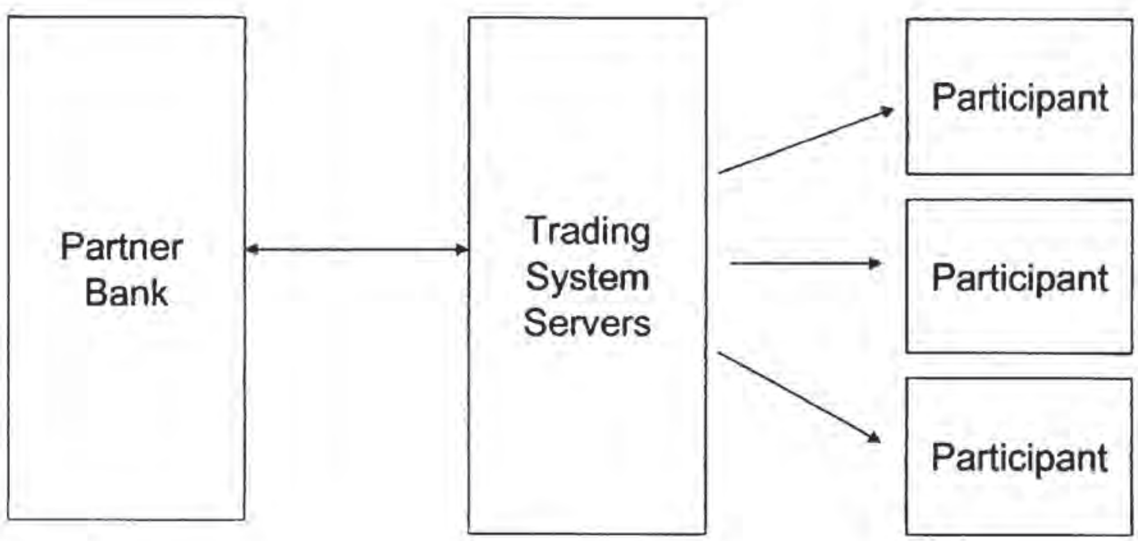
* cited by examiner

Primary Examiner — Hani M Kazimi
Assistant Examiner — Hatem M Ali
(74) *Attorney, Agent, or Firm* — Ward & Zinna, LLC

(57) **ABSTRACT**

In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. In another aspect, the present invention comprises methods for trading currency over a computer network. In another aspect, the present invention comprises software for currency trading over a computer network.

7 Claims, 14 Drawing Sheets



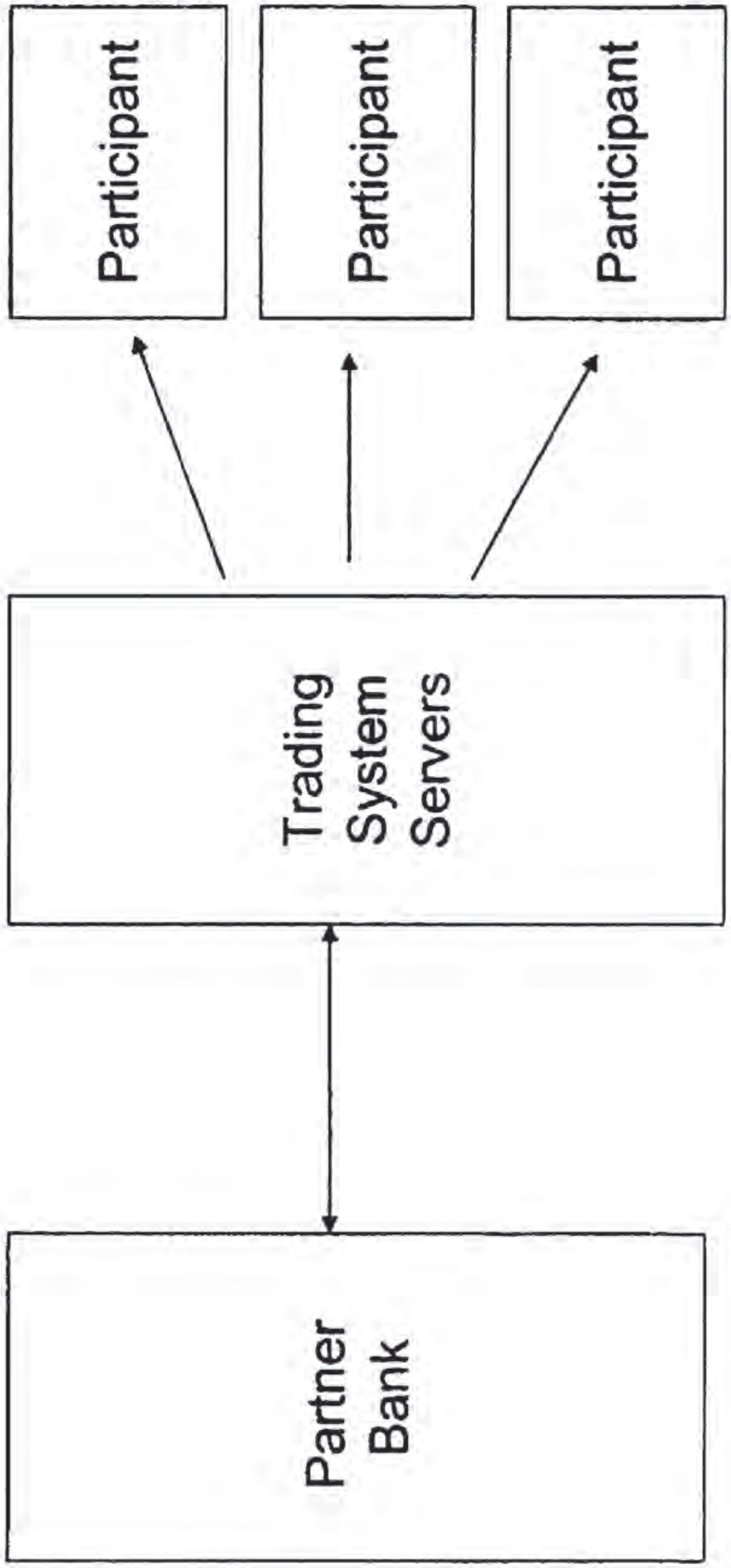

Participant
Participant
Participant
Trading System Servers
Partner Bank

Fig. 1

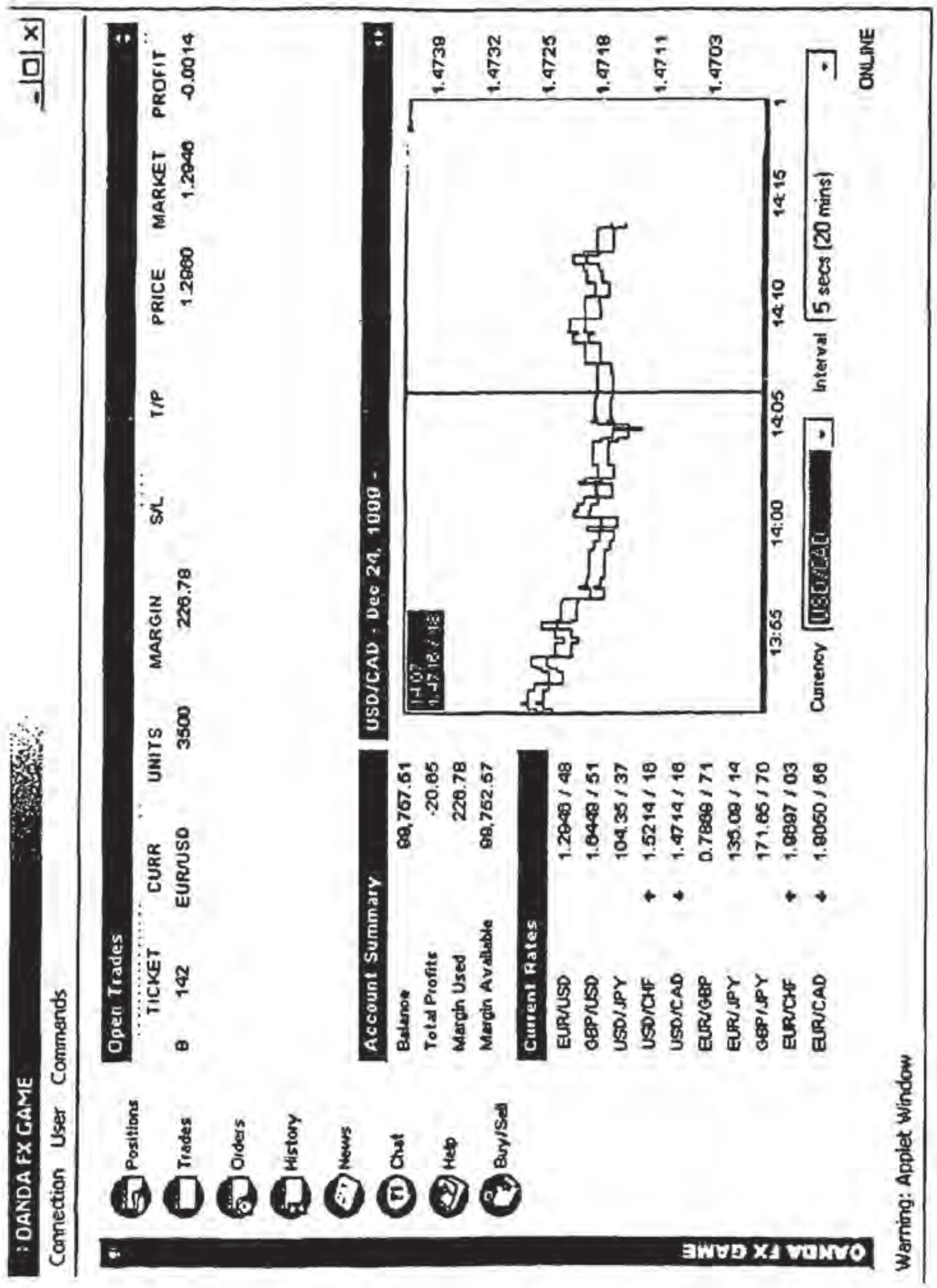

OANDA FX GAME
Connection User Commands
Positions
Trades
Orders
History
News
Chat
Help
Buy/Sell
Open Trades
TICKET CURR UNITS MARGIN S/L T/P PRICE MARKET PROFIT
Account Summary
Balance
Total Profits
Margin Used
Margin Available
Current Rates
USD/CAD - Dec 24, 1000
Currency
Interval
5 secs (20 mins)
ONLINE
Warning: Applet Window

Fig. 2

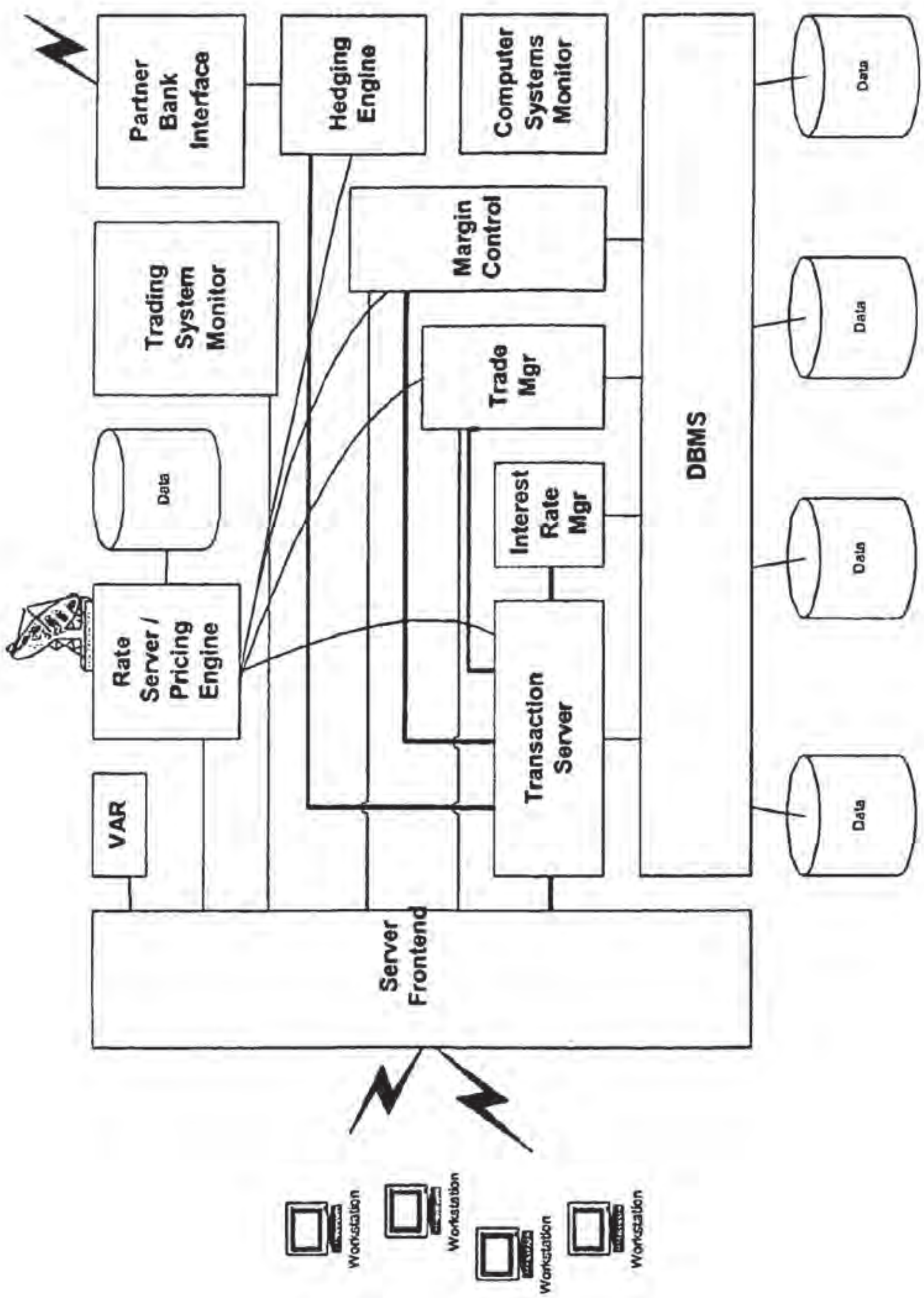

Partner Bank Interface
Hedging Engine
Computer Systems Monitor
Data
Trading System Monitor
Margin Control
Trade Mgr
Data
Data
Interest Rate Mgr
DBMS
Rate Server / Pricing Engine
Data
Transaction Server
VAR
Data
Server Frontend
Workstation
Workstation
Workstation
Workstation

Fig. 3

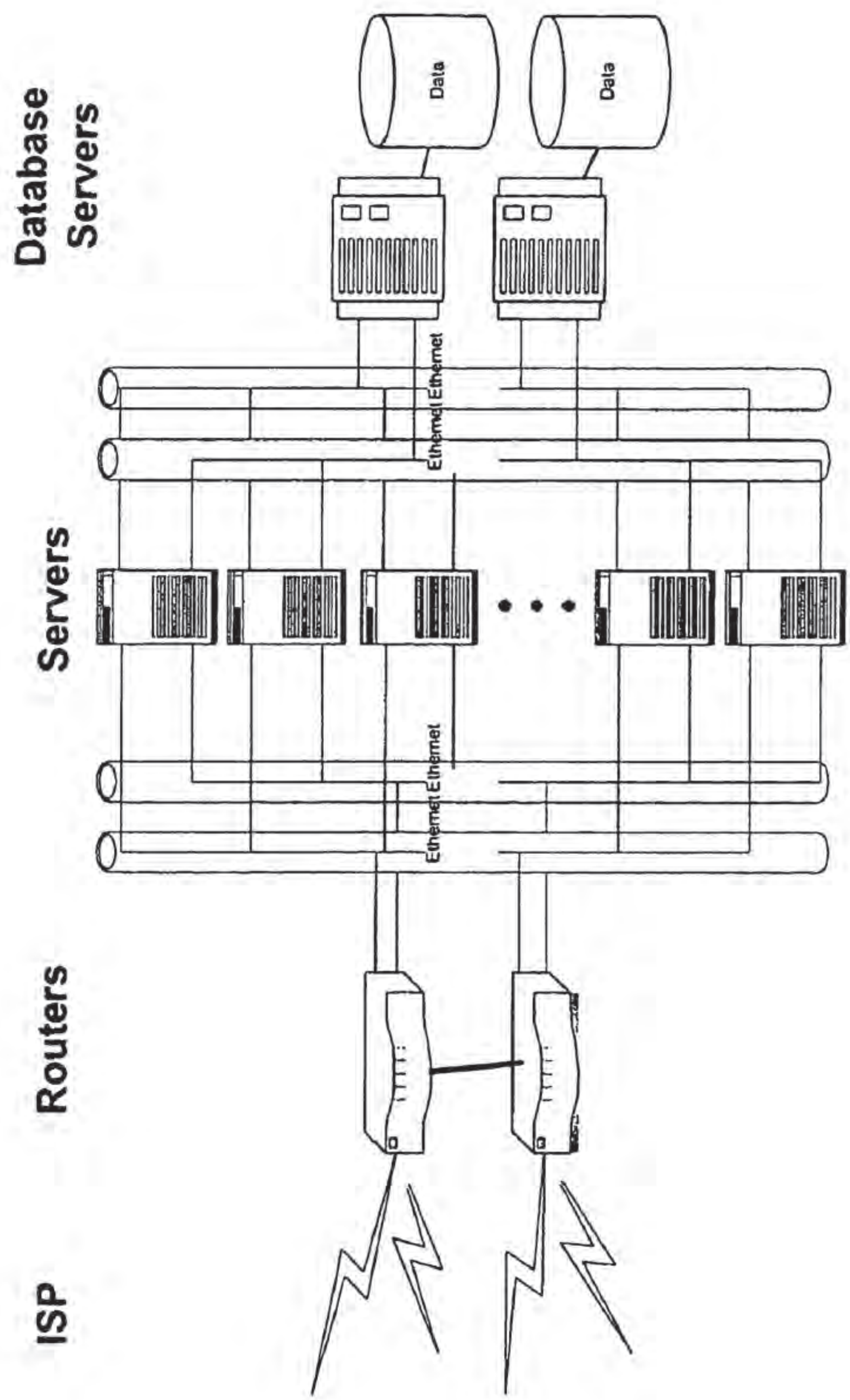

Database Servers
Data
Data
Ethernet
Ethernet
Servers
Ethernet
Ethernet
Routers
ISP

Fig. 4

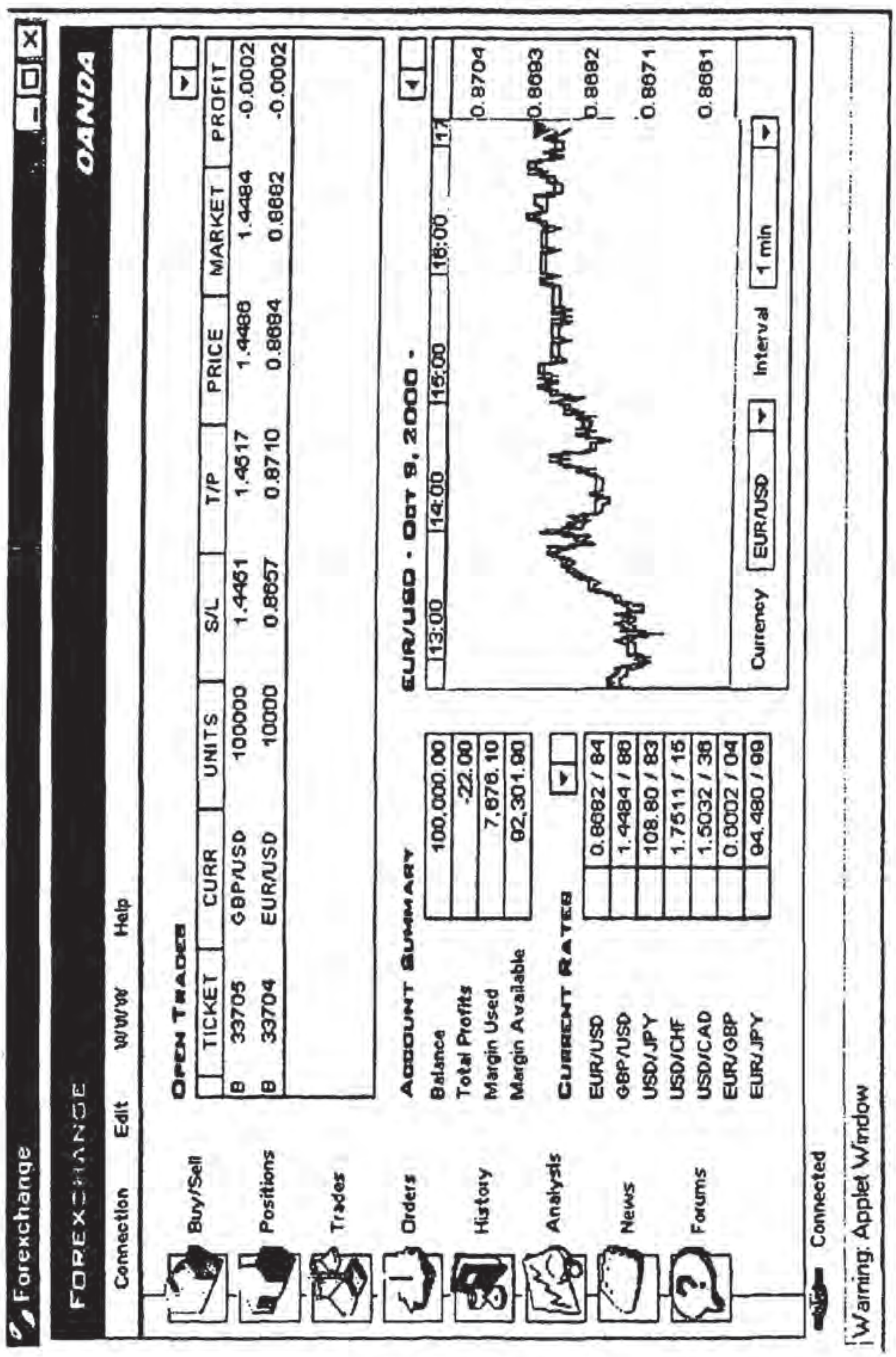

Forexchange
FOREXCHANGE
OANDA
Connection
Edit
WWW
Help
Buy/Sell
Positions
Trades
Orders
History
Analysis
News
Forums
OPEN TRADES
TICKET CURR UNITS S/L T/P PRICE MARKET PROFIT
B 33705 GBP/USD 100000 1.4451 1.4517 1.4486 1.4484 -0.0002
B 33704 EUR/USD 10000 0.8657 0.8710 0.8684 0.8682 -0.0002
ACCOUNT SUMMARY
Balance 100,000.00
Total Profits -22.00
Margin Used 7,678.10
Margin Available 92,301.90
CURRENT RATES
EUR/USD 0.8682 / 84
GBP/USD 1.4484 / 88
USD/JPY 108.80 / 83
USD/CHF 1.7511 / 15
USD/CAD 1.5032 / 38
EUR/GBP 0.6002 / 04
EUR/JPY 94.480 / 99
EUR/USD - OCT 9, 2000 -
13:00
14:00
15:00
16:00
17
0.8704
0.8693
0.8682
0.8671
0.8661
Currency
EUR/USD
Interval
1 min
Connected
Warning: Applet Window

Fig. 5

ACCOUNT SUMMARY

Balance	100,000.00
Total Profits	-22.00
Margin Used	7,676.10
Margin Available	92,301.90

Fig. 6

OPEN TRADES

	TICKET	CURR	UNITS	S/L	T/P	PRICE	MARKET	PROFIT
S	33760	USD/CHF	100000			1.7415	1.7413	0.0002
S	33759	EUR/USD	65000	0.8715	0.8668	0.8689	0.8692	-0.0003

Fig. 7

OPEN POSITIONS

	CURR	UNITS	AVG. PRICE	MARKET	PROFIT
S	USD/CHF	100000	1.7415	1.7407	0.0008
S	EUR/USD	65000	0.8689	0.8697	-0.0008

Fig. 8

OPEN ORDERS

	ORDER	CURR	UNITS	S/L	T/P	PRICE	MARKET	DURATION
B	245	EUR/JPY	200000	93.203	93.789	93.512	93.509	Undefined

Fig. 9

RECENT HISTORY

TRANSACTION	TYPE	CURR	UNITS	PRICE	DATE/TIME
56958	Buy Market	EUR/USD	10000	0.8684	Oct 9, 2000, 16:42
56959	Buy Market	GBP/USD	100000	1.4486	Oct 9, 2000, 16:43
57001	Take Profit	GBP/USD	100000	1.4604	Oct 11, 2000, 12:38
57021	Buy Market	EUR/USD	25000	0.8888	Oct 11, 2000, 15:04
57022	Sell Market	EUR/USD	25000	0.8889	Oct 11, 2000, 15:07

Fig. 10

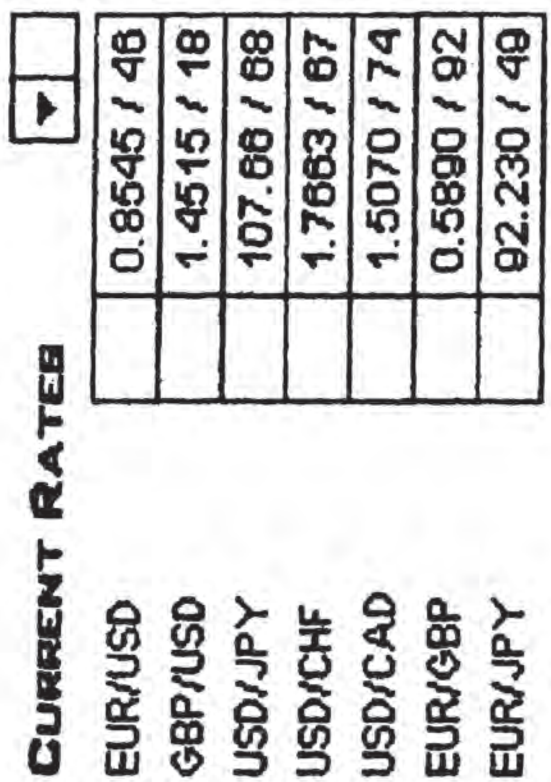

CURRENT RATES
EUR/USD
0.8545 / 48
GBP/USD
1.4515 / 18
USD/JPY
107.66 / 68
USD/CHF
1.7663 / 67
USD/CAD
1.5070 / 74
EUR/GBP
0.5890 / 92
EUR/JPY
92.230 / 49

Fig. 11

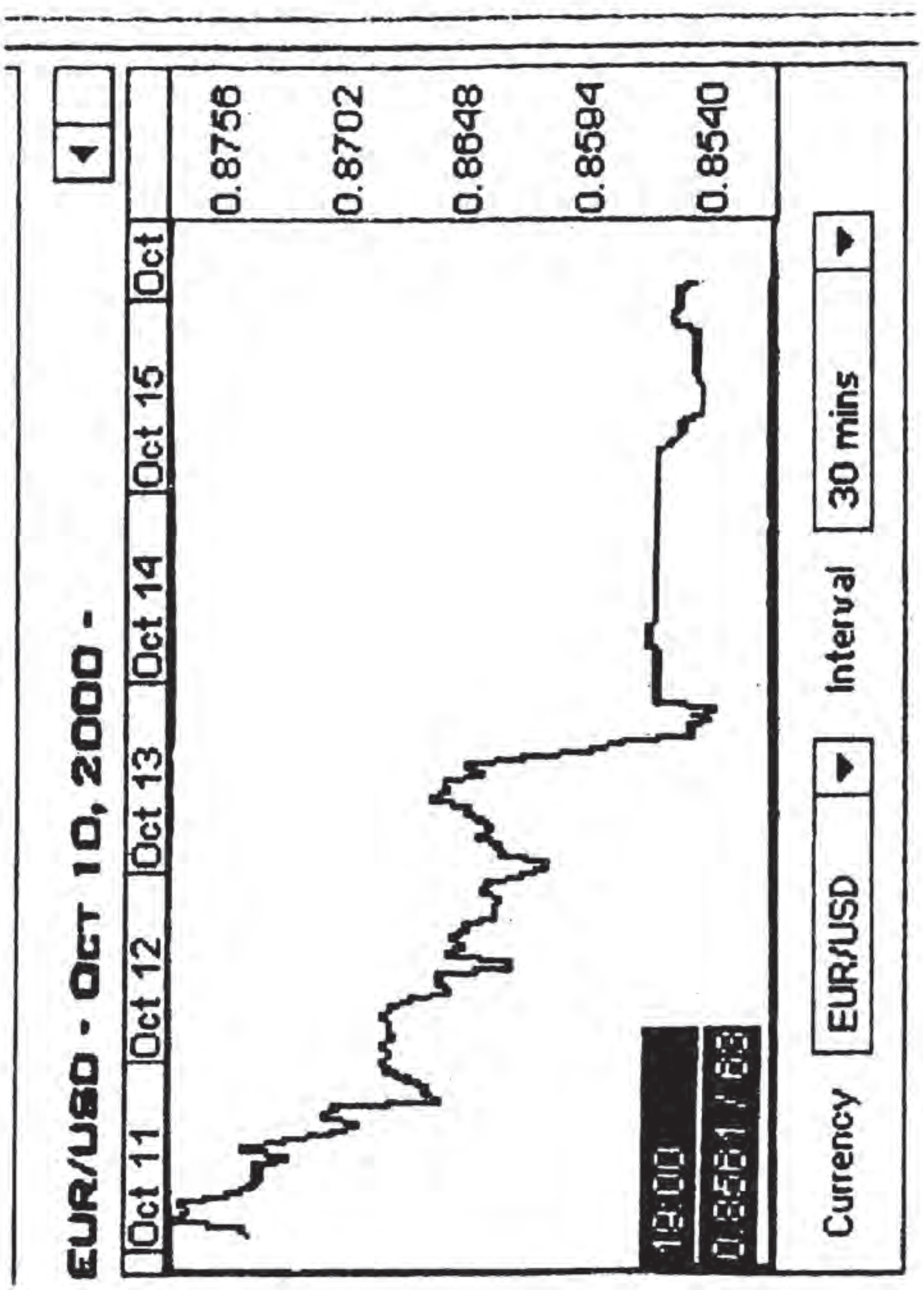

EUR/USD - Oct 10, 2000 -
Oct 11
Oct 12
Oct 13
Oct 14
Oct 15
Oct
0.8756
0.8702
0.8648
0.8594
0.8540
19:00
Currency
EUR/USD
Interval
30 mins

Fig. 12

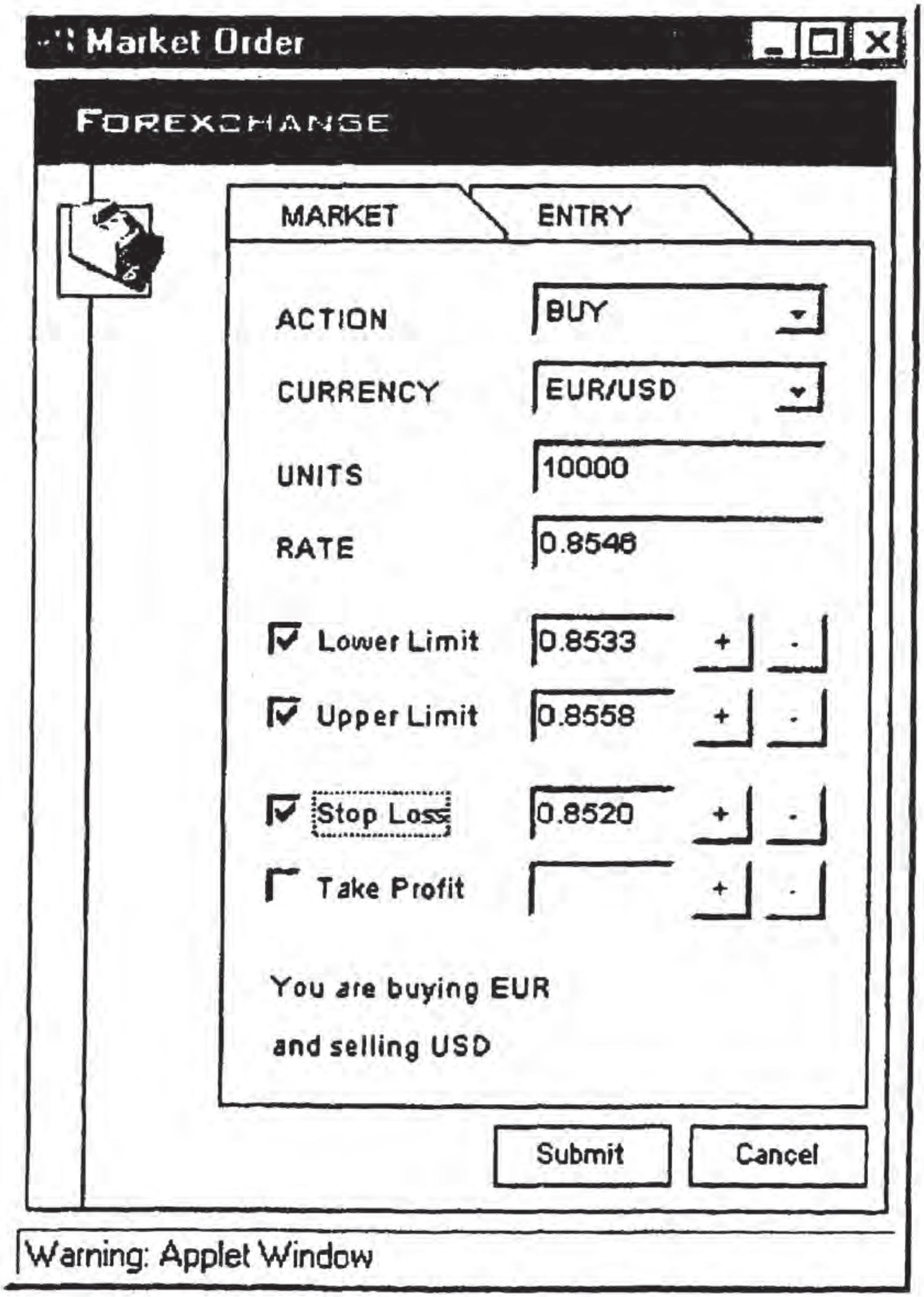

Market Order
FOREXCHANGE
MARKET
ENTRY
ACTION
BUY
CURRENCY
EUR/USD
UNITS
10000
RATE
0.8546
Lower Limit
0.8533
Upper Limit
0.8558
Stop Loss
0.8520
Take Profit
You are buying EUR
and selling USD
Submit
Cancel
Warning: Applet Window

Fig. 13

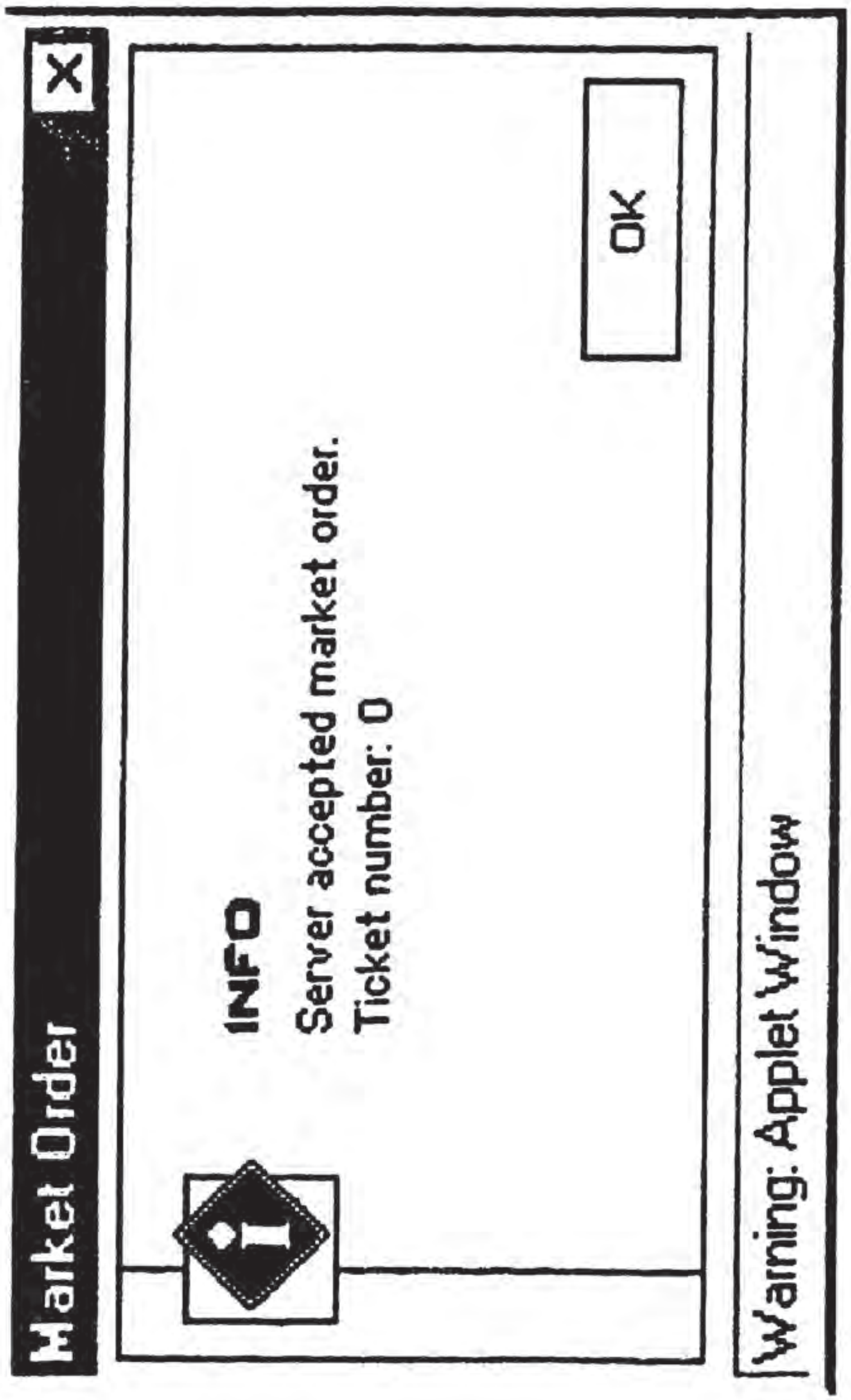

Market Order
INFO
Server accepted market order.
Ticket number: 0
OK
Warning: Applet Window

Fig. 14

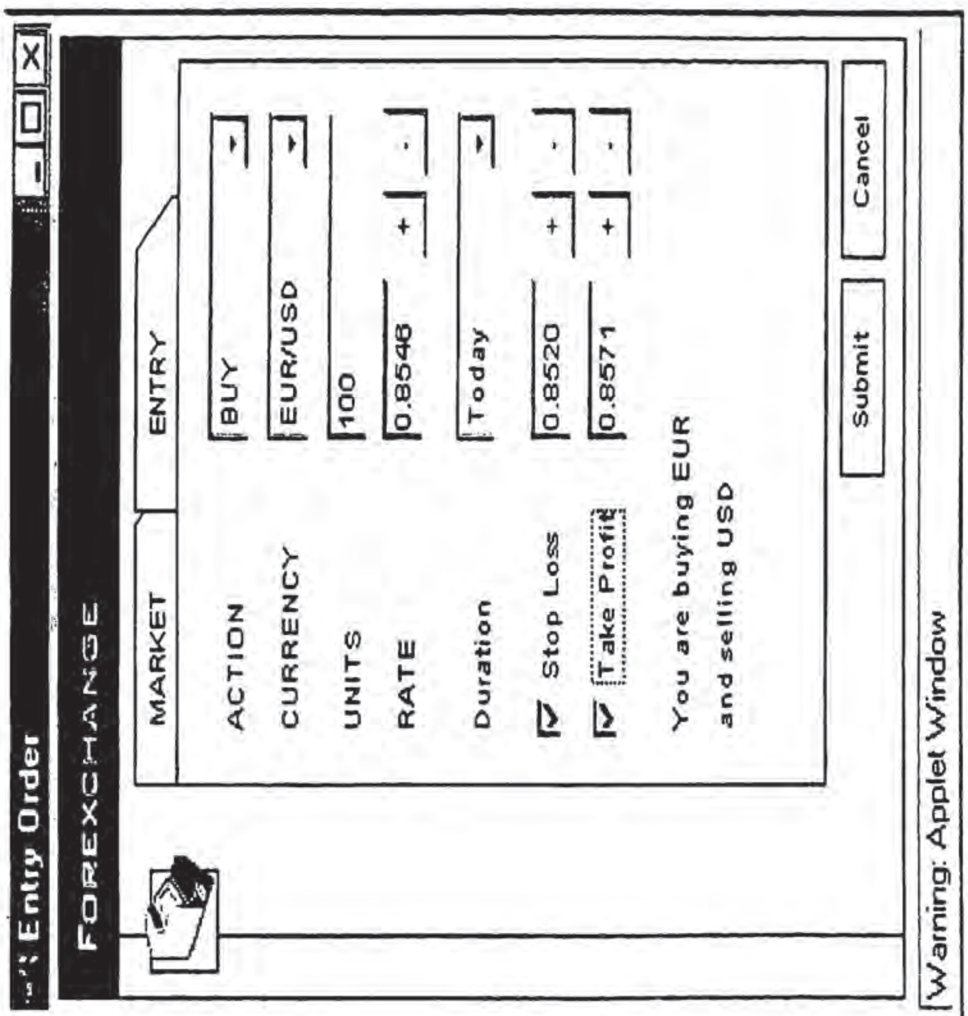

Entry Order
FOREXCHANGE
MARKET
ENTRY
ACTION
BUY
CURRENCY
EUR/USD
UNITS
100
RATE
0.8548
Duration
Today
Stop Loss
0.8520
Take Profit
0.8571
You are buying EUR
and selling USD
Submit
Cancel
Warning: Applet Window

Fig. 15

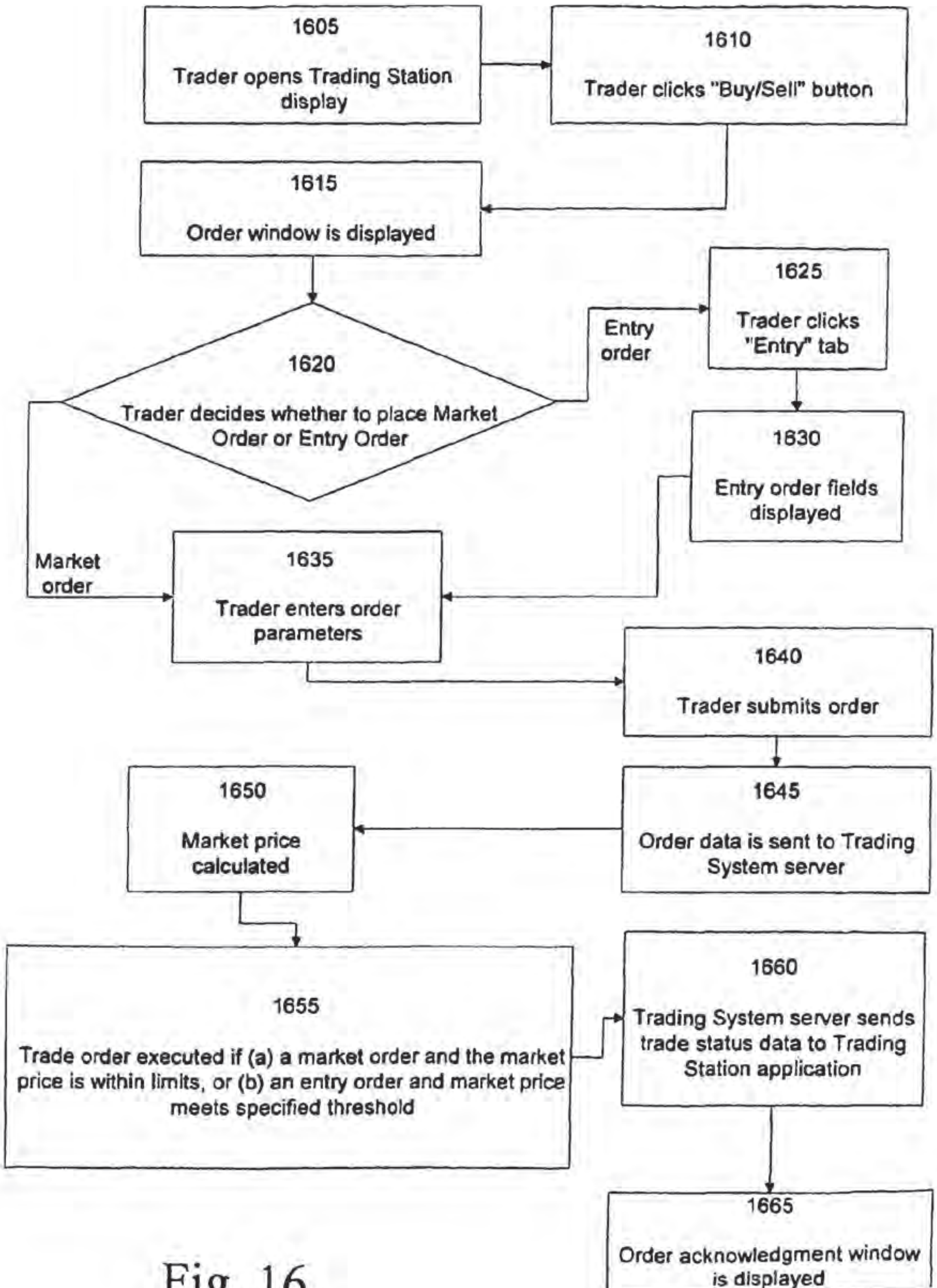

1605
Trader opens Trading Station display
1610
Trader clicks "Buy/Sell" button
1615
Order window is displayed
1620
Trader decides whether to place Market Order or Entry Order
Entry order
1625
Trader clicks "Entry" tab
1630
Entry order fields displayed
Market order
1635
Trader enters order parameters
1640
Trader submits order
1645
Order data is sent to Trading System server
1650
Market price calculated
1655
Trade order executed if (a) a market order and the market price is within limits, or (b) an entry order and market price meets specified threshold
1660
Trading System server sends trade status data to Trading Station application
1665
Order acknowledgment window is displayed

Fig. 16

CURRENCY TRADING SYSTEM, METHODS, AND SOFTWARE

CROSS-REFERENCE TO RELATED APPLICATION

The present application is a continuation of application Ser. No. 09/858,610 filed May 16, 2001 now U.S. Pat. No. 7,146,336 which claims priority to U.S. provisional application No. 60/274,174, filed Mar. 8, 2001, and incorporates the entire contents thereof herein by reference.

FIELD OF THE INVENTION

The present invention is related to currency trading; more particularly, the invention is related to trading currency over a computer network.

BACKGROUND

In a traditional on-line currency market, a trade occurs through three steps: (1) the trader specifies to a dealer the currency pair and the amount that he would to trade (but does not specify whether he would like to buy or sell); (2) the dealer specifies to the trader both a bid and an ask price and gives the trader several seconds to respond (the dealer not knowing whether the trader will buy, sell, or reject the offer); and (3) the trader either rejects the offer or specifies whether he is buying or selling (his response must occur within a time frame of a few seconds).

But performing such a three-way handshake over the Internet is somewhat impractical because of Internet delays: the trader might not actually have a few seconds to respond before the dealer withdraws the offer. Thus, there is a need for a system and method of on-line currency trading that is based on a trading model that is superior to the three-way handshake described above.

Another problem is that many corporations have firewalls that restrict access to the corporate network, and that typically restrict access to the Internet (and to well-known services such as email, the World Wide Web, etc.) from within the corporation. This inhibits the ability of on-line trading systems to access information from and transfer information to users behind corporate firewalls.

SUMMARY

The present invention overcomes the above-described and other disadvantages of previous currency trading systems and methods. In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. Each of these components is described in detail below in the Detailed Description section.

In another aspect, the present invention comprises methods for trading currency over a computer network. In one embodiment, a preferred method comprises: (a) transmitting currency market information over a computer network to an end user; (b) receiving a currency trade order from the end user, wherein the currency trade order comprises limits within which the currency trade will be acceptable to the end user;

(c) calculating a market exchange rate for the currency trade order; and (d) executing the order if the market exchange rate is within the specified limits.

In another embodiment, a preferred method comprises: (a) transmitting currency market information over a computer network to an end user; (b) receiving a currency trade order from the end user, wherein the currency trade order comprises a threshold exchange rate; (c) calculating a market exchange rate for the received currency trade order; and (d) executing the order (1) if the market exchange rate is or goes above the threshold exchange rate and the order is a sell order, and (2) if the market exchange rate is or goes below the threshold exchange rate and the order is a buy order.

In a further embodiment, a preferred method comprises: (a) receiving currency market information over a computer network from a trading system server; (b) transmitting a currency trade order to the trading system server, wherein the currency trade order comprises limits within which the currency trade will be acceptable; and (c) if a market exchange rate is within the specified limits, receiving information from the trading system server indicating that the currency trade order was executed.

In another embodiment, a preferred method comprises: (a) receiving currency market information over a computer network from a trading system server; (b) transmitting a currency trade order to the trading system server, wherein the currency trade order comprises a threshold exchange rate; and (c) if (1) the applicable market exchange rate is or goes above the threshold exchange rate and the order is a sell order, or (2) the applicable market exchange rate is or becomes below the threshold exchange rate and the order is a buy order, receiving information from the trading system server indicating that the currency trade order was executed.

In another aspect, the present invention comprises software for currency trading over a computer network. In one embodiment, preferred software comprises: (a) software for receiving data over a computer network from a trading system server; (b) software for displaying a first graphical user interface display that: (i) displays continuously updated currency exchange rates in real-time based on data received from the trading system server; and (ii) displays action buttons, including a buy/sell button; (c) software for displaying, in response to a user clicking the buy/sell action button, a buy/sell window display that: (i) comprises trade order parameter fields; and (ii) accepts trade order data entered into the trade order parameter fields by a user; and (d) software for transmitting said trade order data to said trading system server over said computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** depicts parties involved in a preferred embodiment.

FIG. **2** depicts a graphical user interface of a preferred embodiment.

FIG. **3** depicts modules of a preferred trading system server.

FIG. **4** depicts hardware components of a preferred embodiment.

FIG. **5** depicts a graphical user interface of a preferred embodiment.

FIG. **6** depicts an account summary table display.

FIG. **7** depicts an open trades table display.

FIG. **8** depicts an open positions table display.

FIG. **9** depicts an open orders table display.

FIG. **10** depicts a transactions table display.

FIG. **11** depicts a currency rates table display.

FIG. **12** depicts a currency exchange rate graph display.

FIG. **13** depicts a buy/sell pop-up window display.

FIG. **14** depicts an acknowledgment window display.

FIG. **15** depicts an entry order display.

FIG. **16** depicts steps of a method of a preferred embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The Currency Trading System of a preferred embodiment (hereinafter "Trading System") of the present invention allows traders to trade currencies over a computer network. Preferably, this computer network is the Internet, and the subsequent description herein is primarily in terms of the Internet. However, those skilled in the art will recognize that the following description also applies to other computer networks. Traders interface to the system using ordinary Web browsers running feature-rich Java applets; they obtain real-time data feeds of current exchange rates, they can analyze past exchange rates using graphical tools, they can review their current portfolio and past trades, and they can place buy and sell orders in the real-time market. Businesses interface to the system using an API. Innovative features that set the Trading System apart from the competition are: (i) extremely low spreads on the order of a few basis points, (ii) the ability to trade very small amounts as low as $1, and (iii) 24 hour a day, 7 days a week availability. This system has the potential to revolutionize the way currency trading is done and to open up currency trading to a new, large market segment of investors and speculators for whom currency trading is not feasible today. Moreover, it allows businesses to address their currency exchange requirements in the most cost-effective and efficient way.

A description of the preferred server infrastructure used in the Trading System follows. We first give a brief introduction of the system as a whole.

The Trading System involves three components (see FIG. **1**): (1) traders that are distributed around the world; (2) Trading System servers; and (3) "Partners" consisting of the financial institution(s) through which real currency exchange trades are executed, and from which real-time data feeds are obtained.

Traders communicate with Trading System servers over a secure, encrypted Internet connection to review their accounts, to monitor currency exchange market conditions, or to initiate currency exchange trades. The Trading System servers are preferably connected to the partners' back-office systems, using direct, private lines.

A trader trades with the Trading System similar to the way she currently trades with a broker, except that the trading is over the Web, occurs 24 hours a day, 7 days a week, and allows very small trades with very low spreads. Moreover, an initial deposit, which may be as low as $20, can be charged to a credit card to get started. Alternatively, the trader can transfer initial funds directly to the Partner bank to be credited to her Trading System account.

The end user interface to the Trading System is a Web page that can be displayed on any standard Java-enabled browser. The Web page (one version is shown in FIG. **2**; a second, preferred version is shown in FIG. **5**) depicts a summary of the trader's current position, recent trading activities, profit/loss performance of the portfolio, and a graphical display of recent past performance of the currencies the trader has positions in, as well as real-time exchange rates.

As discussed above, in a traditional on-line currency market a trade occurs through three steps: (1) a trader specifies a currency pair and an amount he would like to trade (and does

not specify whether he would like to buy or sell); (2) a dealer provides both a bid and an ask price to the trader and gives the trader several seconds to respond (the dealer does not know whether the trader will buy, sell, or reject the offer); and (3) the trader either rejects the offer or accepts the offer and specifies whether he is buying or selling, within the response period set by the dealer.

In the Internet domain, this type of three-way handshake is problematic. The timing constraints are difficult to implement because of frequent delays in transmission over the Internet. To overcome this problem, the present invention uses a "two-way handshake," in which: (1) a trader specifies in her trade order: (a) a currency pair; (b) a desired amount to trade; (c) whether she wishes to buy or sell; and (optionally) (d) upper and lower limits on an acceptable exchange rate; and (2) a dealer (in this case, a preferred Trading System) executes the trade using the most current "market rates" (as calculated by the system). However, the system only executes the order if the calculated market rate lies above any specified lower limit and below any specified upper limit. Note that this method does not require the use of timing constraints, and thus avoids the Internet-implementation disadvantage of previous methods.

In the present invention, trades can be initiated by the trader at the push of a button. A trading request form pops up with fields properly initialized so as to minimize the number of keystrokes required. A trader may elect to execute a trade right away, in which case the buyer of a currency will buy at the current exchange rate market offer price. Conversely, a trader can sell a currency at the current bid price. A range of automatic trading options is available, including setting bid/offer prices with a certain duration and "all-or-nothing" rules. Furthermore, the trader can limit her risks by placing stop-loss orders that are executed automatically. Similarly, she can lock in profits, by issuing take-profit orders.

All communication between the trader's browser and the Trading System server occurs through the Internet, preferably using the strongest available encryption (e.g., 128 bit keys). Moreover, the trader must authenticate herself using private passwords or certification keys obtained from certification authorities, such as Verisign or Entrust.

A request for a market trade preferably proceeds as follows: the trader, at a push of a button, obtains a trade order ticket in a popup window on the browser with key fields pre-initialized (see FIG. **13**). When the trade order is issued, again by the push of a button, a message is sent to a Trading System server, where the market price is calculated based on such factors as market data, size of the transaction, time of day, the Trading System's current exposure, and predictions on market direction. The trade order is executed using this market price. (The trader can specify limits, so that the trade occurs only if the price falls within these limits.) As such, the Trading System operates as a market maker. A message is then sent back to the trader with specific trade details, which is displayed in a popup window (see FIG. **14**) on the trader's browser together with a transaction id (for future reference). Moreover, an open orders table (see FIG. **9**) and current portfolio summary table (not shown) is updated to reflect the change.

Alternatively, the trader can issue in a similar manner an entry order (see FIG. **15**) that requests a trade be made when the currency exchange rate reaches a specified threshold. The trader may specify how long the entry order will be valid.

Referring to the attached figures, a preferred embodiment comprises a method of trading that in turn comprises the following steps (see FIG. **16**): At step **1605**, a trader desiring to trade opens a Trading Station display, and at step **1610** clicks a "Buy/Sell" button **510** on the Trading Station display (see FIG. **5**). At step **1615** an order window is displayed (see FIG. **13**). At step **1620** the trader decides whether to place a market order or an entry order. If a market order, then the trader proceeds to step **1635** and enters desired order parameters (as shown in FIG. **13**). If an entry order, then the trader proceeds to step **1625** and clicks an "Entry" tab **1320** (see FIG. **13**). At step **1630** entry order fields are displayed (see FIG. **15**). Then the trader proceeds to step **1635** and enters desired order parameters (as shown in FIG. **15**).

Once order parameters are entered at step **1635** the trader submits the order by clicking a "Submit" button **1310** (see FIG. **13**) if the order is a market order, or clicking a "Submit" button **1510** (see FIG. **15**) if the order is an entry order. At step **1645** data describing the order is sent by the Trading Station application to a Trading System server, where the data is stored. At step **1650** a current market price for the currency the trader desires to purchase is calculated. At step **1655** the trader's order is executed if (a) the trader's order is a market order and the calculated market price is within the limits set by the trader in the market order form at step **1635**; or (b) order is an entry order and the calculated market price meets the threshold(s) specified in the Entry order form at step **1635**.

At step **1660** the Trading System server sends trade status data to the trader's Trading Station application. This data includes an indication that the order has been executed, if that is the case, and at any rate includes an indication that the order has been received. At step **1665** the Trading Station application displays an order acknowledgment window (see FIG. **14**) that displays order status information.

Over time, the Trading System will accumulate an imbalance in its currency portfolio and, at times, it will need to neutralize its risk exposure to adverse currency fluctuations. The Trading System Pricing Engine can influence its exposure by setting its price quotes accordingly. Moreover, it can close out its positions periodically or take hedging positions by executing larger trades through its Partners. Preferably the Trading System's positions are managed based on state-of-the-art trading models. Preferred trading models are described in U.S. patent application no. [METHODS FOR TRADE DECISION MAKING, to Olsen et al.], filed May 14, 2001, the contents of which are incorporated herein by reference, as well as in U.S. provisional application No. 60/274,174, filed Mar. 8, 2001.

The Trading System servers preferably operate 24 hours a day, 7 days a week. These servers interface with the traders over the Internet on the one hand and on the other hand with the Partner's back-office operations. Using standard, state-of-the-art database technology, it maintains the accounts of all traders and executes trades issued by the traders. The Trading System thus plays the role of a market maker in that it internally aggregates all trades and only occasionally balances its internal positions by trading larger sums through the Partner. These larger trades are issued to the Partner in an automated way. The Trading System also takes hedging positions so as to minimize risks on the unbalanced portions of the traders' account aggregates.

Partner's Role

The Partner maintains all actual funds. It is the source and target of all fund transfers to and from customers; it maintains the aggregate accounts; and it executes all trades issued automatically by the Trading System servers. From a legal point of view, all funds must be maintained in money market instruments. Hence, the Partner will maintain a Long and a Short money market fund for each currency supported.

Overview of Currency Trading System Server Internal Design

The Trading System server architecture is designed to be:

(1) modular, in order to: (a) simplify development (time-to-market); (b) easily allow extensions and modifications; (c) ensure correctness and robustness, and (d) keep it maintainable;

(2) efficient, in order to provide fast response times to online users and minimize the computational and networking resources required to support the service;

(3) scalable, in order to support (with suitable computing infrastructure) a large number of online users and high transaction volumes; and

(4) fault resilient, so that any individual failure of a computing node or network connection does not interrupt service.

The Trading System server software preferably runs exclusively on Unix platforms, and is composed of the following modules, each with a distinct set of responsibilities (see FIG. **3**):

(1) Database (DBMS) **310**. This is the heart of the server. It keeps track of customer profile information, all customer accounts and all transactions, and ensures that atomicity, consistency, independence, and durability ("ACID") properties are maintained. The database is the reference point for all information kept by the system.

The database is preferably a standard commercially-available SQL database, configured for full replication for reliability, availability, and improved performance. The preferred embodiment is based on IBM's DB-2 product line, but Oracle, for example, could also easily be used.

(2) Server Front-end **315**. The server front-end **315** is responsible for all communication with the Web-based clients. It supports both persistent and non-persistent connections to the traders. The persistent connects are used primarily for periodic (i.e., every few seconds) transmission of the latest currency rates so that the traders can update the currency graphs and tables in real-time. Using persistent connections significantly reduces protocol processing overhead and reduces network bandwidth requirements. Non-persistent connections are used for all transaction-oriented requests, such as orders, transaction history requests, logins, etc. All transaction-oriented communication between the trader browsers and the Server Front-end occurs fully encrypted, while rate information is transmitted in unencrypted form for efficiency reasons.

Traders preferably communicate with the server using a request-response type of protocol. The Server Front-end **315** interprets each request it receives and, for each, takes appropriate action. For login requests, it sets up appropriate data structures so that all future requests can be serviced in the most efficient way. It also sets up encryption keys for secure communication, and logs the start of a new session with the Transaction Server. For rate requests, it returns the requested rates it obtains from the rate server. For orders, it executes the orders by issuing appropriate requests to the transaction server after checking the margin requirements, the availability of funds, and using rates as determined by the pricing engine. For stop-loss/take-profit and fixed-price orders (that may get executed in the future), the Trade Manager **365** is also informed. For each trade that gets executed, the Hedging Engine **340** and Margin Control **350** modules are informed, so that they always have an up-to-date snapshot of the state. For transaction history, the appropriate information is returned to the client after obtaining it from the Transaction Server **355**.

The Server Front-end **315** also encapsulates a standard Web server (a la Apache), that services other trader requests that entail formatted text; this includes all of the Help pages, large transaction history requests, and server monitoring information. The Server Front-end **315** also acts as a Firewall.

Internally, the Server Front-end **315** is structured as a set of threads that service one request after another. The threads allow concurrent request servicing so that many requests can be serviced in parallel.

(3) Rate Server/Pricing Engine **325**. The Rate Server obtains currency exchange rate information from a variety of external rate sources and stores it locally. The Pricing Engine computes the currency exchange rates that the traders see and that are used for trading. These are computed from the currency exchange rates obtained from the external rate sources, the directional movement and volatility of the market, the current Trading System exposure and a number of other parameters. The computed rates are made available to the other modules of the system, and are also stored on persistent media. Various methods of calculating such rates are known to those skilled in the art. A preferred method is described in U.S. patent application Ser. No. 09/764,366 filed Jan. 18, 2001, to Müller et al.

Traders can request historical rate data so that they can graphically display the movements of any pair of currencies. The Rate Server serves such requests and preferably has several years of currency exchange rates available for this purpose.

For fast response time, the Rate Server caches in memory all of the frequently and recently requested rates so as to minimize the number of disk accesses required.

(4) VAR (Value at Risk) server **320**. This server obtains and serves Value at Risk information. Various methods of calculating VAR are known in the art. A preferred method is disclosed in U.S. Provisional Patent Application No. 60/200,742, filed May 1, 2000, to Müller.

(5) Transaction Server **355**. This server encapsulates all transaction functionality and communicates the transactions to the Database **310** server (which runs on a separate host) after validating the transactions. The Transaction Server **355** also updates all other modules that need to be informed of new transactions. Finally, the Transaction Server **355** informs the currently online traders when a transaction (that may have been issued by a stop-loss, take-profit, or limit order daemon or by the Margin Control Manager) takes place.

(6) Interest Rate Manager **360**. The Interest Rate Manager **360** periodically (for example, every few minutes, every few seconds, or tick-by-tick) goes through all trader accounts to compute the interest rate due or owed based on the instruments currently in the portfolio, each resulting in a transaction of the trader account. The portfolio information is obtained through the Transaction Server **355**. The interest rates used are obtained from external sources, and the history of interest rates are stored on persistent storage. Because real-time (or near real-time) information is used, the Interest Rate Manager is capable of calculating, paying out, and collecting interest by the second. Interest calculation formulas are known to those skilled in the art, and any appropriate formula can be used in the Interest Rate Manager without departing from the scope of the invention. An example is the formula

$$A = P\left(1 + \frac{r}{m}\right)^{mt}$$

where P is the principal, r is the annual interest rate, t is the time (in years) over which interest is earned, m is the number of times per year that interest is compounded, and A is the amount owed (principal plus interest). The interest earned during time t is A−P. Thus, for example, if the annual interest rate is 3%, and the interest is compounded daily, then the interest I_k earned over each time period $T_k = t_k - t_{k-1}$, where each t_k is a clock-time (i.e., a particular day-hour-minute-second-fraction-of-a-second) to the nearest second (thus T_k is in seconds), is calculated according to the formula $I_k = A_k - P_k$, where P_k is the principal (the amount earning interest, not the "original" principal) at time t_{k-1}, and

$$A_k = P_k\left(1 + \frac{0.03}{365}\right)^{\frac{365T_k}{31,536,000}}.$$

Since there are 31,536,000 seconds per year,

$$\frac{T_k}{31,536,000}$$

is the time in years over which the interest is being calculated. Similar formulas can be used when t_k is given to the nearest tenth, hundredth, or other fraction of a second. If interest is compounded continuously, those skilled in the art will recognize how to apply the well-known formula $A = Pe^{rt}$ appropriately. Thus, to calculate interest on a tick-by-tick basis, the above formulas can be used, with T_k representing time between ticks.

(7) Trade Manager **365**. The Trade Manager **365** continuously checks whether a trade should be executed on behalf of a trader, and if so executes the trade by interacting with the transaction server. The Trade Manager **365** consists of multiple subcomponents: (a) a stop-loss daemon continuously checks to see whether stop-loss orders should be executed and, if so, executes them through the Transaction Server **355**; (b) a take-profit daemon continuously checks to see whether take-profit orders should be executed and, if so, executes them through the Transaction Server **355**; and (c) a limit-order daemon continuously checks to see whether a limit order should be executed and, if so, executes it through the Transaction Server **355**.

The daemons continuously monitor the current rates to determine whether action is required. Moreover, each of the daemons caches in memory all of the orders that it may need to execute. They keep the orders suitably sorted so that they can take fast action when necessary; for example, the stop-loss daemon sorts the orders in descending order of stop-loss price, the take-profit in ascending order of take-profit price.

(8) Margin Control Manager **350**. This module continuously monitors the margin requirements of all trader accounts. When necessary, the Margin Control Manager **350** will liquidate some (or all) of a trader's holdings. It caches in memory all of the information necessary for this computation, sorted in decreasing order of risk, so that it can take swift action when necessary. Holdings are liquidated through the Transaction Server **355**, when necessary.

(9) Trading System Monitor **330**. This module continuously monitors the current state of the Trading System. Among others, state parameters include: (a) current Trading System currency positions; (b) current margin situation; (c) a summary of stop-loss, take-profit, and limit orders that exist; (d) the number of currently online users; (e) the number, size and type of trades executed per second; and (f) a summary of the account positions held by the users.

This information is made available (a) to the Pricing Engine **325** (where it is-used to set the currency exchange rates made available to the traders), (b) to the Hedging Engine **340** so that it can determine when to issue trades with the Partner Bankend Bank, and (c) to system operators and Trading System financial engineers in real time via a feature-rich Web interface. Moreover, this information is logged on persistent storage for later, off-line analysis.

(10) Hedging Engine **340**. This module continuously monitors current Trading System currency positions, the positions held in the trader accounts, recent trading activity, and the market direction and volatility to determine when to issue a trade with the backend Partner Bank. Various methods of performing such calculations are known to those skilled in the art. The Hedging Engine **340** preferably uses the hedging tool described in U.S. patent application Ser. No. 09/764,366, filed Jan. 18, 2001, to Müller et al., the contents of which are incorporated herein by reference.

(11) Partner Bank Interface **335**. This module communicates directly with the backend Partner Bank to issue trades and obtain account information.

(12) Computer Systems Monitor **345**. This module continuously monitors the operation and state of the computer systems on which the Trading System is running. Besides error conditions, such metrics as memory, processor, disk, and network utilization; paging activity; the number of packets sent over the various networks; the number of transactions; and the number of processes and threads are of interest. This information is made available to system operators in real time via a feature-rich Web interface and local consoles. In addition, it is stored on persistent storage for later, off-line analysis.

The Server modules described above are structured so that they can run independently as separate processes that can be independently mapped onto an arbitrary computer within a cluster. Moreover, each of the modules can run in replicated form, providing both fault tolerance and increased throughput.

Preferred Physical Organization of the Trading System Server

A Trading System Server of a preferred embodiment runs on a hardware base consisting of a cluster of hosts and disk farms connected by networks. All of the hardware components are preferably replicated for fault tolerance, as depicted in FIG. **4**.

The cluster is connected to multiple ISPs so that if one ISP goes down, traders can still communicate with the server. The ISPs are connected to the Server through a pair of routers **410** that monitor each other; if one of them goes down, then the other will automatically take over.

For security reasons, the Database **310** is on a separate back-end network; this way, it is not connected directly to the Internet and can only be accessed by the Transaction Server **355**. The Database **310** is setup in a dual configuration, so that the system can continue operating with a single database failure. All disks are mirrored, again, so that any single disk failure will not result in a loss of data.

All of the other server processes run on a cluster of servers **420**, connected to the Internet routers **410** on the one side, and connected to the backend database **430** on the other side. A virtually unlimited number of servers can be used in the cluster, allowing the system to scale up to support a large number of users. The servers can be partitioned by functionality, allowing specialized servers to be used, optimized for the particular functionality. For example, the Rate Servers **325** need minimal CPU power, and only a limited amount of memory. They also can be replicated easily without the introduction of any complexity or overhead. Hence, smaller, less costly hardware can be used for this purpose.

After login, traders typically communicate with a particular server in "sessions" for performance reasons. Using sessions improves cache locality, resulting in far fewer database accesses, and it allows the cost of creating session encryption keys to be amortized over many communication actions. For load balancing purposes, the trader software is directed to henceforth communicate with the least loaded server at the time when the trader first logs in. In case of severe load imbalances, individual traders are redirected to new, less loaded servers. If any of the servers crashes, then the client software that was communicating with the crashed server will detect the failure and automatically (transparently to the user) go through a new login procedure.

User Interface Description

Overview

The following is a description of a preferred user interface of a preferred Trading System. The main user interface display is called a "Trading Station," and it is used for all interactions with the trading system by a trader, such as analyzing changes in currency exchange rates, reviewing the trader's current currency positions, reviewing the trader's past transactions, or issuing buy and sell requests. The key features of the Trading Station are that: (1) it runs on any of the popular Web browsers connected to the Internet; (2) it displays continuously updated currency exchange rates in real-time; (3) it displays all pertinent information in one window; and (4) all interaction with the server occurs over fully-encrypted Internet connections.

System Requirements

The User Interface is preferably implemented in Java so as to run on any browser with JDK 1.2 support, which includes all Netscape Navigators versions 4.2 and up as well as Microsoft's Internet Explorer versions 5.0 and up. The Trading Station is preferably supported for Windows 95, Windows 98, Windows 2000, Windows NT, Linux, Sun Solaris, and other Unix-based operating systems.

If operated from behind a firewall, then the Trading Station may operate properly only if the firewall allows HTTP requests to Port **90**. Many corporations have firewalls that restrict access to the corporate network to well-known services such as email. Typically this restriction is accomplished by restricting the ports that may be used. For example, Port **80** is typically used for http (Web-based) traffic. Some firewalls inspect traffic going through Port **80** to ensure that the port is being used only for Web-based traffic. This is problematic for trading systems that do not use http messages—it causes users behind corporate firewalls to be inaccessible. However, a preferred embodiment of the present invention overcomes this obstacle by prefacing Trading System messages with standard http headers to make them appear to be http requests and responses, even though they are not.

Log in Procedure

In order to log in, a trader must be a registered user. Registering is preferably free and can be accomplished by clicking on a "new users" link on a login page. Logging in requires a user to provide a user-ID and password. If a trader forgets her password, she can click on the "forgot your password" link and fill in the information requested; her password will be then be emailed to her.

If user-ID and password are entered correctly, a small window appears indicating that the Trading Station is being loaded. After a short time, a larger window appears with the Trading Station Graphical User Interface shown in FIG. **5**. Once the Trading Station is properly loaded, the contents of the small window is changed to include a number of useful links. It is important that this small window not be closed while the Trading Station is to remain in operation, although it may be minimized so as not to be in the trader's way. (This small window is necessary due to the limitations of typical Java implementations on most browsers that would otherwise not allow a trader to continue browsing the Web while the User Interface is active.)

Main Window of Trading Station

The Trading Station user interface is shown in FIG. **5**. It can be resized to a convenient size, by using the standard resizing mechanisms supported by the trader's operating system's windowing system.

The Trading Station is preferably partitioned into a number of components that each serve a different purpose:

(A) Action buttons: a vertical panel located on the left hand side of the Trading Station contains a set of action buttons that allow a user to perform the most common operations.

(B) Menus: a set of pull-down menus across the top allows a user to invoke additional operations.

(C) Account summary: an area in the middle of the Trading Station that gives a summary of the user's account.

(D) Table: an area located across the top of the Trading Station that is used to display various information in tabular format. The information displayed depends on the most recently clicked Action Button. It might display currently held instruments, current open positions, or a history of recent transactions.

(E) Currency rates: an area at the bottom left that displays various currency rates. These rates are continuously updated in real time.

(F) Graph: located at the bottom-right corner, graphs display currency rates over time. The graphs are also updated in real-time as new rates become available.

Subsequent sections describe each of these components in detail.

Action Buttons

The Trading Station preferably has the following action buttons in a panel on the left side. Clicking the appropriate button will-invoke the described operation:

Buy/Sell: Pops up a Buy/Sell window, from which a trader can issue trade requests. (See the description of the buy/sell window (FIG. **13**) for more information.)

Positions: Displays the currently open positions in a table. (See the description of the Open Positions Table (FIG. **8**) for the contents of the table.)

Trades: Displays the currently open trades in a table. (See the description of the Open Trades Table (FIG. **7**) for the contents of the table.)

Orders: Displays the open orders (that may be executed some time in the future) in a table. (See the description of the Open Orders Table (FIG. **9**) for the contents of the table.)

History: Displays a recent history of the trader's transactions in the table. (See the description of the History Table (FIG. **10**) for contents of the table.)

Analysis: Pops up a new browser window with access to a number of analysis tools that might help in making trading decisions.

News: Pops up a new browser window with the latest currency news.

Forums: Pops up a new browser window with access to a number of forums (sometimes known as newsgroups) that allow a trader to participate in discussions with other traders and currency trading experts.

Pull-Down Menus

There are preferably 5 pull-down menus (not shown), each offering different operations or services:

Connection Menu: (1) Disconnect: disconnects the Trading Station from the Trading System server. The Trading Station will remain open, but currency rates will no longer be updated, and transactions will not be possible. (2) Connect: connect the Trading Station to the server, so the trader is back on line. (3) Quit: quit and exit this application.

Account Menu: (1) Transaction history: pop up a new browser window to display an extensive list of all transactions that occurred on a trader's account. See the Transaction History section (relating to FIG. **10**) for a description of what is displayed. (2) Clear account balance and P/L: for those who have incurred large losses on their account, this operation allows a trader to start over again with a cleared P/L and new funds in the account. This feature is primarily useful when an account on the Trading System is used as a game—i.e., no real money changes hands. (3) Add funds to account: for a game account, add funds to the account; for a real-money account, transfer money into the account from the trader's credit card or obtain instructions on how to wire transfer money into the trader's account. (4) Buy/Sell: issue a trade or market order (see FIG. **13**). (5) Open positions: display the open positions in a table (see FIG. **8**). (6) Open trades: display all open trades in a table (see FIG. **7**). (7) Open orders: display all open orders in a table (see FIG. **9**). (8) Recent transaction history: display the most recent transactions in a table (see FIG. **10**).

Commands Menu: (1) Change passwords. (2) Graph: specify the currency pair to be displayed in the graph.

Information Menu: (1) Interest rates: display interest rate information in a separate browser window. (2) Market News: display up-to-date currency market news in a separate browser window. (3) Analysis tools: use an analysis tool in a separate window. (4) Forum: participate in various forums related to currency trading. (5) Rankings: see a list of the most successful currency traders using the Trading System.

Help Menu: (1) Documentation: links to descriptive documents. (2) About: display software version number and credits. (3) Debug: display debugging information in a new window.

Account Summary

The account summary display (see FIG. **6** for an example) is a small table that provides a summary of the trader's account status. It preferably shows: (1) Account Balance: the amount of the trader's cash holding in the trader's account. (2) Realized P/L: the amount of profit or loss the trader has incurred with the trader's trading activity to date. (3) Unrealized P/L: the amount of profit or loss that the trader holds with the trader's current open positions. If the trader clears all of his open positions, then this amount would be added to the Realized P/L amount. (4) Margin Used: the amount of the trader's account balance and unrealized P/L tied up for margin purposes.(5) Margin Available: the amount of the trader's account balance and unrealized P/L available as margin for new trading transactions.

This information is preferably continuously updated in real-time to take current market conditions into account. Moreover, the information is always shown in the trader's home currency.

Tables

The table area of the Trading Station shows different information depending on the last Action Button selected. If can include: (1) open trades; (2) open positions; (3) open orders; and (4) transaction history. The default is open positions.

How the information in the table is displayed can be controlled in two ways: (1) Scroll bars are used to scroll the table up or down, allowing a trader to see information that is hidden from view. (2) Sorting can be achieved by clicking on a column header, which causes the table to be sorted so that the column is in increasing or decreasing order. Clicking once sorts the column in increasing order; clicking again sorts it in decreasing order.

For all tables except Transaction history, clicking on a row of the table will cause a pop-up window to appear, offering further actions for that open trade, position, or order.

Open Trades Table

The open trades table (see FIG. **7**) shows a list of the trader's currently open trades. The table preferably has 9 columns, described from left to right (not all are depicted in FIG. **7**): (1) Short/Long: Indicates whether the position is short or long.

(2) Ticket Number: a number that uniquely identifies an open trade. A trader can use this number as a reference for inquiries to the Trading System or its operators, or to search for particular entries in the transaction history table.

(3) Currency pair: the pair of currencies involved in this trade. The first currency of the pair is referred to as the base currency, while the second one is referred to as the quote currency.

(4) Units: the number of transacted units for this trade, expressed in the base currency.

(5) S/L: the trader's stop-loss for this trade. This trade will be closed automatically as soon as the currency exchange rate for this currency pair crosses the S/L value. A stop-loss limit is used to limit the loss a trader may incur with this trade.

(6) T/P: the trader's take-profit for this trade. This trade will be closed automatically as soon as the currency exchange rate for this currency pair crosses the T/P value. A take-profit limit is used to realize the trader's profit as soon as it reaches the T/P value.

(7) Rate: the exchange rate obtained when the trade got executed.

(8) Market: the current exchange rate for this currency pair.

(9) Profit: the unrealized profit (when positive) or loss (when negative) expressed in base currency and on a per unit basis.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order. Clicking on a row with an open trade will cause a pop-up window to appear offering two operations: (1) Close trade. (2) Modify trade. This is used to modify the S/L or the T/P limits.

Open Positions Table

The Open Positions Table (see FIG. **8**) displays a list of the trader's open positions. It is similar to the Open Trades table, except that all trades of the same currency pair are aggregated into one line.

The table preferably has 6 columns, described from left to right (not all are shown in FIG. **8**):

(1) Short/Long: Indicates whether the position is short or long.

(2) Currency pair: the pair of currencies the position refers to. The first currency of the pair is referred to as the base currency, while the second one is referred to as the quote currency.

(3) Units: the number of units held in this position, expressed in the base currency.

(4) Rate: the average exchange rate obtained for the trades in this position.

(5) Market: the current exchange-rate for this currency pair.

(6) Profit: the unrealized profit (when positive) or loss (when negative) expressed in base currency and on a per unit basis.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column displayed in increasing or decreasing order. Clicking on a row with an open position will cause a pop-up window to appear offering the option to close the position.

Open Orders Table

The Open Orders Table (see FIG. **9**) shows a list of the trader's currently open orders. An open order is a request that a particular trade should be made automatically when the exchange rate of the specified currency pair crosses a specified threshold.

The table preferably has 9 columns, described from left to right (not all are shown in FIG. **9**).

(1) Short/Long: indicates whether the position is short or long.

(2) Order ID: a number that uniquely identifies the order. A trader can use this number as a reference for inquiries to the Trading System.

(3) Currency pair: the pair of currencies to be traded.

(4) Units: the number of units to be traded, expressed in the base currency.

(5) S/L: the stop-loss for this trade. This trade, once executed, will be closed automatically as soon as the currency exchange rate for this currency pair crosses the S/L value. A stop-loss limit is used to limit the loss a trader may incur with this trade.

(6) T/P: the trader's take-profit for this trade. This trade, once executed, will be closed automatically as soon as the currency exchange rate for this currency pair crosses the T/P value. A take-profit limit is used to realize the trader's profit as soon as it reaches the T/P value.

(7) Rate: specifies that the trade should be executed as soon as the exchange rate for the specified currency pair crosses this value.

(8) Market: the current exchange rate for this currency pair.

(9) Duration: specifies the amount of time an order should stand, until it is automatically canceled.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order. Clicking on a row with an order will cause a pop-up window to appear offering two operations: (1) Cancel order. (2) Modify order. This is used to modify the exchange rate threshold at which the trade is to be executed, or the S/L or T/P limits.

Transactions Table

The Transactions (or Transaction History) Table (see FIG. **10**) shows a list of the most recent transactions on the account. For access to a full list of past transactions, a user selects the Information pull-down menu and then selects Transaction History.

The Transaction History Table preferably has 6 columns, described from left to right:

(1) Transaction ID: uniquely identifies the transaction.

(2) Type: identifies the type of transaction.

(3) Currency pair the pair of currencies associated with the transaction.

(4) Units: the number of units to traded in the transaction, expressed in the base currency.

(5) Price: the currency exchange rate applied when buying or selling a currency pair.

(6) Date/Time: the date and time of the transaction.

Clicking on the scroll buttons. will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order.

Currency Rates

The Currency Rates Table (see FIG. **11**) shows the current exchange rate for the currency pairs supported by the Trading System. They are preferably updated in real time, approximately every 5 seconds. When there is a significant exchange rate movement for a currency pair, up/down indicators show the direction of the rate change in order to alert a trader, should a trader not currently have the currency pair displayed in the graph.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a currency pair's ask price will pop up a buy window for that currency pair. Clicking on a currency pair's bid price will pop up a sell window for that currency pair.

Graphs

Graphs (see FIG. **12**) show how currency exchange rates change over a period of time, ranging from minutes to months. All graphs are updated in real-time, as new currency rates arrive.

At any given time, the difference between the lower boundary and the upper boundary of the curve represents the difference between the bid and the ask price, and the difference may vary over time depending on market conditions. Thus, the top part of the curve indicates the ask price, and the bottom of the curve indicates the bid price.

As a mouse cursor **1220** is moved over the graph, a sub-area **1230** in the graph shows precise exchange rate information for the target currency pair corresponding to the time instance represented by the position of the mouse cursor.

The graph may also display Buy or Sell widgets that indicate at which point in time a trader bought or sold a currency pair. Downward pointing red arrows indicate a sold currency pair (where a trader is hoping the rates will go down after that point), and upward pointing green arrows indicate a bought currency pair (where a trader is hoping the rates will go up after that point).

A trader can adjust what is shown in the graph: (1) The currency pair displayed is selected using the pull-down menu **1240** at the bottom left. (2) The granularity of the graph is selected using the pull-down menu **1250** at the bottom right of the graph. Selecting a fine granularity, such as 5 seconds (where each point on the horizontal axis represents 5 seconds of time), will display a relatively short time interval (less than an hour, in this case). Selecting a larger granularity, such as one day, will display longer-term trends (9 months of exchange rate information in this case).

Scroll buttons **1260** at the top right of the graph area allow a trader to shift the time interval shown to the left or to the right (backward or forward in time). Clicking on the graph with the mouse will hide the Buy/Sell widgets. Clicking again will cause them to reappear.

Buy/Sell Window

A Buy/Sell pop-up window (see FIG. **13**) allows a trader to issue buy or sell orders. The window can be caused to pop up either by: (1) clicking on the Buy/Sell action button (see FIG. **5**); (2) clicking on the bid or ask price in the Currency Rates Table (see FIG. **11**); or (3) clicking on an existing trade, position, or order in the Table area of the Trading Station display (see FIG. **5**).

Two types of orders are supported: (1) Market Orders are orders that are transacted immediately based on market exchange rates. (2) Entry Orders are orders that are executed when the exchange rate crosses a certain threshold.

The type of order can be selected by clicking on the appropriate tab in the Buy/Sell Window (see FIG. **13**). Market order comes up as the default order.

Issuing a Market Order. To issue a market order with the Buy/Sell Window and the Market Tab selected, a number of fields must be filled out (although most of the fields are pre-initialized with reasonable values):

(1) ACTION: choose between buy and sell.

(2) CURRENCY: choose the currency pair the trader wishes to buy or sell. By default, this field will be initialized as follows: (A) If the Buy/Sell button was used to obtain the window, the currency pair currently shown in the graph. (B) If the bid or ask price was clicked to obtain the window, the currency pair for which the price was clicked. (C) If a trade or position was clicked in the Table area, the currency pair corresponding to the trade or position. The pull-down menu can be used to select another currency pair.

(3) UNITS: the number of units of the currency pair the trader wishes to buy or sell, with units expressed in terms of the base currency.

(4) Lower Limit: the order will result in a trade only if a price is obtained that does not lie below this limit. By default, no limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(5) Upper Limit: the order will result in a trade only if a price is obtained that does not lie above this limit. By default, no limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(6) Stop Loss: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no stop-loss limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(7) Take Profit: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no take profit limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

The RATE field is set by the Trading Station and corresponds to the most recent exchange rate for the selected currency pair.

To issue the order, a Submit button **1310** must be selected. If the order is successful, and a trade occurs, then an acknowledgment window (see FIG. **14**) pops up with a Ticket number that can be used for future reference. Moreover, the Open Trades Table (see FIG. **7**) will be updated to reflect the new trade, as will the Open positions-ions Table (see FIG. **8**) and the Transaction History Table (see FIG. **10**).

Several issues are important to note:

(1) If an order is successful and a trade occurs, then the exchange rate obtained for the trade will correspond to the most current exchange rate maintained at the Trading System servers and not necessarily the rate displayed in the Buy/Sell window.

(2) An order without Lower and Upper Limits will always result in a trade.

(3) An order with both Lower and Upper Limits will result in a trade if and only if the exchange rate for the potential trade lies between the two limits.

Issuing an Entry Order. To issue an entry order with the Buy/Sell Window and the Entry Tab selected (see FIG. **15**), a number of fields must be filled out (although most of the fields are pre-initialized with reasonable values):

(1) ACTION: choose between buy and sell.

(2) CURRENCY: choose the currency pair the trader wishes to buy or sell. By default, this field will be initialized as follows: (A) if the Buy/Sell button was used to obtain the window, the currency pair currently shown in the graph; (B) if the bid or ask price was clicked to obtain the window, the currency pair for which the price was clicked. The pull-down menu can be used to select another currency pair.

(3) UNITS: the number of units of the currency pair the trader wishes to buy or sell, with units expressed in terms of the base currency.

(4) RATE: the order will result in a trade as soon as the exchange rate for the selected currency pair crosses the given value; that is, for buy orders, if the rate goes below this value, and for sell orders if the rate goes above the given value

(5) Duration: this value is used to limit the amount of time an outstanding order will remain effective. By default, the order remains effective indefinitely. However, the duration can be set to the end of the day or for an hour.

(6) Stop Loss: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no stop-loss limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(7) Take Profit: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no take profit limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

To issue the order, a Submit button **1510** must be selected. This results in an acknowledgment window (see FIG. **14**) popping up with a Ticket number that can be used for future reference. Moreover, the Open Orders Table (see FIG. **9**) will be updated to reflect the new order. Note that a trader can modify the parameters of an open order (including the rate representing the trade threshold, or the S/L and T/P) by clicking on the order in the Open Orders Table.

What is claimed is:

1. A method of trading currencies over a computer network connecting a trading system server and at least one trading client system, comprising the steps of:

(i) at the trading system server, determining and dynamically maintaining a plurality of current exchange rates, each current exchange rate relating to a pair of currencies and including a first price to buy a first currency of the pair with respect to a second currency of the pair and a second price to sell the first currency of the pair with respect to the second currency of the pair;

(ii) transmitting data from the trading system server to a trading client system, the transmitted data representing at least one current exchange rate at the time of the transmission;

(iii) at the trading client system, displaying the first and second prices for each received current exchange rate to a user;

(iv) at the trading client system, accepting input from the user identifying a pair of currencies the user desires to trade, an amount of at least one currency of the pair desired to be traded and a requested trade price at which it is desired to effect the trade;

(v) transmitting the accepted input from the trading client system to the trading system server;

(vi) at the trading system server, comparing the requested trade price to the respective first price or second price of the corresponding current exchange rate at that time and, if the respective first price or second price of the corresponding current exchange rate at that time is equal to or better than the requested trade price, effecting the trade at the corresponding respective current exchange rate first price or second price and if the corresponding current exchange rate is worse than the requested trade price, refusing the trade; and

(vii) transmitting from the trading system server to the trading client system an indication of whether the trade was refused or transacted and, if transacted, an indication of the price the trade was transacted at.

2. The method of claim **1** wherein the requested trade price is derived from a respective one of the first price or second price of the received current exchange rate and a user input limit value defining a maximum acceptable difference between the respective one of the first price or second price of the received current exchange rate received at the trading client system and the respective one of the first price or second price of the corresponding current exchange rate determined at the trading client system at which the trade can be effected.

3. The method of claim **2** wherein the user can input a first limit value to define a maximum acceptable difference between the first price of the current exchange rate received at the trading client system and the first price of the corresponding current exchange rate determined at the trading client system and can input a second limit value to define a maximum acceptable difference between the second price of the current exchange rate received at the trading system and the second price of the corresponding current exchange rate determined at the trading client system and the requested trade price is derived from the first price or second price of the current exchange rate received at the trading client system and the corresponding one of the first limit value and second limit value.

4. The method of claim **2** wherein step (iv) comprises the steps of:

(a) the user selecting one of the first price and second price of the current exchange rate displayed at the trading client system;

(b) displaying to the user a set of input fields to define a desired trade, the input fields including an identification of the pair of currencies the user desires to trade, the amount of the currencies desired to be traded, the selected first price or second price of the current exchange rate received at the trading client system and a limit value, and where the input fields to identify the pair of currencies and the first price or second price are populated with appropriate values determined from the user's selection of the one of the first price or second price;

(c) receiving from the user input to the input field defining the desired amount of currency to be traded; and

(d) determining the requested trade price from the selected one of the first price and second price and the limit value.

5. The method of claim 4 wherein in step (b) the displayed set of input fields includes: a first limit value to define a maximum acceptable difference between the first price of the current exchange rate received at the trading client system and the first price of the corresponding current exchange rate determined at the trading client system; and a second limit value to define a maximum acceptable difference between the second price of the current exchange rate received at the trading system and the second price of the corresponding current exchange rate determined at the trading client system and in step (d) the requested trade price is derived from the selected first price or second price and the corresponding one of the first limit value and second limit value.

6. The method of claim 2 wherein, when the limit value is zero, the requested trade price is the current corresponding first price or second price of the current exchange rate at the trading server.

7. A method of trading currencies over a computer network connecting a trading system server and at least one trading client system, comprising the steps of:

(i) at the trading system server, determining and dynamically maintaining a plurality of current exchange rates, each current exchange rate relating to a pair of currencies and including a first price to buy a first currency of the pair with respect to a second currency of the pair and

a second price to sell the first currency of the pair with respect to the second currency of the pair;

(ii) transmitting data from the trading system server to a trading client system, the transmitted data representing at least one current exchange rate at the time of the transmission;

(iii) receiving at the trading system server input from a user of the trading client system identifying a pair of currencies the user desires to trade, an amount of at least one currency of the pair desired to be traded and a requested trade price at which it is desired to effect the trade;

(iv) at the trading system server, comparing the requested trade price to the respective first price or second price of the corresponding current exchange rate at that time and, if the respective first price or second price of the corresponding current exchange rate at that time is equal to or better than the requested trade price, effecting the trade at the corresponding respective current exchange rate first price or second price and if the corresponding current exchange rate is worse than the requested trade price, refusing the trade; and

(v) transmitting from the trading system server to the trading client system an indication of whether the trade was refused or transacted and, if transacted, an indication of the price the trade was transacted at.

* * * * *

EXHIBIT C

03-09-01 A|PROV

Express Mail No.: EL 501 637 910 US

COVER SHEET FOR PROVISIONAL APPLICATION FOR PATENT

Assistant Commissioner for Patents
Box PROVISIONAL PATENT APPLICATION
Washington, DC 20231

Sir:

This is a request for filing a PROVISIONAL APPLICATION under 37 CFR 1.53(c).

Docket Number	10366-021-888	Type a plus sign (+) inside this box →	+	

INVENTOR(s) APPLICANT(s)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
Dacorogna	Michel	M.	Zurich, Switzerland
Gençay	Ramazan		Windsor, Ontario, Canada
Müller	Ulrich	A	Zurich, Switzerland
Olsen	Richard	B.	Zurich, Switzerland
Pictet	Olivier	V.	Zurich, Switzerland

TITLE OF THE INVENTION (280 characters max)

High-Frequency Finance Methods

CORRESPONDENCE ADDRESS:

PENNIE & EDMONDS LLP
1155 Avenue of the Americas
New York, NY 10036-2711
(212) 790-9090

ENCLOSED APPLICATION PARTS (*check all that apply*)

☒ Specification	*Number of Pages*	402	☐ Small Entity Statement
☐ Drawing(s)s	*Number of Sheets*		☐ Other (specify)

METHOD OF PAYMENT (*check one*)

☐ A check or money order is enclosed to cover the Provisional filing fees.

☒ The Commissioner is hereby authorized to charge the required filing fee to Deposit Account Number 16-1150.

ESTIMATED PROVISIONAL FILING FEE AMOUNT: ☒ $150 ☐ $ 75

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.
☒ No. ☐ Yes, the name of the U.S. Government agency and the Government contract number are: ______

Respectfully submitted,
Steven J. Underwood (Reg No. 47,205)

Signature For Francis E. Morris
Francis E. Morris
PENNIE & EDMONDS LLP

REGISTRATION NO (*if appropriate*): 24,615

Date: March 8, 2001

☐ Additional inventors are being named on separately numbered sheets attached hereto

Total number of cover sheet pages: 1

PROVISIONAL APPLICATION FILING ONLY

03/08/01 JC912 U.S. PTO 60/274174 03/08/01

NY2 - 1177882.1

AN INTRODUCTION TO HIGH-FREQUENCY FINANCE

Michel M. Dacorogna
Zurich Re, Switzerland

Ramazan Gençay
University of Windsor, Canada
Olsen & Associates, Switzerland

Ulrich A. Müller
Olsen & Associates, Switzerland

Richard B. Olsen
Olsen & Associates, Switzerland

Olivier V. Pictet
Dynamic Asset Management, Switzerland

CONTENTS

2
MARKETS AND DATA

3
TIME SERIES OF INTEREST

4

ADAPTIVE DATA CLEANING

5
BASIC STYLIZED FACTS

6
MODELING SEASONAL VOLATILITY

7
REALIZED VOLATILITY DYNAMICS

8
VOLATILITY PROCESSES

9
FORECASTING RISK AND RETURN

10
CORRELATION AND MULTIVARIATE RISK

11
TRADING MODELS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This book presents a unified view of high-frequency time series methods with a particular emphasis on foreign exchange markets as well as interest rate spot and futures markets. The scope of this book is also applicable to other markets, such as equity and commodity markets.

As the archetype of financial markets, the foreign exchange market is the largest financial market worldwide. It involves dealers in different geographic locations, time zones, and working hours who have different time horizons, home currencies, information access, transaction costs, and other institutional constraints. The time horizons vary from intraday dealers, who close their positions every evening, to long-term investors and central banks. In this highly complex and heterogeneous market structure, the market participants are faced with different constraints and use different strategies to reach their financial goals, such as by maximizing their profits or maximizing their utility function after adjusting for market risk.

This book provides a framework to the analysis, modeling, and inference of high-frequency financial time series. It begins with the elementary foundations and definitions needed for studying the fundamental properties of high-frequency financial time series. It extends into the adaptive data-cleaning issues, treatment of seasonal volatility, and modeling of intraday volatility. Fractal properties of the high-frequency financial time series are found and explored, and an intrinsic time is used to construct forecasting models. The book provides a detailed study of how the adopted framework can be effectively utilized to build econometric models of the price-formation process. Going beyond the price-formation process, the book presents the techniques to construct real-time trading models for financial assets.

It is designed for those who might be starting research in the area as well as for those who are interested in appreciating the statistical and econometric theory that underlies high-frequency financial time series modeling. The targeted audience includes finance professionals, including risk managers and research professionals in the public and private sectors; those taking graduate courses in finance, economics, econometrics, statistics, and time series analysis; and advanced MBA students. Because the high-frequency finance field is relatively new and the literature is scattered in a wide range of academic and nonacademic platforms, this book aims to provide a uniform treatment of the field and an easily accessible platform to high-frequency financial time series analysis — an exciting new field of research.

With the development of this field, a huge new area of research has been initiated, where work has hardly started. This work could not be more fascinating, and a number of discoveries are waiting to be made. We expect research to increase in this field, as people start to understand how these insights can dramatically improve risk-adjusted performances in asset management, market making, and treasury functions and be the foundation for other applications, such as an early warning system of financial markets.

Michel M. Dacorogna
Ramazan Gençay
Ulrich A. Müller
Richard B. Olsen
Olivier V. Pictet

ACKNOWLEDGMENTS

We should start by acknowledging that 15 years ago, when our research team at Olsen & Associates (O&A) first began using the amazing magnifying glass provided by high-frequency data to see if we could uncover possible patterns in the financial markets, none of us anticipated just how expansive the effort would become.

With the publication of this book originating from our work, sincere thanks are due to so many that we can only hope we have recognized most of the colleagues and friends who have advanced our work. Their help, their encouragement, their criticism, and their friendship have contributed to the style of teamwork we always favored.

We begin with Matthias Schwarz, a biology student who computed the first scaling law working with us in the autumn of 1986. Our first academic visitor was Claude Morgenegg, coming from the University of Geneva, who taught our group of physicists the right language to use to reach the economists. Thanks also to Casper de Vries, who opened up for us the world of extreme value theory; Cindy L. Gauveau, who prepared forecasting models for foreign exchange rates and also brought the economic touch to our work; Rakhal Davé, for his explorations of the LeBaron effect; Marco Tomassini and Bastien Chopard, who brought to our attention the genetic algorithms; Mark Lundin and his correlation studies; Gennady Samorodnitsky and Paul Embrechts, who were able to prove the sufficiency of the stationarity condition of HARCH processes; Giuseppe Ballocchi, who led us into the research on interest rate futures; and Wolfgang

Breymann, who has extended the ϑ-time concept and developed the idea of a heterogeneous market in his cascade model. A particular thanks goes to Gilles Zumbach, who has contributed many graphs to this book. He continues the work, bringing it to new levels and uncovering many more properties with the powerful operator framework and the software tools in C++ that he has developed over the years.

We also want to thank our colleagues who joined us at a later stage and have already reached out for other adventures: Lars Jaeger, Thomas Domenig, Peter Rice, and Hoss Hauksson. Our thanks extend also to Jørgen Olsen, whose wisdom and vast scientific culture has enlightened our seminars and whose road map for building a Richter scale for financial markets we implemented.

One very important and enriching experience has been the visits of many students who spent time with us and brought along their enthusiasm and eagerness to learn: Dominique Guillaume; Lukas Pulver; Petra Korndoerfer; Markus P. Herrchen; Jens Richelsen; Christian Jost; Jürg S. Füssler; Retus G. Sgier; Alexander Dimai; Jonathan Dawes; Jakob E. von Weizsäcker; Philipp Hartmann; Cătălin Stărică; Barbara Piccinato; Carl Hopman; Peter Rice, who later joined our research team; Simone Deparis; Fulvio Corsi; and Paul Lynch. Without them, we would never have been able to explore so many different time series and to accomplish so many studies.

Many of our academic friends around the world visited us and understood early on the interest in research on this type of data. They provided us with encouragement to continue and the sense that we were working in the right direction. Hermann Garbers was the first to invite us to give a seminar at the University of Zurich, where we presented the scaling law in December 1988. From Benoît Mandelbrot in 1989 to Gennady Samorodnitsky just prior to the publication of this book, we have been fortunate to share time and work at O&A with some fine scientists: Tim Bollerslev, William Brock, Hans Bühlmann, Peter Bühlmann, Frank K. Diebold, Christian Dunis, Rüdiger Frey, Hélyette Geman, Charles Goodhart, Rudolf Kalman, Hans Rudolf Lerche, Bruce Mizrach, John Moody, Salih Neftçi, Wolfgang Polasek, Remo Schnidrig, Albert N. Shiryaev, Gerhard Stahl, Massimo Tivegna, Murad Taqqu, Walter Wasserfallen, Andreas Weigend, and Diethelm Würtz. We would like to thank especially Charles Goodhart, whose support and insights led to the O&A "High-Frequency Data in Finance" conferences, but also Richard Baillie, Tim Bollerslev, Rob Engle, Joel Hasbrouck, Michael Melvin, and Maureen O'Hara. Gerhard Stahl has been a great partner in exploring new issues of risk management. His scientific rigor was always refreshing in this field where ad hoc arguments often dominate. Michel Dacorogna would like to thank particularly Blake LeBaron for the many e-mail exchanges we have had over the years on our research. They were always stimulating and encouraged us to think deeper into the problems and find the connections with the traditional economic approach. Manfred Härter and Mico Loretan have been always supportive of our work and have brought to us many ideas and opportunities for presenting it. Ramo Gençay would like to thank William Brock, Dee Dechert, Murray Frank, Blake LeBaron, and

Thanasis Stengos for many exciting research conversations and Michael Charette, Ron Meng and Tibor Toronyi for research support. Ulrich Müller would like to thank Günter Schwarz for his contribution to our understanding of portfolio theory.

It is also clear that without the help and the dedication of our software team we would not have been able to access a database of such quantity and quality, covering more than 14 years of tick-by-tick prices. From Rob J. Nagler to Kris Meissner through J. Robert Ward, William H. Kelly, Daniel P. Smith, Martin Lichtin, Devon Bowen, and Michael Stumm, we learned the subtleties of object-oriented programming and have enjoyed their constant support in our efforts to make sense of all that we were seeing. Paul Breslaw has been so helpful with the data and improving our English. Our thanks go also to our friends from the operation group who kept alive our system, especially Jorge Mota, Jeff Courtade, and Gary Swofford. Whenever we had a problem with bulbs going out of function or air conditioning not working (especially when it was needed most), Filippo Guglielmo would always be here to solve it.

The trading model developments would not have been so interesting, nor so close to reality, without the contribution of our help desk: Pius Dall'Acqua, Stephan Schlatter, and last, but not least, Bernard Hechinger and his deep knowledge of the microstructure of financial markets. His interest in our models has brought us to rethink many aspects of their strategies and implement some of his ideas. In terms of market knowledge, we especially want to thank Dean LeBaron, whose vast experience and enthusiasm for new developments is a source of inspiration and encouragement. Our customers also brought many ideas to us, especially in the group who participated in the development of the interest rate project: Michael Brockmann, Dieter Heitkamp, Luciano Steve, and Giuseppe Ciliberto. We always enjoyed the exchanges with the practitioners who understood the need for a scientific approach to the markets. Another example of this fertile interaction was with Monique Donders, Tjark Tjin, and Marcel Vernooy during our project of building a currency overlay product based on trading models. Special thanks go also to Daniel Huber, who opened up so many doors for us.

There is no question that we benefited greatly from the structure and organization provided by our different administrative assistants over the years. Karin Jost, Rosemarie Arnold-Becker, and Melanie Käslin all brought a strong sense of service and dedication that made our teamwork possible.

The process of writing a book with many authors is complex and demanding but also very rewarding because it gave us the occasion to discuss, deepen our understanding of the matters and interact with interesting people. In this process, Scott Bentley's (the senior editor of Academic Press) help and feedback have been important for keeping the level of motivation high and for the success of this project. The care of Amy Hendrickson for many LaTeX formatting problems of a book of more than 400 pages and containing so many figures was essential for the resulting appearance of this book.

Before closing this page of gratitude, we do not want to forget Dina Weidmann and Elisa Guglielmo, who cooked so many fine dishes with the Italian touch and make O&A's famous "Friday family lunches" a genuine gourmet experience. Faced with mountains of data to unravel, this lovely tradition warmed the soul. Grazie.

Michel M. Dacorogna
Ramazan Gençay
Ulrich A. Müller
Richard B. Olsen
Olivier V. Pictet

1

INTRODUCTION

1.1 MARKETS: THE SOURCE OF HIGH-FREQUENCY DATA

A famous climber, when asked why he was willing to put his life in danger to climb dangerous summits, answered: "Because they are there." We would be tempted to give the same answer when people ask us why we take so much pain in dealing with high-frequency data. The reason is simple: financial markets are the source of high-frequency data. The original form of market prices is tick-by-tick data: each "tick" is one logical unit of information, like a quote or a transaction price (see Section 2.1). By nature these data are irregularly spaced in time. Liquid markets generate hundreds or thousands of ticks per business day. Data vendors like Reuters transmit more than 275,000 prices per day for foreign exchange spot rates alone.

Thus high-frequency data should be the primary object of research for those who are interested in understanding financial markets. Especially so, because practitioners determine their trading decisions by observing high-frequency or tick-by-tick data. Yet most of the studies published in the financial literature deal with low-frequency, regularly spaced data. There are two main reasons for this. First, it is still rather costly and time-consuming to collect, collate, store, retrieve, and manipulate high-frequency data. That is why most of the available financial data are at daily or lower frequency. The second reason is somehow more subtle but still quite important: most of the statistical apparatus has been developed and thought for homogeneous (i.e., equally spaced in time) time series. There is little work done to adapt the methods to data that arrive at random time intervals. Unfortunately in finance, regularly spaced data are not original data but artifacts derived from the original market prices. Nowadays with the development of computer technology, data availability is becoming less and less of a problem. For instance, most of the exchanges and especially those that trade electronically would gladly provide tick-by-tick data to interested parties. Data vendors have themselves improved their data structures and provide their users with tools to collect data for over-the-counter (OTC) markets. Slowly, high-frequency data are becoming a fantastic experimental bench for understanding market microstructure and more generally for analyzing financial markets.

That leaves the researcher with the problems of dealing with such vast amounts of data using the right mathematical tools and models. This is precisely the subject of this book.

1.2 METHODOLOGY OF HIGH-FREQUENCY RESEARCH

From the beginning, our approach has been to apply the experimental method which has been highly successful in "hard" sciences.[1] It consists of three steps, the first one being to explore the data in order to discover the fundamental statistical properties they exhibit with a minimum set of assumptions. This is often called finding the "stylized facts" in the econometric or finance literature. This first step was in fact not so important in the economic literature, because the sparseness of data made it either relatively simple or uninteresting due to the statistical uncertainty.

The second step is to use all of these empirical facts to formulate adequate models. By adequate models, we do not mean models that come from hand-waving arguments about the markets, but rather models that are directly inspired by the empirical regularities encountered in the data. It is the point where our understanding of market behavior and reality of the data properties should meet. There have been many debates between the time series approach and microstructure approach. The first one relying more on modeling the statistical properties of the data and the latter concentrating on modeling market behavior. Both approaches have their value and high-frequency data might be able to reconcile them by enabling us to actually test the microstructure models, Hasbrouck (1998); Rydberg and Shephard (1998).

The third step, of course, is to verify whether these models satisfactorily reproduce the stylized facts found in the data. The ultimate goal is not only a good descriptive model but the ability to produce reasonable *predictions* of future movements or risks and to integrate these tools into practical applications, such

[1] We refer here to experimental sciences such as physics, chemistry, or biology.

as risk management tools or option pricing algorithms. For decades, practitioners have been developing so-called technical analysis, which is a kind of empirical time series analysis based on rudimentary analytical tools. Although some new academic research has analyzed these trading rules,[2] they remain controversial and are looked down upon. We hope that this book will put on a new footing many ideas that have been developed in technical analysis.

We have organized this book along the same lines, we first present the empirical regularities, then we construct models, and lastly we test their power to predict market outcomes.

The novelty of high-frequency data demands to take such an approach. This was not usual in econometrics because so little data were available until the late 1980s. It was quite natural that the researcher's emphasis was to make sure that the methodology was correct in order to obtain the most information out of the sparse data that were available. Only recently the research community in this field has recognized the importance of the first step: finding empirical facts. This step can already be good research in its own right. A good example is the recent paper by Andersen *et al.* (2001), where the authors explore in detail the distributional properties of volatility computed from high-frequency data.

Thanks to the development of electronic trading and the existence of various data providers also on the Internet, it is now possible to follow the price formation in real-time. Ideally, the analysis and modeling of the price-generation process should, in real-time, produce results that add value to the raw data. There is strong demand from the market to have, next to the current price, a good assessment of the current risk of the financial asset as well as a reasonable prediction of its future movement. This means that the models should be made amenable to real-time computations and updates. Techniques for doing so will be presented in the remainder of the book. It is possible to develop methods that allow for the easy computation of models and can thus provide almost instantaneous reaction to market events. Although quite popular among practitioners who want to analyze the past developments of prices, those techniques have had little echo, until now, in the academic world. Very few research papers have studied the statistical foundations and properties of those "technical indicators." In this book (Chapter 3) we provide a unified platform for these methods.

1.3 DATA FREQUENCY AND MARKET INFORMATION

Relating the type of data available for researchers, the effects and the models that are discovered and developed with these different samples, provides insight into the development of research in finance. Figure 1.1 illustrates the sample size versus the measurement frequency of some well-known data sets used in finance. The

[2] Among others, here is a list of interesting papers on the issue of technical trading models. Neftci (1991), Brock *et al.* (1992), Taylor and Allen (1992), Levich and Thomas (1993b), Gençay and Stengos (1998), Gençay (1998a,b), Frances and van Griensven (1998), Allen and Karjalainen (1999), Gençay (1999), LeBaron (1999a), Sullivan *et al.* (1999), and Gençay *et al.* (2001c, 2002).

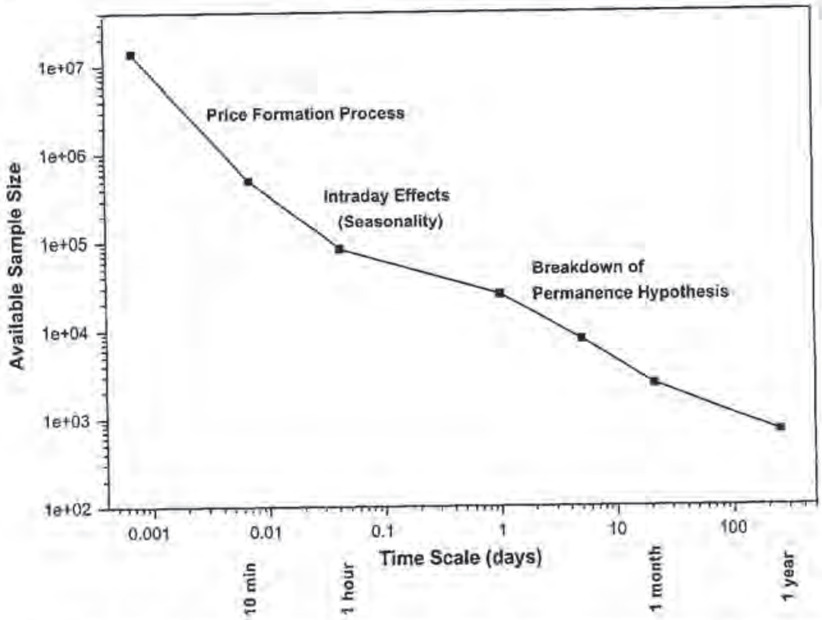


FIGURE 1.1 Available data samples with their typical sizes and frequency. The sample size and the frequency are plotted on a logarithmic scale. The first point corresponds to the O&A database, the last one to the 700 years of yearly data analyzed by Froot *et al.* (1995), the second to its left to the cotton price data of Mandelbrot (1963), and the daily data are computed from the sample used in Ding *et al.* (1993) to show long memory in the S&P 500. The text refers to the effects discovered and analyzed in the different segments of these samples.

double logarithmic scale makes the points lie almost on a straight line. The data sample with the lowest frequency is the one used by Froot *et al.* (1995) of 700 years of annual commodity price data from England and Holland. Beyond 700 years, one is unlikely to find reliable economic or financial data.[3] The data with the highest frequency is the Olsen & Associates (O&A) dataset of more than 14 years of high-frequency foreign exchange data. The tick-by-tick data are the highest frequency available. Between those two extremes, one finds the daily series of the Standard & Poors 500 from 1928 to 1991 used by Ding *et al.* (1993) or the monthly cotton prices used by Mandelbrot (1963) from 1880 to 1940. On this graph, we superimpose those effects that have been identified at these different time scales. One of the questions with data collected over very long periods is whether they really refer to the same phenomenon. Stock indices, for example, change their composition through time due to mergers or the demise of companies. When analyzing the price history of stock indices, the impact of these changes in

[3] Data can be found in natural sciences such as weather data up to a few hundred thousand years.

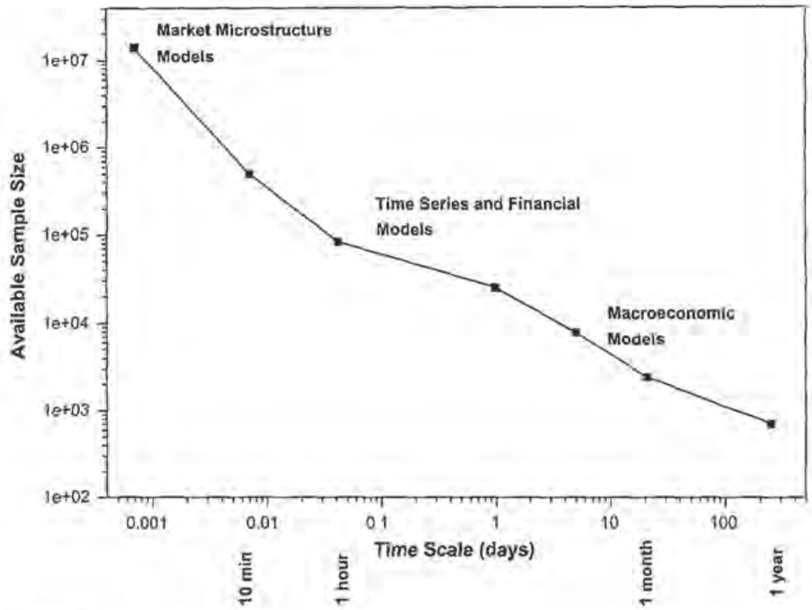


FIGURE 1.2 Available data samples with their typical sizes and frequency. The sample size and the frequency are plotted on a logarithmic scale. The text refers to the models developed and tested in the different segments of these samples.

composition is not obvious. We call this phenomenon the "breakdown of the permanence hypothesis." It is difficult to assess the quality of any inference as the underlying process is not stationary over decades or centuries. At the other end of the frequency spectrum (i.e. with high-frequency data), we are confronted with the details of the price generation process, where other effects, such as how the data are transmitted and recorded in the data-base (see Chapter 4) have an impact. With data at frequencies of the order of one hour, a new problem arises, due to the fact that the earth turns and the impact of time zones, where the seasonality of volatility becomes very important (as we shall see in Chapter 5) and overshadows all other effects.

Figure 1.2 relates the data to the models that are typically developed and tested with them. The high-frequency data have opened great possibilities to test market microstructure models, while traditionally low-frequency data are used for testing macroeconomic models. In between lies the whole area of financial and time series modeling, which is typically studied with daily or monthly data as, for instance, option pricing or GARCH models. It is clear from this figure that we have a continuum of both samples and models. The antagonism that is sometimes encountered between time series and market microstructure approaches should slowly vanish with more and more studies combining both with high-frequency data. Yet the challenge is still open to build models that are simple to implement and describe to a reasonable degree the empirical behavior of the data at all time scales.

1.4 NEW LEVELS OF SIGNIFICANCE

High-frequency data means a very large amount of data. The number of observations in one single day of a liquid market is equivalent to the number of daily data within 30 years. Statistically, the higher the number of independently measured observations, the higher is the degrees of freedom, which implies more precise estimators. The large amount of data allows us to distinguish between different models (model validation) with a higher statistical precision. New statistical methods become possible, for example, tail statistics to examine the probability of extreme events. Almost by definition, extreme events are rare and doing statistics on such extreme events is a challenge. With high-frequency data one can have samples with as many as 400,000 independent observations[4] to study the 0.25% percentile and still have 1,000 observations with which to work. We shall see how important this is when we present the estimation of tail indices for return distributions. Similarly, when different models have to be ranked, the availability of a few hundred thousand observations allows us to find beyond a doubt which model provides the best description of the data-generating process (Müller *et al.*, 1997a).

Figure 1.3 demonstrates the importance of high-frequency data in model selection and inference within the context of Value-at-Risk (VaR) calculations. We report three different calculations all of which use the J. P. Morgan (1996) volatility model, which is in fact a 1-day volatility forecast as further discussed in Section 9.2. The three calculations differ in terms of the sampling and the data frequency. The Japanese volatility calculations are based on prices observed *daily* at 7 a.m. GMT, which corresponds to the afternoon Japanese time. The U.K. volatility calculations are based on prices measured *daily* at 5 p.m. GMT, which is the afternoon in the U. K. The high-frequency volatility calculations are based on the high-frequency tick-by-tick data recorded continuously on a 24-hour cycle. The top panel in Figure 1.3 reports the annualized volatility calculations and the bottom panel shows the underlying prices for January and February 1999. The top panel demonstrates that volatility can be extremely different depending on the time of the day at which it is measured with daily data. If observations are picked randomly once a day, the underlying volatility can be as small as 15% or as large as 22% for a given day and for the same currency. In mid-January 1999, the U.S. Dollar - Japanese Yen (USD-JPY) investors in the U.K. are assumed to be facing the risk of losing 56,676,400 USD in a portfolio of a hundred million USD with a 1% probability. In Japan, this risk would be reduced to 38,643,000 USD for the same day and for the same currency, a difference of approximately 18,000,000 USD between the two geographical locations! The utilization of high frequency leads to more robust

[4] This approximately corresponds to 10 years of returns measured over 10 minutes.

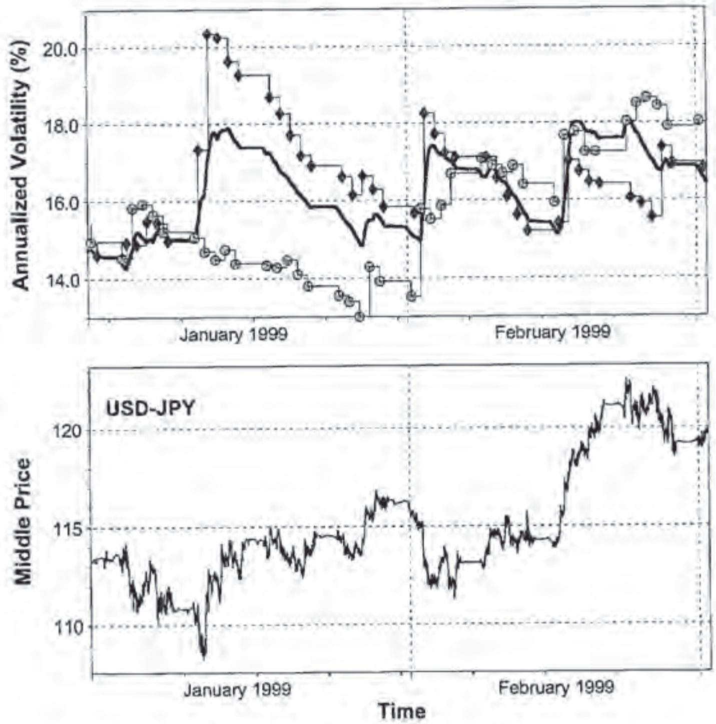


FIGURE 1.3 Top panel: Annualized USD-JPY volatility computed with daily prices observed at 7 a.m. GMT (afternoon Japan, circles), 5 p.m. GMT (afternoon U.K., diamonds) and with high-frequency data (solid line). The data period is from January 1999 to February 1999. Bottom panel: The USD-JPY high-frequency price series from January 1999 to February 1999.

annualized volatility estimations by minimizing the influence of the random noise in the market.

Another aspect of this is the choice of model. With few data, one tends to favor the simpler models because they contain few parameters and because tests like the likelihood ratio test would strongly penalize the increase of parameters. Of course, simplicity is a desirable feature of theoretical models, but one should not seek simplicity at the cost of missing important features of the data-generating process. Sometimes, it is useful to explore more complicated (nonlinear) models, which may contain more parameters. This increasing complexity is strongly penalized when explored with low-frequency data because of the loss of degrees of freedom. In the case of high-frequency data, however, the penalty is relatively small because the abundance of the independently measured observations approximates an asymptotic environment.

Researchers who want to use many observations with low-frequency data are using, for instance, daily observations of the Dow Jones Industrials from January 1897 like Ding *et al.* (1993) or LeBaron (1999a). In such a case, one is entitled to ask if the authors are actually analyzing the same market over the years. The huge technological changes that we experienced during this century have certainly affected the New York Stock Exchange and one is never sure, how this and any reconfiguration of the index has affected the results. To the contrary, high-frequency studies can be done for limited sampling periods with reasonably large samples. The market properties within such periods are nearly unchanged. The results are less affected by structural breaks or shifts in the overall economy than low-frequency studies with samples of many years. This is clearly an advantage when determining microstructure effects but also when examining the stability of some properties over time.

1.5 INTERRELATING DIFFERENT TIME SCALES

High-frequency data open the way for studying financial markets at very different time scales, from minutes to years. This represents an aggregation factor of four to five orders of magnitude.[5] Some empirical properties are similar at different scales, leading to fractal behaviors. Stylized facts observed for daily or weekly data gain additional weight when also observed with high significance for intraday data. An example of this is the long memory effect in 20-minute absolute returns studied by Dacorogna *et al.* (1993). At the time, similar hyperbolic decay of the autocorrelation function was observed on daily returns in Ding *et al.* (1993). It is very difficult to distinguish rigorously in the data between long memory effects and regime shifts. Many mathematicians are working precisely on this problem such as Mansfield *et al.* (1999) and Mikosch and Starica (1999). Yet the fact that hyperbolic decay is empirically found at time scales that differ by two orders of magnitude in aggregation is definitely a sign that the process must include some long range dependence or that there are regime shifts at *all time scales*, which is equivalent.

Scaling properties and scaling laws have been new objects of study since the early work of Mandelbrot (1963) on cotton prices. In 1990, the research group of O&A published empirical studies of scaling properties extending from a few minutes to a few years (Müller *et al.*, 1990). These properties have shown remarkable stability over time (Guillaume *et al.*, 1997) and were found in other financial instruments like interest rates (Piccinato *et al.*, 1997). Mantegna and Stanley (1995) also found scaling behavior in the stock indices examined at high frequency. In a set of recent papers, Mandelbrot *et al.* (1997), Fisher *et al.* (1997) and Calvet *et al.* (1997) have derived a multifractal model based on the empirical scaling laws of different moments of the return distributions. Works on the scaling law of return

[5] By order of magnitude we mean the number of times the time horizon must be multiplied by 10 to achieve the lower frequency. For instance, a weekly frequency is aggregated three orders of magnitude from 10 minutes data (one week is 1008 times 10 minutes).

volatility have been flourishing in the past few years often coming from physicists who started venturing in the field of finance calling themselves "econophysicists." It is a sign that the field is moving toward a better understanding of aggregation properties. Unfortunately, the mathematical theory behind these empirical studies is not yet completely mature and there is still controversy regarding the significance of the scaling properties (LeBaron, 1999a; Bouchaud *et al.*, 2000). Thanks to high-frequency data, this kind of debate can now take place. The challenge is to develop models that simultaneously characterize the short-term *and* the long-term behaviors of a time series.

2

MARKETS AND DATA

2.1 GENERAL REMARKS ON MARKETS AND DATA TYPES

In the discussion of markets and data, we take the point of view of researchers studying high-frequency data rather than the view of traders or other practitioners. Instead of giving complete descriptions of markets (which are changing over time), we focus on those main markets that have produced consistent time series data over many years.

High-frequency data are direct information from markets. One logical unit of information is called a *tick*. This term originated from the language of practitioners and originally meant a number on a ticker tape, in a time before computers became an omnipresent tool. The term "tick" is more neutral and general than the particular terms "price," "interest rate," "quote," and so on. Whatever the quoted quantity, there is always a date and a time attached to every tick, a "time stamp." The sequence of time stamps is usually irregularly spaced. A large part of Chapter 3 deals with the consequences of this fact.

The quoted quantities are often prices, but other information such as transaction volume is also available from some markets. Detailed information on participants (e.g., the counterparty of transactions) is, however, rare because market participants often prefer anonymity.

Some markets are centralized in the form of exchanges or bourses. Other markets are decentralized interbank (over-the-counter) markets, where individual participants directly transact with no intermediary. Data from over-the-counter (OTC) markets are collected and provided in real time by data providers such as Reuters, Bloomberg, or Bridge. Data from centralized markets are available from the same sources and sometimes directly from the exchanges. The recent shift from floor trading to electronic trading helped to make this data more reliable and more easily available. Some data are released to a general audience only after a time delay when its direct value for traders has diminished. Exchanges such as London International Financial Futures Exchange (LIFFE) and the New York Stock Exchange (NYSE) also sell archived historical data, such as data from the TAQ database of the NYSE. There are also vendors of historical data such as O&A specializing in high-frequency data initially collected in real time from different sources.

The foreign exchange (FX) market has the highest market volume of all financial markets. A large part of this volume is traded over-the-counter between banks, but there is also electronic trading through centralized systems. The coexistence of interbank and centralized trading is found also in other markets. This implies that volume figures—if available—often refer to a market segment rather than all transactions of the whole market. Aside from FX, we discuss interest rate, bond, equity, and commodity markets; the latter two are scarcely treated in this book. All of these assets are directly traded in *spot* markets (see Section 2.1.1) and indirectly in *derivative markets*: *futures* and *option* markets as discussed in Sections 2.1.2 and 2.1.3.

In this book, there is no attempt to list all the changing types and trading mechanisms of markets and all the varying formats and availability conditions of data from different data suppliers.[1] Instead, we report stylized properties of time series data for markets with sampling periods of several years.

2.1.1 Spot Markets

Spot markets are direct markets for primary assets, such as foreign exchange or equity. The assets are traded immediately at the time of the transaction.[2] Spot trading is the most original form of trading, but it has some disadvantages. The timing is not flexible, traders have to deal with the physical delivery of the traded assets (such as commodities) and the interest rate spot market is affected by the counterparty default risk. For these reasons, derivative markets have become more important than spot markets in some cases. The FX market is a major example of a market where spot trading is still strong.

[1] A review is provided by Gwilym and Sutcliffe (1999).

[2] Transactions are actually booked at the *value date*, which is usually two days after the day of the spot transaction or, if that is a holiday, the first business day afterward. This fact hardly has an influence on prices, it just affects the timing of bookkeeping. Therefore, value dates can be ignored in most studies.

Some important spot markets are over-the-counter markets between individual institutions (banks), whereas many derivatives are traded at exchanges. However, this is not a general rule.

2.1.2 Futures Markets

In some cases such as most interest rates and commodities, futures markets have a higher liquidity and volume than the underlying spot markets and produce better high-frequency data. The following description of futures markets is quite general, and special features of particular futures markets are discussed in other sections.

Futures contracts are derivatives of an underlying asset, which can be defined as an agreement between two parties to buy or sell an asset at a certain time in the future for a certain price, Hull (1993). At this expiry time, the underlying asset has to be delivered according to settlement rules, after which the contract no longer exists. The expiry dates are regularly scheduled, often in a quarterly sequence. The contract with the nearest expiry is called the first position, the following contract the second position, and so on.

Most futures are traded at exchanges. Trading is typically geographically localized. There is no 24-hour trading, there are rigidly defined opening hours, although the trend is to effectively lengthen the active hours (e.g. with after-hours sessions). Given that futures contracts are exchange traded and each transaction is recorded centrally, futures markets offer a high price transparency. The historical data always include tick-by-tick transaction prices and, depending on the data source, bid and ask quotes and sometimes information on volumes and the flow of orders from the clients of the exchange.

The structure of many futures markets has changed due to the rapid growth of market volumes, some mergers of exchanges, and the shift from floor trading (open outcry) to electronic trading. For some researchers, this shift has been an object of study in itself.

All clients buying or selling futures contracts have to put some money in a collateral account. This account covers the counterparty default risk of the exchange. If the futures prices move to the disfavor of a client, the amount of money on the collateral account may no longer cover the risk, and the exchange will tell the client to increase it through a "call for margin." If the client fails to do this, the futures contract is terminated at its current market value. The flow of money to the collateral account (which earns some interest in its own right) makes the exact bookkeeping of returns (and risks) rather complicated, but this can be ignored in most studies.

The time series of prices coming from a single futures contract is not sufficiently long for certain statistical studies. Moreover, the behavior of a contract changes when approaching expiry and its price volatility systematically grows or shrinks according to the nature of the underlying asset. For these reasons, it is sometimes necessary to construct long samples joining several contracts together. Different empirical prescriptions are used by analysts and traders to join price histories of several futures contracts with successive expiries. Such prescriptions are typically based on rollover schemes – that is, they attempt to replicate the behavior of a trader holding a contract and switching ("rolling over") to the next contract before the expiry of the current contract. Section 2.5.2 gives such an example.

The opening hours of futures markets are sometimes modified, as those of other centralized markets. Long samples may extend over periods with different fixed opening hours. This fact leads to some difficulties in intraday studies, especially those related to daytime. Researchers should be aware of this and know the history of opening hours.

2.1.3 Option Markets

Option prices are very volatile and depend on parameters such as the strike price, the base spot price, and the expiry date. There are many types of options. The options markets are often too volatile for studying consistent time series over long samples and are not the subject of this book—except for the *implied volatility* aspect.

Unlike option prices, implied volatility figures are slowly changing over time. They are computed from option prices through the formulas introduced by Black and Scholes (1973) and some refined methods introduced later. The implied volatility figures provided by data vendors usually refer to at-the-money options (where the strike price is not far from the base spot price). For some markets, these data are available in high frequency with several quotes per day. Implied volatility can be interpreted as the market's forecast of the volatility of the underlying asset for the time period until expiry. Therefore, time series of implied volatility are interesting especially in comparison to historical or realized volatility computed from time series of underlying assets.

2.2 FOREIGN EXCHANGE MARKETS

The foreign exchange (FX) market is the largest financial market. Already in April 1992, its "traditional" part (FX spot and FX forward market, excluding the newer derivatives) had a daily net-net[3] turnover of 832 billion U.S. Dollar (USD) (Bank for International Settlements, 1993) which was more than the total non-gold reserves (USD 555.6 billion) of all industrial countries in 1992 (International Monetary Fund, 1993). Since that time, the FX net-net turnover had grown to USD 1190 billion in April 1995 and to USD 1500 billion in April 1998 (Bank for International Settlements, 1999).

The FX spot market produces high-frequency data that played and still play a central role in high-frequency finance. Unlike other data, these data are available over long sampling periods in high frequency, 24 hours per working day. The market is highly liquid and symmetric as both exchanged assets are currencies.

[3] This figure is adjusted for both local and cross-border double-counting.

Due to these favorable characteristics, new facts have often been found in FX spot data, and FX studies have served as a role model for the investigation of other high-frequency data with less favorable properties.

Since the beginning of the 1990s, academic researchers have been gaining new insights into the behavior of the FX markets through analyzing intraday data. Daily data, which were much used in the 1980s represent only a small subset of the information available at intraday frequencies, as they are only the average of a few intraday prices quoted by some large banks at a particular daytime. The number of data points available for intraday is larger by a factor of 1000.

On the basis of this information set, there is a rapidly growing body of literature in the study of the intraday FX markets, which opens new directions for understanding of financial markets and widening of concepts such as risk management or market efficiency. The analysis of intraday data also leads to insights into the market microstructure where it is possible to study the behavior of intraday traders, whose operations account for more than 90% of the FX market volume.

The stylized facts found for intraday FX rates shed some new light on different modeling approaches to the FX market.[4] Research studies have shown that known and well-accepted empirical regularities of daily or weekly data do not always hold up in intraday analysis. Looking at intraday data, the homogeneity of market agents (which is a working hypothesis for studying daily, weekly, or low-frequency data) disappears. A new wealth of structure is uncovered that demonstrates the complexity of the FX market at the intraday frequency. This complexity can be explained by the interaction of market agents with heterogeneous objectives resulting from different geographical locations, the various forms of institutional constraints, and risk profiles. This evidence will be presented in several chapters of this book. Indeed, the heterogeneous structure of intraday data may explain the fact that practitioners have effectively used methods of "technical analysis" over many years now. These intuitively designed methods try to take advantage of the interaction of different components of the markets, see Dunis and Feeny (1989); Neftci (1991); Surajaras and Sweeney (1992); Taylor and Allen (1992); Pictet *et al.* (1992); Levich and Thomas (1993b); Brock *et al.* (1992); Gençay and Stengos (1998) and Gençay (1998a,b, 1999); Gençay *et al.* (2001c, 2002).

The FX spot market is presented in Section 2.2.1. Aside from the spot market, there is also the over-the-counter FX forward market treated in Section 2.3.2 and the markets for FX futures and FX options. The contracts of these markets refer to a time period in the future, therefore they are affected by interest rate levels.

Exchange-traded FX futures follow the description of Section 2.1.2 and are not discussed here; their market volume is much lower than that of the FX spot market and the over-the-counter derivative markets, particularly the FX forward market.

As mentioned in Section 2.1.3, time series of implied volatility are available from the FX option markets. These are interesting objects of study, together with realized or historical volatility computed from FX spot rates.

[4] For surveys on the FX market at the daily or weekly frequencies, see, for example, the surveys of Mussa (1979); Hsieh (1988); Baillie and McMahon (1989), and de Vries (1992).

2.2.1 Structure of the Foreign Exchange Spot Market

The usual description of the FX markets made by international organizations such as the Bank for International Settlements (1999) or the International Monetary Fund (1993) emphasizes the presence of different geographical markets and different types of agents. However, these structural characteristics are not apparent from the inspection of daily or weekly data and their implications were seldom considered in theoretical modeling.

The FX market consists of two dominating parts, the FX spot and the FX forward market, and a smaller but growing market of FX derivatives. In 1992, both the FX spot and forward markets had a market share of roughly 50% each. In 1998, the FX forward market was moderately larger than the FX spot market, by a ratio 60:40% (Bank for International Settlements, 1999). However, the FX spot rates are used not only in the FX spot market but also for outright FX forward transactions, as explained in Section 2.3.2.

Nowadays, a growing segment of the FX spot transactions goes through automated, electronic order-matching systems,[5] such as the Electronic Broking Services (EBS) and Reuters Dealing 2000. These markets deliver good high-frequency data with transaction prices and volumes. These transaction data have become available to researchers to a limited amount.[6] Results based on these data are discussed in a few places in this book.

Aside from this electronic trading, the over-the-counter FX spot market is a direct market between banks (and brokers). The bid and ask offers of major financial institutions are conveyed to customers' screens by large data suppliers such as Reuters, Bloomberg, and Bridge (formerly Telerate and Knight Ridder) with as little time delay as possible and the deals are negotiated over the telephone. Many researchers have investigated data in various forms from this interbank market; there is no alternative way to obtain large FX samples, especially historical samples extending to the 1980s and early 1990s.

The bid-ask prices from the over-the-counter FX spot market are called *quoted* prices or quotes as opposed to transaction prices. One full tick contains the time stamp, a bid and an ask price and often some information on the origin of the tick (bank code, city, etc.). Transaction prices or volumes are not available, but some additional information such as financial news in text form is available from other pages of the data vendors.

The data suppliers offer a rather easy access to market data nowadays. The real-time data for a financial instrument can be obtained from Reuters, for instance, through the Reuters Instrument Code (RIC). The FX rate EUR-USD, for instance,[7] has the RIC "EUR=," the FX rate USD-JPY has the RIC "JPY=." Rates

[5] In April 1998, the share of deals going through such systems was "almost one quarter" in the United Kingdom, "almost one third" in the United States, and 36% in Japan, (Bank for International Settlements, 1999).

[6] Lyons (1995, 1996a,b); Goodhart *et al.* (1995), and Goodhart and Payne (1996) could obtain a few days of such data.

[7] For the currencies, we use the standard abbreviations of the International Organization for Standardization (ISO, code 4217).

TABLE 2.1 The traditional FXFX page of Reuters.

On this traditional page, the first column gives the time in GMT (for example, for the first line, "07:27"), the second column gives the name of a traded currency ("DEM" for USD-DEM), the third column the name of the bank subsidiary that publishes the quote as a mnemonic ("RABO"), the fourth column the name of the bank ("Rabobank"), the fifth column the location of the bank as a mnemonic ("UTR" for Utrecht), the sixth column gives the bid price with five digits ("1.6290") and the two last digits of the ask price ("00" which means 1.6300), the seventh column repeats the currency ("DEM"), and the last two columns give the highest ("1.6365") and the lowest ("1.6270") quoted prices of the day.

```
0727 CCY PAGE NAME   * REUTER SPOT RATES * CCY    HI*EURO*LO FXFX
0727 DEM RABO RABOBANK UTR    1.6290/00  * DEM    1.6365    1.6270
0727 GBP MNBX MOSCOW   LDN    1.5237/42  * GBP    1.5245    1.5207
0727 CHF UBZA U B S    ZUR    1.3655/65  * CHF    1.3730    1.3630
0727 JPY IBJX I.B.J    LDN    102.78/83  * JPY    103.02    102.70
0727 FRF BUEX UE CIC   PAR    5.5620/30  * FRF    5.5835    5.5582
0726 NLG RABO RABOBANK UTR    1.8233/38  * NLG    1.8309    1.8220
0727 ITL BCIX B.C.I.   MIL 1592.00/3.00 * ITL   1596.00   1591.25
0727 ECU NWNT NATWEST  LDN    1.1807/12  * ECU    1.1820    1.1774

XAU SBZG 387.10/387.60 * ED3  4.43/ 4.56 * FED    PREB * GOVA 30Y
XAG SBCM    5.52/ 5.53 * US30Y YTM  7.39 * 4.31- 4.31 * 86.14-15
```

between two currencies other than the U.S. Dollar (USD) are called cross rates (see Section 2.2.2) and have both currencies in the RIC (e.g. "EURJPY=" for the rate EUR-JPY). There are also RICs serving as "tables of contents," called tiles. These contain information on other RICS. Once a real-time feed for a financial instrument is established, the data are transmitted in data records, each representing a tick. A data record contains several variables such as bid and ask prices and some side information. This data organization is used for all financial data, not just for FX.

Once the instrument codes and the variables of the data records are known, data collection is a straightforward operation. However, many results of this book are based on historical data collected at a time when data extraction was technically difficult. The traditional data feeds of the early 1990s and before were oriented to human users looking at full pages of information, such as the FXFX page of Reuters. Table 2.1 shows a snapshot of this traditional FXFX page. A quoted price of 1.6290/00 for the USD-DEM rate expresses the willingness of the market maker to buy USD at 1.6290 DEM and sell USD at 1.6300 DEM. Telerate also had a page-based feed at that time. Rather than using today's instrument codes, the data collector had to extract the desired information from full pages and from the many real-time updating messages associated to these pages. This implied a delicate text parsing task and also controlling for unexpected changes in the page layout. A large part of the data used in the studies of this book has been extracted and collected in this tedious way.

The FX market has no business-hour limitations. Any market maker can submit new bid-ask prices; many larger institutions have branches worldwide so that trading is continuous. Nevertheless, the bid-ask prices do emanate from particular banks in particular locations and the deals are entered into dealers' books in particular institutions. Although the FX market is virtually global through its electronic linkages, its activity pattern can be divided into three continental components, each with its typical group of time zones: East Asia, Europe, and America with Tokyo, London and New York as major trading centers[8] (Goodhart and Demos, 1990).

Except for some major currencies against the USD, currencies tend to be traded more specifically in their own geographical markets. Major currencies are USD, EUR (until 1998: DEM), JPY, GBP, and CHF. Both the global and local characteristics of the FX markets are reflected by the statistical properties of the data.

Actual trading prices and volumes are not known from the over-the-counter spot market. However, reputation considerations prevent market makers from quoting prices at which they would actually not be willing to trade. Therefore, real transaction prices tend to be contained within the quoted bid-ask spread (Petersen and Fialkowski, 1994). This is also shown by a comparison to simultaneous transaction prices of electronic dealing systems (where the bid-ask spread is narrower).

The growing volume of FX transactions has been increasingly made up of short-term, intraday transactions and results from the interaction of traders with different time-horizons, risk-profiles, or regulatory constraints. Nonfinancial corporations, institutional investors (mutual funds, pension funds, insurance companies), and hedge funds[9] have shifted their FX activities from long-term (buy and hold) investment to short-term (profit-making) transactions. This movement is both enabled and enhanced by the development of real-time information systems and the decrease of transaction costs following the liberalization of cross-border financial flows. This flow of short and long-term transactions initiated by nonfinancial institutions on the retail market is the origin of an even larger—by a factor of four to five times—flow of intradaily transactions between the dealers (the 50 largest banks and a few securities houses) on the wholesale market. These dealers, who are usually not allowed to take overnight positions, transact with each other to reduce the risk arising from their accumulated currency positions (Lyons, 1996a).

[8] In typical historical samples, the data contributors of "East Asia" are located in Australia, Hong Kong, India, Indonesia, Japan, South Korea, Malaysia, New Zealand, and Singapore. "Europe" covers Austria, Bahrain, Belgium, Germany, Denmark, Finland, France, Great Britain, Greece, Ireland, Italy, Israel, Jordan, Kuwait, Luxembourg, the Netherlands, Norway, Saudi Arabia, South Africa, Spain, Sweden, Switzerland, Turkey, and United Arab Emirates. "America" comprises Argentina, Canada, Mexico, and the United States.

[9] The high leverage and unregulated aspects of hedge funds distinguish their investors from other institutional investors.

TABLE 2.2 Numbers of archived ticks of main FX rates.

Tick frequencies of main FX rates: (1) main rates against the USD, (2) main cross rates, and (3) main rates against historical currencies now replaced by the Euro (EUR).

FX rate	Period	Number of ticks	Frequency per business day
EUR-USD	Jan 1999 – May 2000	4,794,958	13,300
USD-JPY	Jan 1987 – May 2000	9,585,136	2,800
GBP-USD	Jan 1987 – May 2000	7,892,919	2,310
USD-CHF	Jan 1987 – May 2000	8,310,226	2,430
EUR-JPY	Jan 1999 – May 2000	1,897,007	5,250
EUR-GBP	Jan 1999 – May 2000	1,740,209	4,820
USD-DEM	Jan 1987 – Dec 1998	18,416,814	6,020
USD-FRF	Jan 1987 – Dec 1998	3,655,638	1,190
DEM-JPY	Oct 1992 – Dec 1998	1,316,933	712

Still on the wholesale market, but in contrast to other players, central banks can afford relatively large open positions and can thereby have a significant impact on the market in the long run. The different types of traders can of course be found within similar types of institutions.[10]

To illustrate the enormous amounts of available FX spot ticks, Table 2.2 displays the size and frequencies of ticks in the Olsen & Associates (O&A) database. The FX rates of this table are between major currencies, the first one of a currency pair being the "exchanged" currency whose value is expressed in the second currency (which can be called the numeraire currency). The analyzed periods have been chosen with respect to the transition of some European currencies such as DEM and FRF to the Euro (EUR) at the beginning of 1999. On the largest market, EUR-USD, more than 10,000 ticks per business day are available; that is an average of almost 10 ticks per minute which can rise to 30 or more ticks per minute during the busiest periods. The daily tick frequencies of Table 2.2 are averages. The values of the 1980s and the early 1990s were distinctly smaller than the values nowadays. In today's data feeds, some of the ticks contain little information if they are copies of ticks from other contributors (as explained in Section 2.2.3) or repeatedly posted ticks (see Section 4.2.2). Minor FX rates have fewer ticks than the rates in Table 2.2, very few ticks if the liquidity is low.

In contrast to daily or weekly data, collecting tick-by-tick quotes presents a number of practical problems such as transmission delays and breakdowns or

[10] For example, Bank Negara of Malaysia was one of the most aggressive (short-term) speculators in the FX market for several years.

aberrant quotes due to human and technical errors. Therefore, it is important to implement a data cleaning filter to eliminate outliers. An extensive discussion of data cleaning is presented in Chapter 4.

2.2.2 Synthetic Cross Rates

FX rates between two currencies other than the U.S. Dollar (USD) are called *cross rates*. There are quotes for some important cross rates such as those in Table 2.2. For many other cross rates, there is little data or no data at all, either because the market for that cross rate is neglected by the data suppliers or because there is no direct market at all. In the second case, traders would go through a *vehicle currency* such as USD or EUR instead of making a direct transaction. A Canadian trader, for example, would obtain Japanese Yen (JPY) by buying USD from the USD-CAD market and selling USD on the USD-JPY market. The actual exchange rate in this case is

$$p_{\mathrm{JPY/CAD,bid}} = \frac{p_{\mathrm{USD/CAD,bid}}}{p_{\mathrm{USD/JPY,ask}}}, \quad p_{\mathrm{JPY/CAD,ask}} = \frac{p_{\mathrm{USD/CAD,ask}}}{p_{\mathrm{USD/JPY,bid}}} \qquad (2.1)$$

These formulas reflect the triangular relation between the three currencies USD, JPY, and CAD. If a direct market for JPY-CAD exists and its prices strongly deviate from this relation, the deviation can be profitably exploited through a set of riskless transactions. Such a strategy is called triangular arbitrage and leads to market adjustments that bring the prices back toward the relation of Equation 2.1. Traders are usally quick enough to make such arbitrage transactions before the prices strongly deviate from Equation 2.1. A trader calling market makers to execute an indirect transaction has to pay the bid-ask spreads of two markets, both USD-CAD and USD-JPY in our example. Both bid-ask spreads together may be higher than the bid-ask spread of a direct transaction.

In the case of cross rates with a direct market but bad data coverage, we are forced to compute synthetic cross rates through formulas such as Equation 2.1 which serve as proxies for the unknown direct rates. The two ticks used for the cross rate computation (a USD-CAD tick and a USD-JPY tick in the example of Equation 2.1) should be synchronous such that their time stamps deviate by no more than a few seconds or perhaps a minute. Otherwise, the synthetic cross rate is distorted by price moves in the time interval between the two ticks. The bid-ask spreads of the missing direct quotes can be expected to be lower than the synthetic bid-ask spreads. The data coverage for cross rates may only be bad during the active market hours of some time zones (e.g., due to the bad coverage of Asian markets by some data suppliers). In this case, we may have to mix quoted cross rate data with synthetic data (at certain daytimes).

2.2.3 Multiple Contributor Effects

The transactions of over-the-counter markets are between many individual institutions (banks and brokers). The market makers among these institutions publish their own price quotes. Data suppliers mix these data with the quotes of other contributors, thus creating a multicontributor data feed. Individual quotes are affected by the positions, views, and trading strategies of individual contributors, rather than behaving uniformly. Researchers using these data should be aware of this fact.

The FX spot market is the key example of a multicontributor market. The following multicontributor effects have been found in FX spot data:

- Depending on their inventory position, market makers have preferences for either selling or buying. They publish new quotes to attract traders to make a deal in the desired direction so that either the bid or the ask quote is competitive. The other price of the bid-ask pair is pushed away to a less attractive region by adding or subtracting a rather large, nominal spread. This leads to high unrealistic quoted bid-ask spreads and a negative autocorrelation of returns at lags of around one minute as discussed in Section 5.2.1.

- There are contributors of low reputation that abuse some quotes in attempts to manipulate the market into a desired direction.

- FX quotes lag behind the real market prices. This is confirmed by FX traders we have interviewed and from comparisons to transaction data from electronic trading systems. A closer look shows that some leading contributors do not have a considerable delay, whereas many other contributors lag behind by more than a minute. This can be shown through a *lead-lag* correlation analysis of returns of contributor-specific time series.

- Some contributors are laggards because they publish prices copied from the quotes of other contributors (e.g., moving averages of recent quotes with a tiny random modification). The motivation is to advertise the market presence of the contributor in the data feed (whereas true prices are negotiated over the telephone). These contributors often employ computers to publish fake quotes at high frequency. The described data-copying methods lead to lower data quality in general and lead to data-cleaning problems as discussed in Chapter 4.

Similar multicontributor effects are also found and expected in markets other than the FX spot market.

2.3 OVER-THE-COUNTER INTEREST RATE MARKETS

Two financial markets related to interest rates are over-the-counter (OTC) markets between individual banks.[11] These are the spot interest rate (IR) market and the FX forward market.

[11] This is similar to the interbank FX market.

2.3.1 Spot Interest Rates

Interest rate quotes have been directly available from the over-the-counter market for many years, for example, through the multicontributor "deposit" pages of Reuters. These interest rates are offered by banks to other banks who want to make either a deposit (at the bid interest rate) or take a credit (at the ask interest rate). The quotes come in bid-ask pairs and are called spot interest rates, cash interest rates, or interbank interest rates. Another traditional name, "Eurodeposits," suffers from possible confusion with the new currency named Euro. For quite some time, the spot interest rate (IR) market is no longer the most liquid IR market. This role is now taken by the IR futures market, see Section 2.4. The low liquidity of the spot IR market is reflected by the rather large spread between bid and ask quotes.

The actual rate at which a bank is ready to lend money to another one also depends on the credit rating of that bank. A bank with a low credit rating has to be ready to pay a higher IR in order to attract a lender; the IR level is increased by the *credit spread*. This fact makes the IR quotes less universally applicable than the FX quotes.

The credit spread can lead to serious data problems, which can lead to spurious statistical results. The story of the Japanese interest rates in the second half of the 1990s is the best illustration. There was a banking crisis in Japan that lowered the credit ratings of Japanese banks. In some data sources such as the Reuters deposit pages, these banks dominated some daytimes corresponding to the working hours of East Asian time zones. All of the IR quotes during these daytimes were systematically higher than the quotes at other daytimes. The market was split between low-rating banks and high-rating banks, causing two spurious statistical effects: (1) too high absolute values of returns (level changes) over time intervals of around 12 hr and (2) strong negative autocorrelation of returns at lags of around 12 hr. These spurious effects are solely due to periodically shifting credit ratings, which can be avoided by eliminating all the low-rating (or all high-rating) quotes from the sample. For a long time, Japanese Yen (JPY) interest rates were very low, below 1% for Japanese banks. The credit spread led to a strange effect around 1998 by pushing the IR levels for non-Japanese banks (dominating the European and American time zones) slightly *below zero*. Since then, this has been a classical example rejecting zero as an absolute minimum of IRs. Slightly negative interest rates can be possible and valid under special circumstances.

Spot interest rates always refer to a deposit of fixed duration, the maturity period. The following maturity periods are quoted in the market: overnight (O/N), "tomorrow next" (T/N, the next business day after tomorrow), 1 week (S/W), 1 month (1M), 2 months (2M), 3 months (3M), 6 months (6M), 9 months (9M), and 1 year (1Y). Among these maturity periods, the 3-month maturity often has the largest market and the best data followed by the 1-month, 2-month, 6-month, and 1-year maturities. Spot interest rates are the only IR instruments that always inform us on interest rate levels for time intervals starting now and extending over less than 3 months. Therefore, their use is inevitable when constructing yield curves; see Section 2.4.2.

Spot IRs are quoted in annualized form and in percentage terms. A 3-month IR of 6%, for example, means that the invested capital is multiplied by 1.015 after 3 months, because 3 months = 0.25 years and $0.25 \cdot 6\% = 1.5\% = 0.015$.

2.3.2 Foreign Exchange Forward Rates

Foreign exchange (FX) forward rates share some characteristics with the spot IRs. They also refer to the interbank market and are quoted for the same maturities as the spot IRs (see Section 2.3.1). They are quoted on the same traditional Reuters deposit pages.

FX forward transactions are similar to FX spot transactions, except that the actual transaction takes effect in the future, at maturity.[12] The timing of FX forward transactions leads to a difference in interest payments as compared to FX transactions. Due to the delayed transaction, the buyer of an FX forward contract earns some interest on the base currency of the FX rate (the currency in which the FX rate is expressed) instead of the exchanged currency. If the interest rates of the two currencies deviate, there is a net interest payment flow from or to the buyer during the maturity period. This fact determines the price of the FX forward contract, called the *outright* forward rate. In order to avoid riskfree arbitrage, the outright forward rate deviates from the simultaneously quoted FX spot price by an amount to offset the deviations in interest payments. The price deviation between FX spot prices and outright forward rates thus reflects the interest rate *differential* between the two exchanged currencies rather than the absolute level of those IRs.

The arbitrage relation between outright FX forward rates f, spot interest rates i and FX spot rates p can be formulated as follows:

$$\begin{aligned} f_{\text{bid}} &= p_{\text{bid}} \frac{1 + i_{\text{expr,bid}} \frac{m}{1\,\text{year}}}{1 + i_{\text{exch,ask}} \frac{m}{1\,\text{year}}} \\ f_{\text{ask}} &= p_{\text{ask}} \frac{1 + i_{\text{expr,ask}} \frac{m}{1\,\text{year}}}{1 + i_{\text{exch,bid}} \frac{m}{1\,\text{year}}} \end{aligned} \tag{2.2}$$

where m is the maturity period (e. g. 0.25 years for a 3-month period). This formula can be found in usual textbooks such as Walmsley (1992) but only for a middle price, not for bid and ask. It is valid for maturity periods up to one year. For longer periods, formulas based on compound interest are needed. The interest rates i should not be used in percentage (e.g., 0.05 should be used instead of 5%). The index "exch" denotes the exchanged currency of the FX rate; the index "expr" defines the (numeraire) currency in which one unit of the exchanged currency is expressed.

[12] The transaction is actually booked at the *value date*, which is usually two days after the maturity date or, if that is a holiday, the first business day afterward. This fact hardly has an influence on prices, it just affects the timing of bookkeeping. Therefore, value dates can be ignored in most studies.

Instead of outright FX forward prices f, the *difference* $f-p$ is usually quoted, which is the outright forward price minus the simultaneously valid FX spot price. This difference is less volatile than the outright forward price and can be positive or negative according to the sign of the interest rate differential. A positive difference is called *forward premium*, a negative difference is a *forward discount*. Both are also called "forward points." This formulation relates to the units of "basis points" in which they are usually quoted, which is the multiples of the last decimal digit of normal FX spot quotes. As an example, assume an FX spot rate of 1.5025/30 ($p_{\text{bid}} = 1.5025$, $p_{\text{ask}} = 1.5030$) and quoted forward points of -23/-20. The outright forward price is therefore $f_{\text{bid}} = 1.5002$ $(= 1.5025 - 0.0023)$ and $f_{\text{ask}} = 1.5010$ $(= 1.5030 - 0.0020)$.

FX forward premiums and discounts are also called FX swap rates, following another common view where an FX forward transaction is seen as a spot transaction plus a "swap" transaction (swapping two currencies during the maturity period). In fact, there is a large market for such FX swap transactions independent from spot or outright forward transactions.

Equation 2.2 can be used to compute synthetic forward rates from an FX spot rate and the spot IRs of both underlying currencies. Such synthetic FX forward rates are less reliable than direct quotes. They have a distinctly higher bid-ask spread. This implies that an FX forward transaction is more efficient than a substitute set of transactions, consisting of an FX spot transaction plus a deposit in one currency plus a loan in the other one. Direct forward quotes exist for most FX rates against the USD, but may not be available for some FX *cross* rates (see Section 2.2.2). For these currency pairs, synthetic forward rates are needed.

2.4 INTEREST RATE FUTURES

2.4.1 General Description of Interest Rate Futures

Short-term interest rate (IR) futures are the most liquid financial instrument for the interest rate markets. In particular, the IR futures markets with expiry periods of up to one year (or slightly longer) are more liquid than the over-the-counter spot IR market, as presented in Section 2.3.1. The transaction costs are lower; a typical bid-ask spread is about 10% (or less) of the quoted spread of cash interest rates.[13] The mechanism of price formation for futures is faster than for cash contracts (Fung and Leung, 1993; Garbade and Silber, 1985). As a consequence, IR futures markets yield high-quality intraday data.

IR futures markets are futures markets in the sense of Section 2.1.2. More specifically, an interest rate futures contract is a futures contract on an asset whose price is dependent solely on the level of interest rates, (Hull, 1993).

[13] The bid-ask spread on the Chicago Mercantile Exchange (CME) Eurodollar contract, for example, can be as small as half a basis point. A basis point corresponds to 1/100 of 1%, and its monetary value (in the case of three-month IR futures) is $25. The minimum price movement for the CME Eurodollar is half a basis point.

IR futures are traded at the Chicago Mercantile Exchange (CME),[14] the EUREX (formerly the DTB, MATIF, etc.), the London International Financial Futures Exchange (LIFFE), the Singapore International Monetary Exchange (SIMEX), and other exchanges. In the first three quarters of 1997, the CME Eurodollar time deposit had a mean daily volume of 461,098 contracts,[15] with each contract corresponding to a notional 1 million dollar 3-month deposit. IR futures are exchange-traded contracts and this entails several differences with respect to over-the-counter (OTC) instruments, as already explained in Section 2.1.2. IR futures are linked to a specific exchange, except when a fungibility agreement is in effect.[16] Trading is typically limited to opening hours. Most IR futures exchanges have replaced or are replacing floor trading (open outcry) with electronic trading. In studies of long data samples, we may be forced to use a first half of data originating from floor trading and a second half from electronic trading.

IR futures are traditionally known under the name *Eurofutures*; the contracts were called "Eurolira," "Euroyen," and similar after the underlying currency. In statistical studies based on historical samples, these names may still be used, but nowadays, they lead to confusion with the new currency named the Euro.

Information on IR futures is particularly valuable for financial institutions and above all for banks. A quick analysis of a typical balance sheet would often reveal a higher exposure to IR risk than, for example, to foreign exchange risk. IR futures can also be used as hedging instruments. From a practical point of view, there is widespread need for a better understanding of the empirical behavior of IR futures on an intraday basis; nevertheless, in the literature there is little material on intraday IR futures markets. The studies by Ballocchi *et al.* (1999a,b), and Ballocchi *et al.* (2001) offer deep insights. The impact of scheduled news releases has been investigated by Ederington and Lee (1993, 1995).

IR futures refer to an underlying deposit (usually a 3-month deposit). An IR future price f (bid, ask, or transaction price) is quoted as a number slightly below 100 according to the following formulas

$$f = 100\left(1 - \frac{r}{100\%}\right), \quad r = \left(1 - \frac{f}{100}\right) 100\% \qquad (2.3)$$

where r is the annualized forward interest rate[17] with a forward period of usually 3 months. There are four main settlement months in a year (March, June, September, and December), known as *quarterly expiries*. Serial expiry contracts (i.e., contracts expiring in months that do not correspond to the quarterly sequence) have been introduced more recently and typically have lower liquidity.

Unlike the spot IRs, the futures prices are not affected by the individual credit ratings of clients, because the collateral account required by the exchange already

[14] They are traded at the International Monetary Market (IMM) Division of the CME.

[15] All expiries combined, as reported in the January 1, 1998, issue of *Futures* magazine.

[16] One example of a fungibility agreement is the mutual offset system between CME and SIMEX, through which contracts opened in one exchange can be liquidated on the other one.

[17] For instance, a value $r = 3\%$ implies a futures price of 97.00.

covers the credit risk. The existence of the collateral account can usually be ignored in studies of IR futures data.

Unlike other futures markets contracts, IR futures contracts are settled in cash. The notional deliverable asset is a (3-month) deposit starting at expiry, but the exchange or the party with the short position does not deliver such a contract at expiry. Instead, a cash payment corresponding to the value of the notional deposit is made. This value is determined by the short-term offered rate (e.g., EURIBOR or LIBOR) at expiry.

As for futures in general, single contracts have a nonstationary time series of limited lifetime (e.g., Fung and Leung (1993)). A typical nonstationary effect of IR future contracts is the systematic decrease of mean volatility when moving closer to the expiry (which is fixed in calendar time). In order to study long time series, we have to connect the data from several contracts. Rollover schemes as suggested in Section 2.1.2 are not the most suitable method in the case of IR futures, because several contracts with different expiries trade at the same time with comparable liquidity, unlike what happens in bond futures and other futures markets, where basically only one contract (or at most two) are actively traded at any given time.

In the case of IR futures, the problem is inherently multivariate such that the interplay of several contracts with different expiries but simultaneous high open interest levels cannot be neglected. The method followed here is to infer *implied interest rates* from the prices of futures as discussed in the next section.

2.4.2 Implied Forward Interest Rates and Yield Curves

Implied interest rates have some advantages over IR futures prices. They can be studied in long time series. The behavior of their returns is closer to stationary, although more subtle effects such as local volatility peaks before expiry dates may still remain in time series of implied rates. Implied interest rates can be studied in two different forms:

- Forward interest rates: The interest rate for a period of usually 3 months, with a starting point always shifted to the future by a fixed time interval;
- Yield curve: Spot interest rates for periods starting now. The yield curve is the full term structure of interest rates of different maturities.

Both forms need to be computed as discussed next. Futures prices alone are not sufficient to construct implied spot rates (points on the yield curve), because the IR futures market does not convey any information about the applicable spot rate for the period from the current time to the next futures expiry. The necessity to use data from other instruments (spot IRs) can be avoided by studying forward IRs, with a minimum starting point of 3 months in the future.

There are many methods to construct forward interest rates or full yield curves from IR futures. Instead of presenting the wide field of methods in the literature, we explain the method that was actually used to obtain the results presented in this book, the *polynomial* method. To use the information from IR futures to construct a yield curve of forward (or spot) rates, a timing problem needs to be dealt with.

Futures are defined in terms of the contract expiry date (a fixed quarterly date, every 3 months for the major contracts) and the maturity period of the underlying reference rate (a fixed time interval, usually 3 months), whereas the implied forward rates are defined in terms of fixed time intervals, not in terms of fixed calendar dates. Those time intervals (forward periods) can be written as $[t_{\text{exp}}, t_{\text{exp}} + \Delta T_m]$. They start at time $t_{\text{exp}} = t + T_{\text{exp}}$ (t is the current time) where T_{exp} is called the time-to-start for the implied forward rate and ΔT_m is the maturity[18] period of the notional deposit.

The polynomial method presented here is based on the interpolation of rates between points on the expiry time axis. Polynomials of degree 2 are used for the interpolation. The choice of polynomial rather than linear interpolation is motivated by the seasonal behavior of returns. Forward IR series generated by linear interpolation exhibit some 3-month seasonalities, which are weak but distinctly stronger than those of forward IR series generated by polynomial interpolation. The seasonality of linearly interpolated data turns out to be an artifact due to insufficient modeling within the 3-month interpolation intervals.

A continuously compounded forward interest rate ϱ is assumed and modeled as a function of the time T where T is the size of the time *interval* from the time when the quote was issued to the time point of interest. The annualized implied forward rate r, whose relation to futures prices f is given by Equation 2.3, can be expressed by

$$r = \frac{1 \text{ year}}{T_{\text{end}} - T_{\text{start}}} \left\{ \exp\left[\int_{T_{\text{start}}}^{T_{\text{end}}} \varrho(T)\, \mathrm{d}T \right] - 1 \right\} 100\% \qquad (2.4)$$

where the forward period is from T_{start} to T_{end}. For a futures contract, $T_{\text{start}} = T_{\text{exp}}$ is the expiry time and $T_{\text{end}} = T_{\text{mat}}$ is the maturity time, which terminates at the maturity period, $T_{\text{mat}} = T_{\text{exp}} + \Delta T_m$, with the forward period ΔT_m (often 3 months). The inverse formula is

$$\bar{\varrho} = \frac{\ln\left(1 + \frac{T_{\text{mat}} - T_{\text{exp}}}{1 \text{ year}} \frac{r}{100\%}\right)}{T_{\text{mat}} - T_{\text{exp}}} \qquad (2.5)$$

which gives the mean value $\bar{\varrho}$ of $\varrho(T)$ within the forward period. The function $\varrho(T)$ consists of piecewise, continuously connected, quadratic polynomials

$$\varrho(T) = a\, t^2 + b\, t + c\,, \quad \text{with } t = \frac{2\,T - T_{\text{exp}} - T_{\text{mat}}}{T_{\text{mat}} - T_{\text{exp}}} \qquad (2.6)$$

for $T_{\text{exp}} \le T \le T_{\text{mat}}$ (i.e., $-1 \le t \le 1$). The polynomial coefficients should obey the requirements of Equation 2.4 and all quoted forward rates are reproduced by integration of Equation 2.6. The other requirement is the continuity of $\varrho(T)$ at the

[18] We do not use the term "maturity" as a synonym of "expiry," but we reserve it to denote the duration of deposits, including the underlying deposits of futures contracts. The maturity period of a futures contract thus starts at expiry.

edge points T_{exp} and T_{mat}, where the forward period meets the forward periods of the neighbor contracts. The value of ϱ at the meeting point T_{exp} is determined from the four implied forward rates nearest to T_{exp} (in a regular sequence of 3-month futures contracts) by polynomial interpolation

$$\varrho(T_{exp}) = \frac{9\,(\bar{\varrho}_2 + \bar{\varrho}_3) - \bar{\varrho}_1 - \bar{\varrho}_4}{16} \qquad (2.7)$$

where $\bar{\varrho}$ is computed by Equation 2.5 and its index $(1, 2, 3, 4)$ indicates the position in the series of the four nearest forwards. For instance, $\bar{\varrho}_3$ refers to the forward from T_{exp} to T_{mat}. Equation 2.7 can be interpreted as the interpolation of a cubic polynomial going through the values $\bar{\varrho}_1 \ldots \bar{\varrho}_4$, located at the midpoints of the corresponding forward periods. The same equation if shifted by one forward period leads to $\varrho(T_{mat})$. If a contract at the edge ($\bar{\varrho}_1$ or $\bar{\varrho}_4$ in Equation 2.7) is not available from the data source, we extrapolate

$$\bar{\varrho}_1 = \frac{3\,\bar{\varrho}_2 - \bar{\varrho}_3}{2} \qquad (2.8)$$

and analogously for $\bar{\varrho}_4$. The numerical impact of an extrapolation error is small, as $\bar{\varrho}_1$ and $\bar{\varrho}_4$ have little impact in Equation 2.7. By fulfilling all the mentioned requirements, the coefficients of the polynomial $\varrho(T)$ of Equation 2.6 can be formulated as

$$\begin{aligned} a &= \frac{3}{4}\,[\varrho(T_{exp}) + \varrho(T_{mat}) - 2\,\bar{\varrho}] \qquad (2.9)\\ b &= \frac{1}{2}\,[\varrho(T_{mat}) - \varrho(T_{exp})]\\ c &= \frac{1}{4}\,[6\,\bar{\varrho} - \varrho(T_{exp}) - \varrho(T_{mat})] \end{aligned}$$

where $\bar{\varrho}$ follows from Equation 2.5 ($\bar{\varrho} = \bar{\varrho}_3$, using the indexing of Equation 2.7).

Now we can compute the annualized forward rate r for a given forward period by substituting Equations 2.5 through 2.9 into Equation 2.4. The integration is simple because the integrand, Equation 2.6, is a simple polynomial. The forward period to be considered (from T_{start} to T_{end}) may typically extend over many original forward periods of the futures contracts. In this case, we need to integrate over several piecewise polynomials.

A consequence of the polynomial interpolation is the potential overshooting of the yield curve. If the forward rates implied by a series of IR futures have a maximum somewhere around medium-term expiries, an interpolated forward rate (for a period close to that maximum) may exceed all the implied forward rates corresponding to original futures quotes. The modest overshooting of polynomial interpolation is not undesirable as it leads to a strong reduction of the 3-month seasonalities obtained for a forward IR series generated by linear interpolation (which has no overshooting). We conclude that polynomials with their smooth but sometimes overshooting behavior represent true yield curves better than piecewise linear interpolation with hard corners at the nodes. In our method, overshooting is a local effect—distant contracts cannot affect the behavior of $\varrho(T)$. This is better than some methods based on spline interpolation, where even very distant contracts have an influence on the local behavior.

A special case of overshooting might be "undershooting" of (forward) interest rates. Some parts of $\varrho(T)$ might have values below zero. The method should include an element to avoid strongly negative $\varrho(T)$, although interest rates can move *slightly* below zero under extreme circumstances.

The polynomial method relies on a regular quarterly sequence of expiries. It can be adapted to include also irregular contracts such as contracts based on serial months. The period from the current time to the first expiry cannot be covered by IR futures. We need spot IRs to fill that gap, which is a method described by Müller (1996). After filling the gap, the described methods to compute implied forward interest rates can also be used to compute implied spot rates, simply by choosing $T_{\text{start}} = 0$, which implies the current time. The yield curve consists of a set of implied spot rates with different T_{end}.

There is an interesting application for the yield curve of implied interest rates by comparing the curve derived from IR futures at any point in time with the curve derived from other instruments (such as deposits or over-the-counter forward rate agreements), which allows an investigation of arbitrage opportunities.

Convexity corrections of the yield curve, which represent the difference between futures and forward contracts due to the presence of margining arrangements for futures (collateral accounts, see Burghardt and Hoskins, 1995), are negligibly small in our case, because the futures contracts under consideration are never more than 18 months from expiry.

Time series of forward rates share many properties of FX time series, as studied in later chapters of this book.

2.5 BOND FUTURES MARKETS

2.5.1 Bonds and Bond Futures

Bonds are the dominating financial instrument related to long-term interest rates.[19] There is a wealth of *bonds* issued by governments or individual companies with different credit ratings. Bonds are interest-paying contracts. After a lifetime of several *years*, the capital is paid back to the holder of the contract. Some bonds are complicated financial constructions including special option contracts. The world of bonds is not simple enough to be studied in the form of few, long, consistent time series.

The bond *futures* market, on the other hand, is more standardized. Similar to the short-term interest rate futures discussed in Section 2.4, bond futures are a

[19] There is also a market for interest rate swap transactions with maturities of few years (with a focus on shorter maturities than those of bonds). The IR swap rates of this market constitute yet another source of high-frequency IR data.

liquid financial instrument in the area of interest rate markets, traded at the same exchanges. High-frequency, high-quality intraday data are available in the form of transaction prices, sometimes bid and ask prices, and volume figures. Bond futures markets supply more accurate and more frequent information on bonds than the cash market for bonds. In spite of this, there is little published research on the intraday behavior of bond futures, Ballocchi and Hopman (1997).

Bond futures are futures contracts as discussed in Section 2.1.2. As for most short-term IR futures, there are four settlement months in a year (March, June, September, and December) known as quarterly expiries. The exact settlement and delivery rules can be obtained from the exchanges (or their web sites). A practical introduction to the bond futures markets can be found in LIFFE (1995a,b). Active trading of bond futures is focused on the first two positions which are the contracts with the nearest expiries.

The underlying instruments of bond futures are often government bonds with maturity periods of years, for example the 30-year U.S. Treasury bond futures traded at Chicago Board of Trade (CBOT). This long duration is the main difference from short-term IR futures where the underlying instruments are notional 3-month deposits. Three-month interest rates are strongly influenced by monopolistic players such as the central banks with powers to set short-term rates. In this respect, the bond market and thus the bond futures market is "freer" than the short-term IR futures market. The underlying instrument of some typical bond futures is the cheapest bond(s) available to the exchange under certain conditions, the "cheapest-to-deliver." Unfortunately, the choice of the cheapest-to-deliver can change sometimes, leading to a disruptive behavior of bond futures prices.

For studying long samples, we need to create continuous time series.[20] A theoretically appealing method to construct such a series is to use arbitrage formulas such as in Hull (1993), with input from both short-term and long-term interest rates, as well as from the underlying deliverable instrument, to find a relationship between two successive bond futures contracts. Such an approach, which could be seen as an extended variation of the approach of Section 2.4.2 based on forward IRs, would require data input from several sources and imply considerable methodological efforts. Instead of this, we suggest using rollover schemes, which allow for an analysis based on bond futures data only.

2.5.2 Rollover Schemes

As mentioned in Section 2.1.2, a rollover scheme is a general way to create a continuous time series from the time series of futures contracts with different expiries. The proposed schemes have been mainly applied to bond futures in Ballocchi and Hopman (1997). The schemes are recommended to researchers but not necessarily to traders or investors who pay transaction costs. Investment strategies may also include rollovers, but the different optimization goal leads to different schemes.

[20] As before, the word "continuous" means a consistent behavior as close to stationary as possible. The series should not suffer shocks unrelated to market movements when crossing the contract expiries.

The rollover algorithm must follow an essential economic constraint imposing that the value of the portfolio changes only when the market prices of the individual bond futures move, with no change arising solely from the rollover procedure. When a rollover occurs, the number of new contracts to be bought is calculated so that the total amount of capital invested is constant, that is, the number of new contracts is given by the number of old contracts being sold multiplied by their middle price, divided by the middle price of the new contracts being bought. For bid-ask data, middle prices are defined as means of bid and ask prices.[21] Middle price are used because traders can go long or short; the continuous series should not only reflect one of the two directions.

Two different schemes are presented:

- A simple scheme involving a conversion factor to render continuous the transition from one contract to the next one, at a fixed date before the first contract's expiry.
- The construction of bond futures portfolios with "constant mean time-to-expiry," through a daily partial rollover, whereas the constituent bond futures have a fixed calendar date expiry.

The timing of the rollover determines the character of the obtained continuous time series. It is possible to make a "first position series" or a "second position series" or something in between. Typically there is not sufficient data in the third position to allow for a serious study.

The proposed simple rollover scheme has just one contract in the reference portfolio at once. At a fixed delay D_1 before the expiry of the contract in the portfolio, we roll over the entire holdings to the next expiry. If we start with one contract and the price at the rollover time is F_{1,T_1-D_1}, we can afford to buy a number α_{12} of new contracts at a price F_{2,T_1-D1},[22] where

$$\alpha_{12} = \frac{F_{1,T_1-D_1}}{F_{2,T_1-D1}} \tag{2.10}$$

Clearly the number of contracts we hold at any time can be calculated as the product of the α factors of all past rollovers.

The execution of rollover procedures (even if it happens only in theory, not in real transactions) requires the approximate simultaneous availability of reliable market prices for the two involved contracts. This is not always guaranteed, but it is likely if the rollover time is chosen when both contracts are liquid. The α factors computed by Equation 2.10 are slowly varying over time, often slower than the prices themselves. This fact can be used to take a mean α from those daytimes where simultaneous quotes of both contracts are found, rather than an α determined at only one fixed daytime.

[21] In the case of transaction data, the transaction prices take the role of middle prices.

[22] In reality, we can only buy an integer number of contracts. This does not matter for our theoretical rollover formula: α may be a fraction.

In empirical studies, however, few discontinuities of α factors over time are detected. Ballocchi and Hopman (1997) explain these discontinuities by asynchronicities in the underlying bonds, the "cheapest-to-deliver." If the underlying bonds of two successive contracts do not change exactly at the same time, the α factor is affected by the difference. When analyzing a continuous series, the exact knowledge of the underlying bonds (cheapest-to-deliver, or benchmark bonds) is helpful.

The second scheme based on a "constant time-to-expiry" is to have a portfolio that does not expire at a fixed calendar date, but keeps a constant mean time-to-expiry as time moves on, by means of an appropriate daily rollover procedure.[23] The time-to-expiry (or horizon) h for a portfolio is defined as a weighted average of the time-to-expiry of the constituents. We consider a constant time-to-expiry portfolio consisting at time t of two contract expiries, with a number $\beta_{i,t}$ of contracts in the first expiry, corresponding to time T_i and a number $\gamma_{i+1,t}$ of contracts in the second expiry, corresponding to time T_{i+1}. The time-to-expiry of this reference portfolio is then

$$h = \frac{\beta_{i,t}}{\beta_{i,t} + \gamma_{i+1,t}}(T_i - t) + \frac{\gamma_{i+1,t}}{\beta_{i,t} + \gamma_{i+1,t}}(T_{i+1} - t) \qquad (2.11)$$

Each day we arrange a partial rollover procedure, selling a proportion of the holdings in the first expiry and buying the second one, in order to keep h constant.

Some tools for generating long samples from several contracts through rollover schemes are commercially available, such as the Liffestyle program from LIFFE, see Gwilym and Sutcliffe (1999). This software also offers volume-dependent timing of the rollover (e.g., rolling over when the volume of a new contract overtakes that of an old contract).

2.6 COMMODITY FUTURES

Commodity futures are similar to the futures contracts presented in Section 2.1.2. The settlement at expiry means physical delivery of the underlying commodity. Commodities such as raw materials or agricultural products often exist in different variations and quality levels. Therefore a typical holder of a long position in commodity futures does not want to receive the commodity exactly in the form delivered at expiry. Most commodity futures traders offset their contracts (or roll them over) before expiry, in some markets so early that the second position (the contract with the second next expiry) has a higher liquidity than the first position.

A purpose of commodity futures trading is hedging. A manufacturer confronted with rapidly rising raw material prices is protected by holding some corresponding futures of simultaneously increasing value. Some investors use commodity futures as a vehicle for portfolio diversification.

[23] The related high transaction costs are irrelevant, since this rollover procedure does not need to be executed in practice.

Futures of agricultural commodities may have an irregular schedule of expiry dates due to the seasonality of agricultural production. The cocoa futures market of New York, for example, has five irregularly spaced expiry dates per year, in March, May, July, September, and December.

As in other futures markets, contracts with different expiry dates are not independent. A contract with distant expiry, for instance, cannot be *much* more expensive than a near contract; otherwise traders would buy the near contract and profitably store the commodity afterward in a warehouse. This condition is similar to the condition that forward interest rates cannot be far below zero.

Commodity futures markets are often much smaller than FX or money markets. They are not liquid enough for huge transactions. Large orders often cause considerable *slippage* with immediate price movements to the unfavorable direction.

High-frequency commodity futures data are available from the exchanges and from data vendors. Rollover schemes similar to those of Section 2.5.2 are needed to build long time series from different contracts.

2.7 EQUITY MARKETS

Equity markets are a major source of high-frequency data. The authors of this book have only casually investigated time series from equity markets. Therefore, only a brief description is given.

Equities are traded at stock exchanges of different kinds. Also the instruments derived from equities are exchange-traded. High-frequency data are mainly produced during the opening hours of the exchanges. In some main markets, there is also some electronic trading outside the normal opening hours, which yields some sparse additional data.

High-frequency data are available from the following markets:

- Equity of *individual companies* as traded by stock exchanges. This data type is strongly determined by the specific behavior of an individual firm and some general trends of the market and the economy. Stock splits (e.g. five new equity units replacing one old unit) and dividend payments affect the equity prices, and price series can only be understood with a full account of all these events. Only the most traded individual equities have a data frequency high enough to be called high-frequency.
- Equity *indices*, also called stock indices. These are weighted sums of individual equity prices according to a formula. The basket of equities includes important equities of specific countries or industry sectors. The basket and the weights are adapted from time to time, according to the changing size of the companies. A *performance index* reflects the value of a realistic portfolio of investments according to the basket, including all dividend payments and reinvested profits. It is thus possible to replicate the behavior of an equity index by a real portfolio (it is a better approximation if the index is a performance index). Equity indices represent large segments

of an economy rather than individual companies and their behavior is less erratic than that of individual equities. High-frequency data for the main indices are available and are interesting objects of research. Due to their mathematical definition, they often show a positive autocorrelation of returns at a lag of up to 15 minutes. This may be a consequence of a lag structure between leading main equities and the less liquid equities of the basket.

- Individual equity *futures* or equity index futures are liquid instruments with high-quality, high-frequency data that have been studied by many researchers.
- *Options* also exist for individual equities as well as equity indices. Their implied volatility figures can be investigated by time series methods.

3

TIME SERIES OF INTEREST

An adequate analysis of high-frequency data relies on explicit definitions of the variables under study. In this chapter, we study the common mathematical framework used to analyze these variables.

Some aspects of preparing and preprocessing a time series are rather technical. Readers interested in economic results may prefer to skip the technical Chapters 3 and 4 and continue their reading in Chapter 5.

Researchers conducting their own high-frequency studies may profit from Chapters 3 and 4. If they have no access to preprocessed time series (i.e., cleaned time series with regular spacing in time), they will need the techniques described in these chapters. The literature often ignores technicalities of dealing with irregularly spaced high-frequency data, so we have a good reason to discuss them in two chapters of this book.

3.1 TIME SERIES AND OPERATORS

Many types of time series data can be obtained at high frequency, often intraday, at market tick-by-tick frequency. For most methods, these raw time series are not suitable to work with, because market ticks arrive at random times.[1] The

[1] There are models for the stochastic nature of these times, such as Engle and Russell (1997, 1998).

time series operator formalism developed by Zumbach (1996) and Zumbach and Müller (2001) offers a powerful way to deal with irregularly spaced data. Section 3.3 is based on this formalism and the notations of Zumbach and Müller (2001).

In time series analysis, a first important classification is done according to the spacing of data points in time. Regularly spaced time series are called *homogeneous*, whereas irregularly spaced series are called *inhomogeneous*. The concept of inhomogeneous time series also has to be distinguished from two other concepts, which are the concept of missing observations (where a series is essentially homogeneous with few gaps) and the concept of continuous-time finance (which belongs to theory rather than data sampling). When considering the spacing of data in time, a discussion of the time scale is necessary. Many time series of daily data in finance, for example, have only five observations per week; there are no observations on Saturdays and Sundays. Such a time series is homogeneous only if using a special "business time" scale, which omits weekends (and holidays). Even more sophisticated business time scales can be introduced in order to cope with some characteristics of intraday data such as the seasonality of volatility, (see Chapter 6 and Dacorogna *et al.*, 1993), the heteroskedasticity (Zhou, 1993), or both seasonality and heteroskedasticity (Guillaume *et al.*, 1997; Müller *et al.*, 1993a). Time is denoted by t in Chapter 3, but t may stand for any choice of time scale, not only physical time or clock time. The terms "homogeneous" and "inhomogeneous" have to be understood in the context of the chosen time scale. Inhomogeneous time series by themselves are conceptually simple. The difficulty lies in efficiently extracting and computing information from them. *Time series operators* are a major tool used to transform a raw, inhomogeneous time series to the (homogeneous *or* inhomogeneous) time series of the variable to be analyzed.

In most books on time series analysis, the field of time series is restricted to homogeneous time series.[2] In Section 3.2, we follow this restriction, which induces numerous simplifications, both conceptually and computationally. There, we need only one time series operator type: an operator to transform an inhomogeneous time series to a homogeneous one, type (a) of Figure 3.1.

In Sections 3.3 and 3.4, we follow Zumbach and Müller (2001) and build a computational toolbox for directly and efficiently treating inhomogeneous time series. In practice, this toolbox is attractive enough to be applied to any time series, including homogeneous ones. Given a time series z, such as an asset price, the general point of view is to compute another time series, such as the volatility of the asset, by the application of an operator $\Omega[z]$ where the resulting series stays inhomogeneous with the same time points as the original series. This operator type is called type (b) in Figure 3.1.

[2] Classical textbooks on homogeneous time series are Granger and Newbold (1977); Priestley (1989); Hamilton (1994).

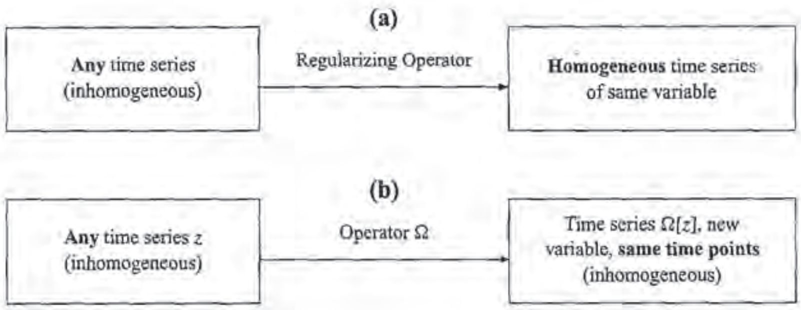


FIGURE 3.1 Different operator types to study time series:

(a) Sampling an inhomogeneous time series at regular time intervals. The resulting homogeneous time series can be treated by standard methods of time series analysis.

(b) Computing a new variable from the initial variable while keeping the initial (inhomogeneous) time points. Example: computing a series of local volatility values from the initial price series.

An important distinction between operators has to be made. We introduce two operator types:

- Microscopic operators depend on the actual sampling of the inhomogeneous time series. Eliminating some random ticks leads to very different results.
- Macroscopic operators extract *average* behaviors of their time series argument. They are essentially immune to small variations of the individual ticks, including adding or eliminating few ticks.

A possible technical definition of macroscopic operators is that they have a well-defined limit when the price quotes become infinitely dense. Practically, if the price quotes are sufficiently dense inside the range of the operator, we are close enough to this limit. For inhomogeneous time series, macroscopic operators are better behaved and more robust than microscopic operators. For homogeneous time series, this distinction is unnecessary because the sampling frequency is fixed and there is no reason to take a continuous-time limit or to formally add or remove ticks. Moreover, because homogeneous time series analysis is based on the backward shift operator $\mathcal{B}$ (which is microscopic), most of the conventional time series analysis becomes unusable for inhomogeneous time series. The classification between operators is further explored by Zumbach and Müller (2001).

Macroscopic operators can be represented by convolutions and are discussed in Section 3.3. The archetype of a macroscopic operator is the exponential moving average (EMA) that computes a moving average with an exponentially decaying weight of the past.

Microscopic operators are presented in Section 3.4. Examples are the time difference δt between ticks (e.g., in $\delta t_j = t_j - t_{j-1}$) and the backward shift operator $\mathcal{B}$ defined in Section 3.4.1.

3.2 VARIABLES IN HOMOGENEOUS TIME SERIES

Basic variables such as the price, the return, the realized volatility, and the spread are defined in Section 3.2. In order to capture the dynamics of the intraday market, some more variables such as the tick frequency are of interest.

3.2.1 Interpolation

Before defining different variables, the generation of homogeneous time series has to be explained. A homogeneous time series, although taken for granted in time series analysis, is an artifact that has to be constructed from the raw data, which is an inhomogeneous series with times t_j and values $z_j = z(t_j)$. The index j refers to the irregularly spaced sequence of the raw series. By utilizing an *interpolation* method, we construct a homogeneous time series with values at times $t_0 + i\,\Delta t$, regularly spaced by Δt, rooted at a time t_0. The index i refers to the homogeneous series.

The time $t_0 + i\,\Delta t$ is bracketed by two times t_j of the raw series

$$j' \; = \; \max(j \mid t_j \le t_0 + i\,\Delta t) \quad , \quad t_{j'} \; \le \; t_0 + i\,\Delta t \; < \; t_{j'+1} \tag{3.1}$$

We interpolate between $t_{j'}$ and $t_{j'+1}$. The two most important interpolation methods are *linear interpolation*

$$z(t_0 + i\,\Delta t) \; = \; z_{j'} + \frac{t_0 + i\,\Delta t - t_{j'}}{t_{j'+1} - t_{j'}} \, (z_{j'+1} - z_{j'}) \tag{3.2}$$

and *previous-tick interpolation* (taking the most recent value),

$$z(t_0 + i\,\Delta t) \; = \; z_{j'} \tag{3.3}$$

which was already proposed by Wasserfallen and Zimmermann (1985).

Both methods which are illustrated by Figure 3.2 have their merits. Previous-tick interpolation respects causality as it exclusively uses information already known at time $t_0 + i\,\Delta t$, whereas linear interpolation uses information from time $t_{j'+1}$, which lies in the *future* of time $t_0 + i\,\Delta t$. When using previous-tick interpolation over a gap (a long period of missing data) in the raw data, a spurious jump of z may be observed at the end of the gap, which may spoil a statistical analysis of extreme returns of z. In this example, linear interpolation would be the appropriate choice. As advocated in Müller *et al.* (1990), linear interpolation is the appropriate method for a random process with identically and independently distributed (i.i.d.) increments. Many statistical studies and model estimations can be alternatively done with both interpolation methods, in practice. The difference between the results indicates the sensitivity to the choice of the interpolation method. The difference is often small, even negligible thanks to high-frequency data. In the empirical studies of the book, the choice of the interpolation method is discussed whenever it matters.

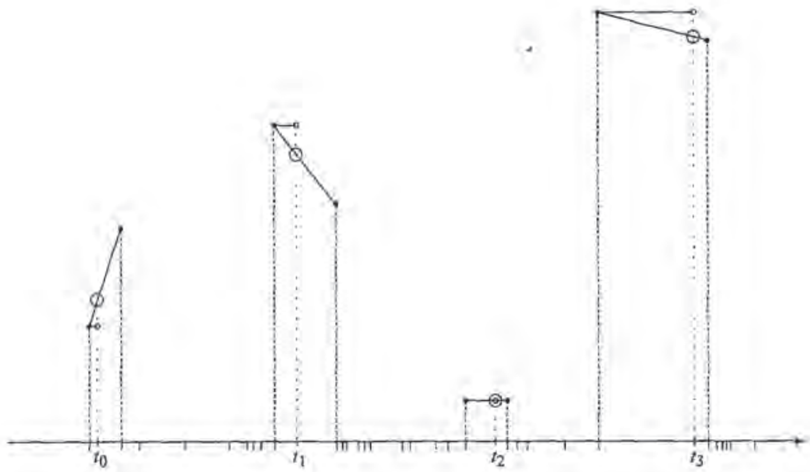


FIGURE 3.2 Interpolation methods to obtain a homogeneous time series: selecting values at equally spaced time points t_i, indicated by dotted vertical lines. The inhomogeneous time sequence of raw observations is indicated by ticks below the horizontal time axis and by dashed vertical lines (only for the observations bracketing the time points t_i). Two important interpolation methods are illustrated by empty circles: linear interpolation (big circles) and previous-tick interpolation (small circles).

The transformation of an inhomogeneous time series to a homogeneous one can also be understood as the result of a special microscopic time series operator which is discussed in Sections 3.3.1 and 3.4.2.

3.2.2 Price

Prices of assets are the most important variables explored in finance. Depending on the market structure and the data supplier, prices are available as quotes in different forms:

- Bid-ask price pairs: p_{bid} and p_{ask}
- Transaction prices (which may or may not be former bid or ask quotes)
- Bid, ask, transaction prices in irregular sequence (not in pairs, not synchronous)
- Middle prices

One individual observation at a time t_j, also in the case of bid-ask pairs, is called a *tick*.

Bid-ask price pairs are discussed first. FX prices and other asset prices, as well as nonprice variables such as spot interest rates and implied volatility figures from option markets, are quoted as bid-ask pairs. The most important variable under study is the logarithmic middle price x. At time t_j, it is defined as

$$x(t_j) = \frac{\log p_{\mathrm{bid}}(t_j) + \log p_{\mathrm{ask}}(t_j)}{2} = \log \sqrt{p_{\mathrm{bid}}(t_j)\ p_{\mathrm{ask}}(t_j)} \qquad (3.4)$$

where t_j is the inhomogeneous sequence of the tick recording times. The variable x may simply be called the *price* in a context where its logarithmic nature is obvious and not explicitly relevant. It is based on the (geometric) average of the bid and ask price rather than either the bid or the ask price alone; this is a better approximation of the true price. The best choice, even better than Equation 3.4, might be a so-called effective price as discussed at the end of this Section 3.2.2.

In the foreign exchange (FX) market, a further advantage of Equation 3.4 is obvious. FX prices can be seen from two sides, the value of the U.S. Dollar (USD) in Japanese Yen (JPY) and the value of the JPY in USD. Equation 3.4 is perfectly antisymmetric: if x is the USD-JPY price, the JPY-USD price is simply $-x$. Statistical results based on absolute differences of x (or volatility) are identical for USD-JPY and JPY-USD. This is a desired property because USD-JPY and JPY-USD are the same market. If the logarithmic transformation was avoided or the logarithm of the arithmetic average of bid and ask was taken instead of Equation 3.4, the antisymmetry would be violated and the statistical results of USD-JPY and JPY-USD would differ. The logarithmic transformation has the additional advantage of making returns (differences of x) dimensionless—that is, independent of the original units in which the price is measured.

In the case of transaction or middle prices, Equation 3.4 is obviously replaced by $x(t_j) = \log p_{\mathrm{transact}}(t_j)$ or $\log p_{\mathrm{middle}}(t_j)$. For certain data types, the logarithmic transformation is less suitable; it is avoided or made in a mathematically different form. For spot interest rates as discussed in Section 2.3.1, the logarithm of the capital increase factor can be taken:

$$x_{\mathrm{bid}}(t_j) = \log[1 + i_{\mathrm{bid}}(t_j)] \qquad (3.5)$$

and analogous for the ask quote, where the interest rate i is inserted as a plain value, not in percent (e.g., 0.05 instead of 5%). Alternative definitions of x are explained whenever they are applied in this book.

The inhomogeneous series $x(t_j)$ can be transformed to a homogeneous time series by using an interpolation method as explained in Section 3.2.1, using Equation 3.2 or 3.3. For the homogeneous series of prices, we use the index i:

$$x(t_i) = x(\Delta t, t_i) = \frac{\log p_{\mathrm{bid}}(t_i) + \log p_{\mathrm{ask}}(t_i)}{2} \qquad (3.6)$$

where t_i is the homogeneous sequence of times regularly spaced by time intervals of size Δt. As already mentioned, t and Δt may refer to any definition of the time scale, not only physical time.

In some markets such as the FX spot market, bid and ask prices are just indicative quotes produced by market makers who are often interested primarily in either the bid or the ask price; the other price acts as a noncompetitive dummy value. This leads to a small error that affects Equation 3.4. Moreover, the quoted spread (ask minus bid price) does not exactly reflect the real spread, which is usually smaller as reported in Goodhart *et al.* (1995).[3] Furthermore, because of transmission delays, it may be, for example, that market maker B enters a quote after market maker A, but that the quote of market maker B is the first to appear on a multi-contributor data feed. Data gaps due to transmission breakdowns become more significant at high frequencies.

All these effects can be modeled in the form of an *effective price* which is closer to the transaction price than the price x of Equation 3.4. In the absence of real transaction prices, we may define an effective price algorithm by looking at the properties of the prices and the market organization. All quotes have a finite lifetime, which is roughly around two minutes during periods of average FX market activity and can strongly vary depending on the market and its state. We can define the effective price as consisting of the best bid and ask quotes available (or the averages of bid and ask) in a time window of the size of a quote lifetime. Another idea for such an algorithm would be to eliminate the negative first-order autocorrelation of the returns present at very high frequencies (see Section 5.2.1). An example of an algorithm for the computation of effective price is given in Bollerslev and Domowitz (1993) where the trade-matching algorithm of the interbank market system "Reuters Dealing 2000" is used. Interestingly, the prices generated by this algorithm exhibit a positive rather than negative first-order autocorrelation. In contrast, Goodhart *et al.* (1995) still obtain a negative first-order autocorrelation, though less pronounced, in their analysis of the Dealing 2000-2 system. In this book, no definition of an effective price is given, but the behavior of prices in the very short term (seconds to minutes) is discussed in several aspects at several places.

3.2.3 Return

The *return* at time t_i, $r(t_i)$, is defined as

$$r(t_i) \;=\; r(\Delta t; t_i) \;=\; x(t_i) - x(t_i - \Delta t) \tag{3.7}$$

where $x(t_i)$ is a homogeneous sequence of logarithmic prices as defined by Equation 3.6, and Δt is a time interval of fixed size. In the normal case, Δt is the interval of the homogeneous series, and $r(t_i)$ is the series of the first differences of $x(t_i)$. If the return interval is chosen to be a multiple of the series interval, we obtain overlapping intervals as discussed in Section 3.2.8. Returns are sometimes also called price changes.

The return is usually a more suitable variable of analysis than the price, for several reasons. It is the variable of interest for traders who use it as a direct measure of the success of an investment. Furthermore, the distribution of returns

[3] In their one-day study of real transaction prices, Goodhart *et al.* (1995) found that although the actual spread is usually within the quoted spread, it could be larger in highly volatile periods.

is more symmetric and stable over time than the distribution of prices. The return process is close to stationary whereas the price process is not.

3.2.4 Realized Volatility

The *realized volatility* $v(t_i)$ at time t_i is computed from historical data and it is also called *historical volatility*. It is defined as

$$v(t_i) = v(\Delta t, n, p; t_i) = \left[\frac{1}{n} \sum_{j=1}^{n} |r(\Delta t; t_{i-n+j})|^p \right]^{1/p} \quad (3.8)$$

where the regularly spaced returns r are defined by Equation 3.7, and n is the number of return observations. There are two time intervals, which are the return interval Δt, and the size of the total sample, $n\Delta t$. The exponent p is often set to 2 so that v^2 is the variance of the returns about zero. In many cases, a value of $p = 1$ is preferred, although p may also be a fraction, $p > 0$. The choice of p is further discussed below.

In order to compute realized volatility, the return interval, Δt and a sample of length $n\Delta t$ need to be chosen. By inserting $\Delta t = 10$ min in Equation 3.8, one can compute the volatility of regularly spaced 10-min returns. One important issue is that many users want their realized volatility in *scaled* form. Although the volatility may be computed from 10-min returns, the expected volatility over another time interval (e.g., 1 hr or 1 year) may also be calculated. Through a Gaussian scaling law, $v^2 \propto \Delta t$, the following definition of scaled volatility is obtained:

$$v_{\text{scaled}} = \sqrt{\frac{\Delta t_{\text{scale}}}{\Delta t}}\, v \quad (3.9)$$

The most popular choice of the scaling reference interval Δt_{scale} is $\Delta t_{\text{scale}} = 1$ year. If this is chosen, v_{scaled} is called an *annualized* volatility, v_{ann}:

$$v_{\text{ann}} = \sqrt{\frac{1\ \text{year}}{\Delta t}}\, v \quad (3.10)$$

Practitioners often use annualized volatility in percent (multiplying v_{ann} by 100%). Typical annualized volatility values for some FX rates are around 10%.

In practice, various volatility definitions may lead to confusion. Terms such as "one-day historical volatility" should be avoided because they do not express whether "one day" refers to the return measurement interval Δt, the sample size $n\Delta t$ or the scaling reference interval Δt_{scale}. In order to clarify this, we give a detailed recipe to compute realized volatility in practice:

- Consider and choose *three* time intervals:
  - The time interval of return observations, Δt

- The sample size $n\Delta t$ (the number n of return observations)
- The scaling interval Δt_{scale}, e.g., 1 year if annualized volatility is desired

- Choose the exponent p of Equation 3.8 (often 2 or 1 as discussed below) and the basic time scale of the computation. Instead of physical time, a business time omitting the weekends may be used.
- Compute realized volatility according to Equation 3.8.
- Scale the result by applying Equation 3.9. If $\Delta t_{\text{scale}} = 1$ year, this means annualization.
- If the volatility has to be expressed in percentage, multiply the result by 100%.

A traditionally popular choice of realized volatility is the annualized volatility of daily returns on a yearly sample (using $p = 2$): $\Delta t = 1$ working day, $n \approx 250$, sample size $n\Delta t = 250$ working days ≈ 1 year, $\Delta t_{\text{scale}} = 1$ year. It should be noted that a business time scale with approximately 250 working days per year is used in this example instead of physical time.

The following examples illustrate the computation of realized volatility. Suppose we have regularly spaced 10-min returns from January 1, 1998 to December 31, 1999. This two-year sample has a total of $n = 105120$ return observations. From this data, we want to compute realized volatility in three forms:

1. Volatility of 10-min returns on May 12, 1999, scaled to one day. The return interval is $\Delta t = 10$ min, $n = 144$, and therefore, the sample size is $n\Delta t = 1440$ min $= 1$ day. v is computed from 144 returns according to Equation 3.8. To obtain the desired scaling, Equation 3.9 is used to compute $v_{\text{scaled}} = \sqrt{1\ \text{day}/10\ \text{min}}\, v = \sqrt{144}\, v = 12\, v$.
2. Annualized volatility of 10-min returns over the whole sample where $\Delta t =$ 10 min. All return observations, i.e., $n = 105120$ are used to calculate v according to Equation 3.8. The sample size is $n\Delta t = 1051200$ min $=$ 2 years. To obtain annualization according to Equation 3.10, $v_{\text{ann}} = \sqrt{1\ \text{year}/10\ \text{min}}\, v = \sqrt{52596}\, v \approx 229.3\, v$ is computed. This reflects the fact that an average year contains about 365.25 days and thus 52596 10-min intervals. Note that physical time is used in this particular example and weekends are not omitted.
3. Annualized volatility of 20-min returns over the whole sample. This is analogous to the example above, except for the different time interval $\Delta t =$ 20 min. The 20-min returns R_j can be obtained by taking the sum of two 10-min returns: $R_1 = r_1 + r_2$, $R_2 = r_3 + r_4$, and so on.[4] The number of R_j observations is half that of the 10-min returns: $n = 52560$. The sample size is $n\Delta t = 52560 \times 20$ min $= 2$ years, as above. v is computed

[4] An alternative scheme to obtain 20-min returns from 10-min returns would be as follows: $R_1 = r_1 + r_2$, $R_2 = r_2 + r_3$, $R_3 = r_3 + r_4$ and so on. This scheme leads to overlapping 20-min returns and will be treated in Section 3.2.8.

from Equation 3.8 from the 20-min returns R_j instead of r_j. To obtain annualization according to Equation 3.10, $v_{\text{ann}} = \sqrt{1\ \text{year}/20\ \text{min}}\ v = \sqrt{26298}\ v \approx 162.2\ v$ is computed. One average year contains 26298 20-min intervals, hence the annualization factor.

All of these examples are static and based on calculating realized volatility values at a fixed time t_i. Of course, we can treat realized volatility as a time series and compute it for a sequence of time points: $t_i, t_{i+1}, t_{i+2}, \ldots$. For each of these realized volatility computations, the sample is shifted by one interval Δt.

The concept of historical or realized volatility is rather old. We find it already in Taylor (1986)[5] and in early high-frequency studies such as Müller *et al.* (1990). We distinguish three main types of volatility:

- Realized volatility, also called historical volatility: determined by past observations by a formula such as Equation 3.8.
- Model volatility: a virtual variable in a theoretical model such as GARCH or stochastic volatility (but there may be means to estimate this variable from the data).
- Implied volatility: a volatility forecast computed from market prices of derivatives such as options (see e.g., Cox and Rubinstein, 1985), based on a model of the underlying process such as the log-normal random walk assumed by Black and Scholes (1973).

The *term* "realized volatility" has recently been popularized by Andersen *et al.* (2000) and others. By exploring realized volatility, Andersen *et al.* (2000) show that this is more than a conveniently measured quantity; it can also be used for process modeling.

An alternative definition of realized volatility is

$$v'(t_i) \;=\; \left\{ \frac{1}{n-1} \sum_{j=1}^{n} \left| r(\Delta t;\, t_{i-n+j}) - \frac{1}{n} \left[\sum_{k=1}^{n} r(\Delta t;\, t_{i-n+k}) \right] \right|^{p} \right\}^{1/p} \tag{3.11}$$

For $p = 2$, this is the standard deviation of the returns about the sample mean. This definition is popular in portfolio analysis where the risk is measured in terms of deviations of the return from the average. In most other applications such as risk management and in the examples of this book, Equation 3.8 is preferred to Equation 3.11. The two definitions essentially differ only in the presence of a strong linear drift (i.e., if the returns have an expectation far from zero).

[5] There, absolute values of returns and squared returns were explicitly introduced as proxies of volatility in autocorrelation studies. Note that these quantities are special cases of Equation 3.8 with $n = 1$ and $p = 1$ (absolute return) or $p = 2$ (squared return).

Realized volatility $v(t_i)$ is based on a homogeneous series of returns as defined by Equation 3.7 and is a homogeneous time series in its own right. As an alternative, we can also compute a homogeneous series of realized volatility based on overlapping returns (see Section 3.2.8) or directly compute volatility from an inhomogeneous series with the help of convolution operators (see Section 3.3.11).

The parameters of Equation 3.8 have to be carefully chosen. A large exponent p gives more weight to the tails of the distribution. If p is too large, the realized volatility may have an asymptotically infinite expectation if returns have a heavy-tailed density function. In practice, p should stay below the tail index of the distribution, which is empirically estimated to be around 3.5 for typical high-frequency FX data (as explained in Section 5.4). The fourth moment of the return distribution often diverges. Moreover, there are studies where realized volatility appears in the squared form (as in autocorrelation studies of volatility). There, p should be limited to the *half* of the tail index. The empirical autocorrelation of squared returns is of little relevance. Instead, autocorrelation studies can be made with absolute values of returns ($p = 1$), as already done by Taylor (1986), Müller *et al.* (1990) and, Granger and Ding (1995).

The choice of Δt and n is also important. Given a constant total sample size $T = n\Delta t$, Andersen *et al.* (2001) recommend choosing Δt as small as possible. This means a large number n of return observations and thus high precision and significance. However, realized volatility results become *biased* if Δt is chosen to be too small, as found by Andersen *et al.* (2000). Therefore, the best choice of Δt is somewhere between 15 min and 2 hr, depending on the market and the data type. The bias has several implications, among them the negative short-term autocorrelation found for some financial data (see Section 5.2.1). Corsi *et al.* (2001) propose a bias-corrected realized volatility with Δt around 5 min, in order to maintain the high precision gained due to a large n. Interval overlapping is a further method to make realized volatility more precise by a limited amount (see Section 3.2.8).

A more fundamental question has to be discussed. Does a realized volatility with a constant $T = n\Delta t$ essentially stay the same if the time resolution, Δt, is varied? "Coarse" realized volatility (with large Δt) predicts the value of "fine" volatility (with small Δt) better than the other way around, as discussed in Section 7.4.1. This lead-lag effect indicates that the dynamics of volatility are complex, and realized volatility with one choice of Δt is not a perfect substitute for realized volatility with another value of Δt.

The relative merits of realized, modeled, and implied volatility are discussed at several places in this book. For low-frequency (including daily) data, models such as GARCH (Bollerslev, 1986) and option markets may yield volatility estimates that are as good or better than realized volatility. For high-frequency, intraday data, realized volatility is superior. Intraday data cannot be described by one homogeneous GARCH model because of the seasonality and heterogeneity of the markets, as shown by Guillaume *et al.* (1994) and Gençay *et al.* (2001c, 2002).

3.2.5 Bid-Ask Spread

In bid-ask price pairs, the ask price is higher than the bid price.[6] The bid-ask spread is their difference. A suitable variable for research studies is the *relative spread* $s(t_j)$:

$$s(t_j) \;=\; \log p_{\mathrm{ask}}(t_j) - \log p_{\mathrm{bid}}(t_j) \qquad (3.12)$$

where j is still the index of the original inhomogeneous time series. This definition has similar advantages as the definition of the logarithmic price x, Equation 3.4. The nominal spread $(p_{\mathrm{ask}} - p_{\mathrm{bid}})$ is in units of the underlying price, whereas the relative spread is dimensionless; relative spreads from different markets can directly be compared to each other. In the FX market, another advantage is obvious. If the relative spread of USD-JPY, for example, is s, the relative spread of JPY-USD is also s, because the roles of bid and ask are interchanged. Results of spread studies are invariant under inversion of the rate. Other spread definitions do not have this perfect symmetry. The relative spread is sometimes just called the "spread" if its relative nature is obvious from the context.

For spot interest rates, we can adapt the relative spread definition in the same sense as Equation 3.5:

$$s(t_j) \;=\; \log[1 + i_{\mathrm{ask}}(t_j)] - \log[1 + i_{\mathrm{bid}}(t_j)] \qquad (3.13)$$

The relative spread is a positive bounded quantity that has a strongly skewed distribution. This can be a problem for certain types of analysis. A further transformation leads to the "log spread," $\log s(t_j)$. For bid-ask prices, the log spread is

$$\log s(t_j) \;=\; \log[\log p_{\mathrm{ask}}(t_j) - \log p_{\mathrm{bid}}(t_j)] \qquad (3.14)$$

Müller and Sgier (1992) have shown that its distribution is much less skewed and closer to symmetric than the distribution of s.

The bid-ask spread reflects the transaction and inventory costs and the risk of the institution that quotes the price. On the side of the traders who buy or sell at a quoted price, the spread is the only source of costs as intraday credit lines on the foreign exchange markets are free of interest.[7] The spread can therefore be considered as a good measure of the amount of friction between different market participants, and thereby as a measure of market efficiency. The relatively high efficiency of the major FX spot markets is reflected by the small average size of s.

[6] Some data sources have prices of minor markets in the spurious technical form of bid-ask pairs with bid = ask, presumably because only one of the two prices is available at a time. These quotes are not true bid-ask pairs.

[7] A trader taking a forward position will, of course, have to pay the interest on his or her position between the trade and the settlement as well as the additional spread on the forward rate.

In markets with indicative quotes from different market makers, individual spreads $s(t_j)$ are often affected by individual preferences of market makers and by habits of the market (see Section 5.6.5). Therefore, a homogeneous time series of spreads $s(t_i)$ generated by interpolation contains a rather high level of noise. A more suitable alternative is to compute *average* spreads within time windows and to build a homogeneous time series of these average spreads.

3.2.6 Tick Frequency

The *tick frequency* at time t_i, $f(t_i)$, is defined as

$$f(t_i) \;=\; f(\Delta t; t_i) \;=\; \frac{1}{\Delta t} N\{x(t_j) \mid t_i - \Delta t \;<\; t_j \;\leq\; t_i\} \qquad (3.15)$$

where $N\{x(t_j)\}$ is the counting function and Δt is the size of the time interval in which ticks are counted.

The "log tick frequency," $\log f(t_i)$, has been found to be more relevant in Demos and Goodhart (1992). We can also define the average time interval between ticks, which is simply the inverse tick frequency, $f^{-1}(t_i)$. Tick frequency can also be computed by a time series operator as explained in Section 3.4.5.

The tick frequency is sometimes taken as a proxy for the transaction volume on the markets. As the name and location of the quoting banks are also available, the tick frequency is also sometimes disaggregated by banks or countries. However, equating tick frequency to transaction volume or using it as a proxy for both volume and strength of bank presence suffers from the following problems. First, although it takes only a few seconds to enter a price quotation in the terminal, if two market makers happen to simultaneously enter quotes, only one quote may appear on the data collector's screen; second, during periods of high activity, some operators may be too busy to enter the quote into the system; third, a bank may use an automatic system to publish prices to advertise itself on the market; fourth, the representation of the banks depends on the coverage of the market by data vendors such as Reuters or Bridge. This coverage is changing and does not entirely represent the whole market. For example, Asian market makers are not as well covered by Reuters as their European counterparts; they are more inclined to contribute to the local financial news agencies such as Minex. Big banks have many subsidiaries; they may use one subsidiary to quote prices made by a market maker in another subsidiary on another continent. Quotes from differently reliable and renowned sources have very different impacts on the market. For all these reasons, we should be cautious when drawing conclusions on volume or market share from tick frequency.

3.2.7 Other Variables

A set of other variables of interest are as follows:

- The "realized skewness" of the return distribution. Roy (1952) evaluates the extreme downside risk in portfolio optimization in terms of the cubic root of the third moment of returns. The skewness of returns can also be measured by a time series operator (see Section 3.3.12).

- The volatility ratio, the ratio of two volatilities of different time resolutions: $v_{\text{ann}}(m\Delta t, n, p)/v_{\text{ann}}(\Delta t, mn, p)$, based on Equations 3.8 and 3.10, with an integer factor $m > 1$. This is a generalization of the variance ratio studied in Lo and MacKinlay (1988), Poterba and Summers (1988) and Campbell *et al.* (1997). The volatility ratio is around 1 for a Brownian motion of x, higher if x follows a trend, lower if x has mean-reverting noise. The volatility ratio (or an analogous volatility difference) is thus a tool to detect trending behavior.

- The direction change indicator, counting the number of essential trend reversals within a time interval as defined by Guillaume *et al.* (1997).

3.2.8 Overlapping Returns

Some variables, notably returns, are related to time intervals, not only single time points. When statistically investigating these variables, we need many observations. The number of observations can be increased by choosing overlapping intervals. For returns, a modified version of Equation 3.7 is used:

$$r_i = r(t_i) = x(t_i) - x(t_i - m\,\Delta t) = x_i - x_{i-m} \qquad (3.16)$$

where t_i is again a regular sequence of time points (for any choice of the time scale), separated by intervals of size Δt. The interval of the return, however, is $m\Delta t$, an integer multiple of the basic interval Δt. If r_i is considered for every i, we obtain a homogeneous series of overlapping returns with the overlap factor m. The corresponding series of nonoverlapping returns would be $r_m, r_{2m}, r_{3m}, \ldots$.

Figure 3.3 illustrates the concept of overlapping intervals. Does a statistical study gain anything from using overlapping as opposed to non-overlapping returns? The number of observations can be increased by using overlapping intervals with a growing overlap factor m, thereby keeping the return interval $m\Delta t$ constant. At the same time, neighboring return observations become increasingly dependent. Thus a gain in statistical significance is not obvious. The problem was discussed in Hansen and Hodrick (1980), where a method of estimating parameters and their significance limits from overlapping observations has been developed and applied. In Dunis and Keller (1993), a "panel regression" technique is presented and applied where the overlapping observations are grouped in several nonoverlapping series with phase-shifted starting points.

Müller (1993) has investigated this question under the simplifying assumption that x is drawn from an identical and independent distribution

$$r'_i = x(t_i) - x(t_{i-1}) \in \mathcal{N}(0, \sigma^2) \qquad (3.17)$$

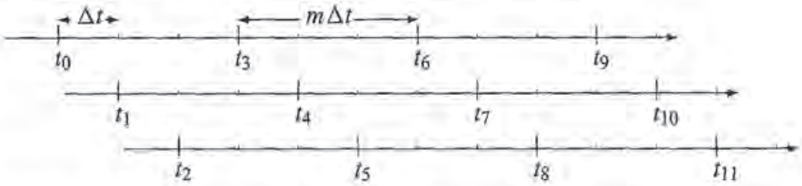


FIGURE 3.3 An overlapping scheme of time intervals to compute a homogeneous time series of overlapping returns. All returns are measured over a time interval of size $m\Delta t$. The intervals on the top time axis alone are not overlapping; an overlapping scheme arises when the phase-shifted intervals on the two lower time axes are added. In this example, the overlap factor is $m = 3$.

where the r'_i are independent. The investigated returns are sums of small-interval returns:

$$r_i = \sum_{i'=1}^{m} r'_{i-m+i'} \tag{3.18}$$

The conclusion of Müller (1993) is that the method of overlapping leads to a distinct but not overwhelming increase of precision and significance in most applications. Some main derivations and results of that study are presented here. There is one special case where overlapping does not help, which is the empirical *mean* of returns. Assume a large sample with n nonoverlapping intervals, covering the mn small intervals of size Δt with returns from r'_1 to r'_{mn}. The same information can be used to compute the $m(n-1)+1$ overlapping returns from r_m to r_{mn}. The mean of these overlapping returns is

$$\frac{1}{m\,(n-1)+1}\sum_{i=m}^{mn} r_i = \frac{1}{m\,(n-1)+1}\sum_{i=m}^{mn}\sum_{i'=1}^{m} r'_{i-m+i'} = \tag{3.19}$$

$$\frac{1}{m\,(n-1)+1}\sum_{i=1}^{m-1}[i\,(r'_i + r'_{mn+1-i})] + \frac{m}{m\,(n-1)+1}\sum_{i=m}^{m(n-1)+1} r'_i$$

The corresponding nonoverlapping mean is

$$\frac{1}{n}\sum_{i=1}^{n} r_{mi} = \frac{1}{n}\sum_{i=1}^{mn} r'_i \tag{3.20}$$

by using Equation 3.18. A comparison to the last term of Equation 3.19 shows that both means are essentially equal for $n \gg 1$; they are based on the sum of the small-interval returns r'. The remaining difference vanishes with n^{-1} in the large-sample limit, whereas the error of both means is proportional to $n^{-0.5}$ in the same limit.[8] Overlapping remains a valid method, but it is unable to reduce the error of the empirical mean.

[8] Müller (1993) has more details of this analysis.

Applications other than the computation of mean returns are more interesting. The most important example is realized volatility. A special version of Equation 3.8 is used:

$$v_i^p = v^p(t_i) = \frac{1}{N} \sum_{j=1}^{N} |r_{i-N+j}|^p \tag{3.21}$$

where the returns r_i are defined by Equation 3.16 and the returns are overlapping if $m > 1$. An analytical exploration of this realized volatility is possible under the assumption of Equation 3.17 and the special choice $p = 2$. By substituting Equation 3.18, we obtain

$$v_i^2 = \frac{1}{N} \sum_{j=1}^{N} \left(\sum_{j'=1}^{m} r'_{i-N-m+j+j'} \right)^2 \tag{3.22}$$

The expectation value of v_i^2 can be computed by expanding this expression where the cross products of r' vanish due to the iid assumption. The result is

$$\mathrm{E}[v_i^2] = m\,\sigma^2 \tag{3.23}$$

This is the theoretical expectation of the squared return of the Brownian motion over an interval of size $m\Delta t$. Equation 3.21 is thus an unbiased estimator, at least for $p = 2$.

The realized volatility v_i^2 of Equation 3.21 has an error whose variance is

$$\mathrm{E}\{[v_i^2 - m\,\sigma^2]^2\} = \mathrm{E}[v_i^4] - m^2\,\sigma^4 = \tag{3.24}$$
$$\frac{1}{N^2}\,\mathrm{E}\left\{ \left[\sum_{j=1}^{N} \left(\sum_{j'=1}^{m} r'_{i-N-m+j+j'} \right)^2 \right]^2 \right\} - m^2\,\sigma^4$$

where Equations 3.22 and 3.23 have been used. The further computation of this expression is somewhat tedious because of the two squares. A long sum is obtained where each term is a constant times four factors of the type r'_i. The expectation values of these terms follow from the Gaussian i.i.d. assumption: (1) r'^4_i has an expectation of $3\sigma^4$ (the fourth moment of the Gaussian random variable), (2) $r'^2_i r'^2_j$ has an expectation of σ^4 (if $i \neq j$), and (3) each term with a factor x_i to an odd power has the expectation zero. The resulting variance of the error is

$$\mathrm{E}\{[v_i^2 - m\,\sigma^2]^2\} = \frac{2\,m\,(2\,m^2+1)\,\sigma^4}{3\,N} \left[1 - \frac{m\,(m^2-1)}{2\,(2\,m^2+1)\,N} \right] \tag{3.25}$$

for $N \geq m$.

This has to be compared to the corresponding error of an overlap-free computation from the same sample. There are only n nonoverlapping return observations (with $N = mn$). The variance of the error is

$$E\left\{\left[\frac{1}{n}\sum_{j=1}^{n} r_{mj}^2 - m\,\sigma^2\right]^2\right\} = \frac{2\,m^2\,\sigma^4}{n} = \frac{2\,m^3\,\sigma^4}{N} \tag{3.26}$$

This has been computed as a special case of Equation 3.25 (case $m = 1$ with only n observations, but with a variance $m\sigma^2$ where m is the original overlap factor).

Equations 3.25 and 3.26 are now compared. The ratio of the two error variances is

$$\frac{E\{[v_i^2 - m\,\sigma^2]^2\}}{E\{[\frac{1}{n}\sum_{j=1}^{n} r_{mj}^2 - m\,\sigma^2]^2\}} \approx \frac{2}{3} + \frac{1}{3\,m^2}\,, \quad \text{for } N \gg m \tag{3.27}$$

This ratio is ≤ 1, so overlapping is indeed a means to reduce the stochastic error of realized volatility to a certain extent. In the limit of a very high overlapping factor m, the error variance is reduced to a minimum of two-thirds of the value without overlapping. An overlap factor of only $m = 2$ already reduces the error variance to 75%.

This finding can also be formulated by defining the *effective number of observations*, n_{eff}. This is the number of nonoverlapping observations that would be needed to reach the same error variance as that based on overlapping observations. From Equations 3.25 and 3.26, we obtain

$$n_{\text{eff}} = \frac{3\,m\,N}{2\,m^2 + 1} = \frac{3\,m^2}{2\,m^2 + 1}\,n\,, \quad \text{for } n \gg 1 \tag{3.28}$$

This can be expressed as a rule of thumb. Using the method of overlapping enhances the significance of realized volatility like adding up to 50% of independent observations to a nonoverlapping sample.

In Müller (1993), an analogous study was made for "realized covariance," the empirically measured covariance between two time series, based on simultaneous overlapping returns of both series. The result is similar. The estimator based on overlapping returns is unbiased and has a reduced error variance. The effective number of observations is again given by Equation 3.28.

In two cases, we have found an error reduction due to overlapping intervals. This provides a motivation for a general use of overlapping returns, also in other cases such as realized volatility with another exponent p (see Equation 3.8) and in other studies such as the analysis of the distribution of returns. The user has to be aware of and to account for the serial dependence due to overlapping and its possible effects on the results, as we have done in the derivations presented earlier. The increase in significance can roughly be expected to be as given by Equation 3.28, with some deviations due to the non-Brownian nature of the raw data. Heavy-tailed distributions, serial dependence, and heteroskedasticity as well as the choice of p may affect the behavior of the stochastic error.

The statistical significance can alternatively be increased by choosing return intervals shorter than $m\Delta t$. However, these short-term returns would be a different object of study. The technique of overlapping has the advantage of leaving the object of study unchanged while increasing the precision.

3.3 CONVOLUTION OPERATORS

The original inhomogeneous data can be processed by convolution operators to build new inhomogeneous time series. This approach, developed by Zumbach and Müller (2001), is fundamentally different from the construction of homogeneous time series as discussed in Section 3.2. A set of basic convolution operators is defined that can be combined to compute more sophisticated quantities, for example, different kinds of volatility or correlation. A few stylized properties of these operators are explored, but the main emphasis is to build a sufficient vocabulary of operators well suited to high-frequency data analysis.

In this process, we should keep in mind a few important considerations:

- The computations must be efficient. Even if powerful computers are becoming cheaper, typical tick-by-tick data in finance are 100 or even 10,000 times more dense than daily data. Clearly, we cannot afford to compute a full convolution for every tick. For this reason, our basic workhorse is the exponential moving average (EMA) operator, which can be computed very efficiently through an iteration formula. A wealth of complex but still efficient operators can be constructed by combining and iterating the basic operators.
- A stochastic behavior is the dominant characteristic of financial processes. For tick-by-tick data, it is not only the values but also the time points of the series which are stochastic. In this random world, pointwise values are of little significance and we are more interested in average values inside intervals. Thus the usual notion of return also has to be changed. With daily data, a daily return is computed by Equation 3.7, as a pointwise difference between the price today and the price yesterday. With high-frequency data, a better definition of the daily return may be the difference between the average price of the last few hours and an average price from one day ago. In this way, it is possible to build smooth variables well suited to random processes. The calculus has to be revisited in order to replace pointwise values by averages over some time intervals.
- Analyzing data typically involves a characteristic time range; a return $r[\tau]$, for example, is computed on a given time interval τ. With high-frequency data, this characteristic time interval can vary from a few minutes to several weeks. This is taken care of by making explicit all of these time range dependencies in the formulation of operators.

- We usually want smooth operators with smooth kernels (weighting functions of moving averages). A simple example of a discontinuous operator is an average with a rectangular weighting function, say of range τ. The second discontinuity at "now$-\tau$," corresponding to forgetting events, creates unnecessary noise. Instead, we prefer kernels with a smooth decay to zero. Only at t = now, we often prefer a jump in the kernel form. This jump gives a positive weight to the last piece of information and thus a rapid response in real time. For a discontinuous kernel, the weight at t = now is inversely proportional to the range of the operator. Therefore, there is a trade-off between a fast reaction, which has more noise, and a smooth average behavior with a slow reaction time. Besides this fundamental noise created by the advance of events, it is better to have continuous and smooth operators.

The generalization to inhomogeneous time series introduces a number of technical peculiarities. In this Section 3.3, only macroscopic operators are treated, which, because of their time-translation invariance, can be represented by convolutions. A convolution is defined as an integral, therefore the series should have representation in continuous time. Actual data is known only at discrete sampling times, so some interpolation needs to be used in order to properly define the convolution integral. The same problem is present when constructing an artificial homogeneous time series from inhomogeneous data as in Section 3.2.1. Another technical peculiarity originates from the fact that our macroscopic operators are ultimately composed of iterated moving averages. All such EMA operators have noncompact kernels where the kernels decay exponentially, but strictly speaking they are positive. This implies an infinite memory; a build-up must be done over an initialization period before the error of an operator value becomes negligible.

The examples of Sections 3.3 and 3.4 are from the foreign exchange market. The data set is USD-CHF for the week of Sunday, October 26, to Sunday, November 2, 1997. This week has been selected because on Tuesday, October 28, some Asian stock markets crashed, causing turbulences in many markets around the world, including the FX market. Yet the relation between a stock market crash originating in Asia and the USD-CHF foreign exchange rate is quite indirect, making this example interesting. The prices of USD-CHF for the example week are plotted in Figure 3.4. When not specified otherwise, all figures from Figure 3.4 to 3.17 display quantities for the same example week. All of these figures have been computed using high-frequency data. The results have been sampled each hour using linear interpolation. The computations have been done in physical time, therefore exhibiting the full daily and weekly seasonalities contained in the data.

Finally, we want to emphasize that the techniques presented in this section are suitable for application to a wide range of statistical computations in finance such as in risk management. An early application can be found in Pictet *et al.* (1992) and a recent application is in Zumbach *et al.* (2000).

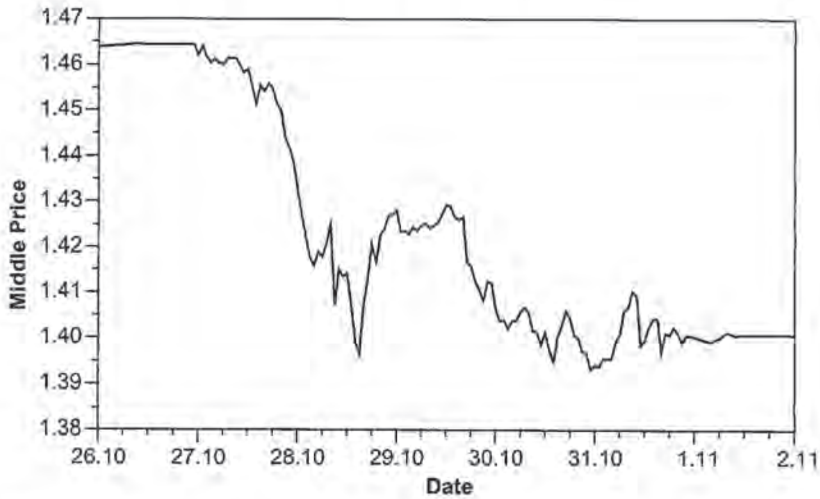


FIGURE 3.4 The FX rate USD-CHF for the week of Sunday, October 26, to Sunday, November 2, 1997. The high-frequency data are sampled hourly, using linear interpolation with geometric middle price $\sqrt{\text{bid} \cdot \text{ask}}$.

3.3.1 Notation Used for Time Series Operators

For time series operators in Sections 3.3 and 3.4, we use a suitable notation that sometimes differs from the conventions used for homogeneous time series. The letter z is used to represent a generic time series. The elements or ticks, (t_j, z_j), of a time series z consist of a time t_j and a scalar value z_j. As everywhere in Chapter 3, t may stand for any (business) time scale, not only physical time. The generalization to multivariate inhomogeneous time series is fairly straightforward (except for the business time scale aspect) and will not be discussed. The value $z_j = z(t_j)$ and the time point t_j constitute the j-th element of the time series z. The sequence of sampling (or arrival) times is required to be growing, $t_j > t_{j-1}$. The strict inequality is required in a true univariate time series and is theoretically always true if the information arrives through one channel. In practice, the arrival time is known with finite precision, say of a second, and two ticks may well have the same arrival time. Yet for most of the formulae that follow, the strict monotonicity of the time process is not required. In the special case where the time series is homogeneous, the sampling times are regularly spaced, $t_i - t_{i-1} = \delta t$. If a time series depends on some parameters θ, these are made explicit between square brackets, $z[\theta]$.

An operator Ω, from the space of time series into itself, is denoted by $\Omega[z]$, as already illustrated by Figure 3.1(b). The operator may depend on some parameters $\Omega[\theta; z]$. The value of $\Omega[z]$ at time t is $\Omega[z](t)$. For linear operators, a product notation Ωz is also used. The average over a whole time series of length T is denoted by $\mathrm{E}[z] := 1/T \int dt\, z(t)$. For the probability density function (pdf) of z, we use $p(z)$. A synthetic regular (or homogeneous) time series (RTS), spaced by δt, derived from the irregular time series z, is denoted by RTS[δt; z]. For a standardized time series for z, we use the notation $\hat{z} = (z - \mathrm{E}[z])/\sigma[z]$ and $\sigma[z]^2 = \mathrm{E}\left[(z - \mathrm{E}[z])^2\right]$. The letter x is used to represent the logarithmic middle price as defined by Equation 3.4.

3.3.2 Linear Operator and Kernels

We focus on operators with the following useful properties:

- Linear operators, where $\Omega[z_1 + c\, z_2] = \Omega[z_1] + c\, \Omega[z_2]$
- Time-translation invariant operators, where $\Omega[z(t - \Delta t)](t) = \Omega[z(t)](t - \Delta t)$
- Causal operators, where $\Omega[z](t)$ exclusively depends on information already know at time t. If $\Omega[z](t)$ depends on future events after t, it does not respect causality at time t and is noncausal.

An operator with all these three properties can be represented by a convolution with a kernel $\omega(t)$:

$$\Omega[z](t) = \int_{-\infty}^{t} dt'\, \omega(t - t')\, z(t') = \int_{0}^{\infty} dt'\, \omega(t')\, z(t - t') \tag{3.29}$$

The causal kernel $\omega(t)$ is defined only on the positive semiaxis $t \geq 0$ and should decay for t large enough. With this convention for the convolution, the weight given to past events corresponds to the value of the kernel for positive argument. The value of the kernel $\omega(t - t')$ is the weight of events in the past, at a time interval $t - t'$ from t. In this convolution, $z(t')$ is a continuous function of time. Actual time series z are known only at the sampling time t_i and should be interpolated between sampling points. As in Section 3.2.1, we can define different interpolation procedures for the value of z(t) between t_{j-1} and t_j. Three are used in practice:

- Previous value, $z(t) = z_{j-1}$
- Next value, $z(t) = z_j$
- Linear interpolation, $z(t) = z_{j-1} + (z_j - z_{j-1})(t - t_{j-1})/(t_j - t_{j-1})$

The linear interpolation seems preferable as it leads to a continuous interpolated function. Moreover, linear interpolation defines the mean path of a random walk, given the start and end values. Unfortunately, it is non-causal, because in the interval between t_{i-1} and t_i, the value at the end of the interval z_i is used. Only the previous-value interpolation is causal, as only the information known at t_{i-1} is used in the interval between t_{i-1} and t_i. Any interpolation can be used for historical computations, but for the real-time situation, only the causal previous-value interpolation is defined. In practice, the interpolation scheme is almost irrelevant for good macroscopic operators, (i.e., if the kernel has a range longer than the typical sampling rate).

The kernel $\omega(t)$ can be extended to $t \in \mathbb{R}$, with $\omega(t) = 0$ for $t < 0$. This is useful for analytical computation, particularly when the order of integral evaluations has to be changed. If the operator Ω is linear and time-translation invariant but noncausal, the same representation can be used except that the kernel may be nonzero on the whole time axis.

We often use two broad families of operators that share general shapes and properties:

- An average operator has a kernel which is nonnegative, $\omega(t) \geq 0$, and normalized to unity, $\int dt\, \omega(t) = 1$. This implies that Ω[Parameters; Const] =Const.

- Derivative and difference operators have kernels that measure the difference between a value now and a value in the past (with a typical lag of τ). Their kernels have a zero average, $\int dt\, \omega(t) = 0$, such that Ω[Parameters; Const] = 0.

The integral in Equation 3.29 can also be evaluated in scaled time. In this case, the kernel is no more invariant with respect to physical time translation (i.e., it depends on t and t'), but it is invariant with respect to translation in business time. If the operator is an average or a derivative, the normalization property is preserved in scaled time.

The n-th moment of a causal kernel ω is defined as

$$\langle t^n \rangle_\omega := \int_0^\infty dt\; \omega(t)\, t^n \tag{3.30}$$

The range R and the width w of an operator Ω are defined, respectively, by the following relations:

$$\begin{aligned} R[\Omega] &= \langle t \rangle_\omega = \int_0^\infty dt\; \omega(t)\, t \\ w^2[\Omega] &= \langle (t-R)^2 \rangle_\omega = \int_0^\infty dt\; \omega(t)\, (t-R)^2 \end{aligned} \tag{3.31}$$

For most operators $\Omega[\tau]$ depending on a time range τ, the formula is set up so that $|R[\Omega[\tau]]| = \tau$.

Linear operators can be applied *successively*:

$$\Omega_C[z] = \Omega_2 \circ \Omega_1[z] = \Omega_2\, \Omega_1\, z := \Omega_2[\Omega_1[z]]$$

It is easy to show that the kernel of Ω_C is given by the *convolution* of the kernels of Ω_1 and Ω_2.

$$\omega_C = \omega_1 \star \omega_2 \qquad \text{or} \tag{3.32}$$

$$\omega_C(t - t') = \int_{-\infty}^{\infty} dt''\, \omega_1(t - t'')\, \omega_2(t'' - t') \tag{3.33}$$

For causal operators,

$$\omega_C(t) = \int_{-t/2}^{t/2} dt'\, \omega_1(\frac{t}{2} - t')\, \omega_2(t' + \frac{t}{2}) \qquad \text{for } t \geq 0 \tag{3.34}$$

and $\omega_C(t) = 0$ for $t < 0$. Under convolution, range and width obey the following simple laws:

$$\begin{aligned} R_C &= R_1 + R_2 \\ w_C^2 &= w_1^2 + w_2^2 \\ \langle t^2 \rangle_C &= \langle t^2 \rangle_1 + \langle t^2 \rangle_2 + 2 r_1 r_2 \end{aligned} \tag{3.35}$$

3.3.3 Build-Up Time Interval

As our basic building blocks are EMA operators, most kernels have an exponential tail for large t. This implies that, when starting the evaluation of an operator at time T, a build-up time interval must be elapsed before the result of the evaluation is accurate enough (i.e., the influence of the initial error at T has sufficiently faded). This heuristic statement can be expressed by quantitative definitions. We assume that the process $z(t)$ is known since time $-T$ and is modeled before as an unknown random walk with no drift. Equation 3.29 for an operator Ω needs to be modified in the following way:

$$\Omega[-T; z](t) = \int_{-T}^{t} dt'\, \omega(t - t')\, z(t') . \tag{3.36}$$

The "infinite" build-up corresponds to $\Omega[-\infty; z](t)$. For $-T < 0$, the average build-up error ϵ at $t = 0$ is given by

$$\epsilon^2 = E[(\Omega[-T; z](0) - \Omega[-\infty; z](0))^2] = E\left[\left(\int_{-\infty}^{-T} dt'\, \omega(-t')\, z(t')\right)^2\right] \tag{3.37}$$

where E is the expectation operator. For a given build-up error ϵ, this equation is the implicit definition of the build-up time interval T. In order to compute the expectation, we need to specify the considered space of random processes. We assume simple random walks with constant volatility σ, namely

$$E[(z(t) - z(t + \delta t))^2] = \sigma \frac{\delta t}{1\text{y}} \tag{3.38}$$

The symbol $1y$ denotes one year, so $\delta t/1y$ is the length of δt expressed in years. With this choice of units, σ is an annualized volatility, with values roughly from 1% (for bonds) to 50% (for stocks), and a typical value of 10% for foreign exchange. For $t < -T$, $t' < -T$, we have the expectation

$$E[z(t)z(t')] = z(-T)^2 + \sigma \min(\frac{-t-T}{1y}, \frac{-t'-T}{1y}) \tag{3.39}$$

Having defined the space of processes, a short computation gives

$$\epsilon^2 = z(-T)^2 \left(\int_T^\infty dt\, \Omega(t) \right)^2 + 2\sigma \int_T^\infty dt\, \omega(t) \int_T^t dt'\, \omega(t') \frac{t'-T}{1y} \tag{3.40}$$

The first term is the "error at initialization" corresponding to the decay of the initial value $\Omega[-T](-T) = 0$ in Equation 3.36. A better initialization is $\Omega[-T](-T) = z(-T) \int_0^\infty \omega(t)$, corresponding to a modified definition for $\Omega[T](t)$:

$$\Omega[T; z](t) = z(-T) \int_{-\infty}^T dt'\, \omega(t-t') + \int_T^t dt'\, \omega(t-t')\, z(t') \tag{3.41}$$

Another interpretation for the above formula is that z is approximated by its most probable value $z(-T)$ for $t < T$. With this better definition for Ω, the error reduces to

$$\epsilon^2 = 2\sigma \int_T^\infty dt\, \omega(t) \int_T^t dt'\, \omega(t') \frac{t'-T}{1y} \tag{3.42}$$

For a given kernel ω, volatility σ and error ϵ, Equation 3.42 is an equation for T. Most of the kernels introduced in the next section have the scaling form $\omega(\tau, t) = \tilde{\omega}(t/\tau)/\tau$. In this case, the equation for $\tilde{T} = T/\tau$ reduces to

$$\epsilon^2 = 2\sigma \frac{\tau}{1y} \int_{\tilde{T}}^\infty dt\, \tilde{\omega}(t) \int_{\tilde{T}}^t dt'\, \tilde{\omega}(t')\, (t' - \tilde{T}) \tag{3.43}$$

Because this equation cannot be solved for general operators, the build-up interval should be computed numerically. This equation can be solved analytically for the simple EMA kernel, and it gives the solution for the build-up time

$$\frac{T}{\tau} = -\ln \epsilon + \frac{1}{2} \ln \left(\frac{\sigma}{2} \frac{\tau}{1y} \right) \tag{3.44}$$

As expected, the build-up time interval is large for a small error tolerance and for processes with high volatility. For operators more complicated than the simple EMA, Equation 3.43 is in general not solvable analytically. A simple rule of thumb can be given such that the heavier the tail of the kernel, the longer the required build-up. A simple measure for the tail can be constructed from the first two moments of the kernel as defined by Equation 3.30. The aspect ratio $AR[\Omega]$ is defined as

$$AR[\Omega] = \langle t^2 \rangle_\omega^{1/2} / \langle t \rangle_\omega$$

Both $\langle t \rangle$ and $\sqrt{\langle t^2 \rangle}$ measure the extension of the kernel and are usually proportional to τ; thus the aspect ratio is independent of τ and dependent only on the shape of the kernel, in particular its tail property. Typical values of this aspect ratio are $2/\sqrt{3}$ for a rectangular kernel and $\sqrt{2}$ for a simple EMA. A low aspect ratio means that the kernel of the operator has a short tail and therefore a short build-up time interval in terms of τ. This is a good rule for nonnegative causal kernels; the aspect ratio is less useful for choosing the build-up interval of causal kernels with more complicated, partially negative shapes.

3.3.4 Homogeneous Operators and Robustness

There are many ways to build nonlinear operators; an example is given in Section 3.3.13 for the (moving) correlation. In practice, most nonlinear operators are homogeneous of degree p, namely $\Omega[ax] = |a|^p \, \Omega[x]$ (here the word "homogeneous" is used in a sense different from that in the term "homogeneous time series"). Translation-invariant homogeneous operators of degree pq take the simple form of a convolution

$$\Omega[z](t) = \left[\int_{-\infty}^{t} dt' \, \omega(t - t') \, |z(t')|^p \right]^q \tag{3.45}$$

for some exponents p and q. An example is the moving norm (see Section 3.3.8) with ω corresponding to an average and $q = 1/p$.

Nonlinear operators can also be used to build robust estimators. Data errors (outliers) should be eliminated by a data filter prior to any computation, as discussed in Chapter 4. As an alternative or in addition to prior data cleaning, robust estimators can reduce the dependency of results on outliers or the choice of the data cleaning algorithm. This problem is acute mainly when working with returns, because the difference operator needed to compute returns (r) from prices (x) is sensitive to outliers. The following modified operator achieves robustness by giving a higher weight to the center of the distribution of returns r than to the tails:

$$\Omega[f; r] = f^{-1} \{ \Omega[f(r)] \} \tag{3.46}$$

where f is an odd, monotonic function over $\mathbb{R}$. Possible mapping functions $f(x)$ are

$$\mathrm{sign}(x)|x|^\gamma \;=\; x\,|x|^{\gamma-1} \tag{3.47}$$

$$\mathrm{sign}(x) \quad \text{when} \quad \gamma \to 0 \tag{3.48}$$

$$\tanh(x/x_0) \tag{3.49}$$

Robust operator mapping functions defined by Equation 3.47 have an exponent $0 \leq \gamma < 1$. In some special applications, operators with $\gamma > 1$, emphasizing the tail of the distribution, may also be used. In the context of volatility estimates, the usual L^2 volatility operator based on squared returns can be made more robust by using the mapping function $f = \text{sign}(x)\sqrt{|x|}$ (the signed square root); the resulting volatility is then based on absolute returns as in Equation 3.67. More generally, the signed power $f(x) = \text{sign}(x)|x|^p$ transforms an L^2 volatility into an L^{2p} volatility. This simple power law transformation is often used and therefore included in the definition of the moving norm, moving variance or volatility operators, Equation 3.60. Yet some more general transformations can also be used.

3.3.5 Exponential Moving Average (EMA)

The basic exponential moving average (EMA) is the simplest linear operator, the first one in a series of linear operators to be presented. It is an averaging operator with an exponentially decaying kernel:

$$\text{ema}(t) = \frac{e^{-t/\tau}}{\tau} \tag{3.50}$$

This EMA operator is our foundation stone, because its computation is very efficient and other more complex operators can be built with it, such as moving averages (MAs), differentials, derivatives, and volatilities. The numerical evaluation is efficient because of the exponential form of the kernel, which leads to a simple iterative formula first proposed by Müller (1991):

$$\begin{aligned} \text{EMA}[\tau; z](t_n) &= \\ &\mu\ \text{EMA}[\tau; z](t_{n-1}) + (\nu - \mu)\, z_{n-1} + (1 - \nu)\, z_n \end{aligned} \tag{3.51}$$

with

$$\alpha = \frac{t_n - t_{n-1}}{\tau}$$

$$\mu = e^{-\alpha}$$

where ν depends on the chosen interpolation scheme,

$$\nu = \begin{cases} 1 & \text{previous point} \\ (1-\mu)/\alpha & \text{linear interpolation} \\ \mu & \text{next point} \end{cases} \tag{3.52}$$

Due to this iterative formula, the convolution is never computed in practice; only few multiplications and additions have to be done for each tick. In Section 3.3.14, the EMA operator is extended to the case of complex kernels.

3.3.6 The Iterated EMA Operator

The basic EMA operator can be iterated to provide a family of iterated exponential moving average operators EMA$[\tau, n]$. Practitioners of technical analysis have applied simple and (occasionally) iterated EMA operators to homogeneous time series for a long time. Iterated EMA operators for inhomogeneous time series were first explored by Müller (1991) and systematically developed and discussed by Zumbach and Müller (2001). A simple recursive definition is

$$\mathrm{EMA}[\tau, n; z] = \mathrm{EMA}[\tau; \mathrm{EMA}[\tau, n-1; z]] \tag{3.53}$$

with $\mathrm{EMA}[\tau, 1; z] = \mathrm{EMA}[\tau; z]$. This definition can be efficiently evaluated by using the iterative formula in Equation 3.51 for all its basic EMAs. There is one subtle point related to the choice of the interpolation scheme in Equation 3.52. The EMA of z necessarily has an interpolation scheme different from that used for z. The correct form of $\mathrm{EMA}[\tau; z]$ between two points is no longer a straight line but a nonlinear (exponential) curve. (Theoretically, it is straightforward to derive the corresponding exact interpolation formula.) When using one of the interpolation schemes of Equation 3.52 after the first iteration, we are making a small error. Yet if the kernel is wide as compared to $t_n - t_{n-1}$, this error is indeed very small. As a suitable *approximation*, we recommend using linear interpolation in the second and all further EMA iterations, even if the first iteration was based on the next-point interpolation. The only exception occurs if z_n is not yet known; then we need a causal operator based on the previous-point interpolation.

The kernel of $\mathrm{EMA}[\tau, n]$ is

$$\mathrm{ema}[\tau, n](t) = \frac{1}{(n-1)!} \left(\frac{t}{\tau}\right)^{n-1} \frac{e^{-t/\tau}}{\tau} \tag{3.54}$$

This family of functions is related to Laguerre polynomials, which are orthogonal with respect to the measure e^{-t} (for $\tau = 1$). Through an expansion in Laguerre polynomials, any kernel can be expressed as a sum of iterated EMA kernels. Therefore, the convolution with an arbitrary kernel can be evaluated by iterated exponential moving averages. Yet the convergence of this expansion may be slow, namely high-order iterated EMAs may be necessary, possibly with very large coefficients. This typically happens if one tries to construct operators that have a decay other (faster) than exponential. Therefore, in practice, we construct operators empirically from a few low-order EMAs, in a way to minimize the build-up time. The set of operators provided by Section 3.3 covers a wide range of computations needed in finance. The range, width, and aspect ratio of the iterated EMA are

$$\begin{aligned} R &= n\tau \\ \langle t^2 \rangle &= n(n+1)\tau^2 \\ w^2 &= n\tau^2 \\ AR &= \sqrt{(n+1)/n} \end{aligned} \tag{3.55}$$

The iterated $\mathrm{EMA}[\tau, n]$ operators with large n have a shorter, more compact kernel and require a shorter build-up time interval than a simple EMA of the same range

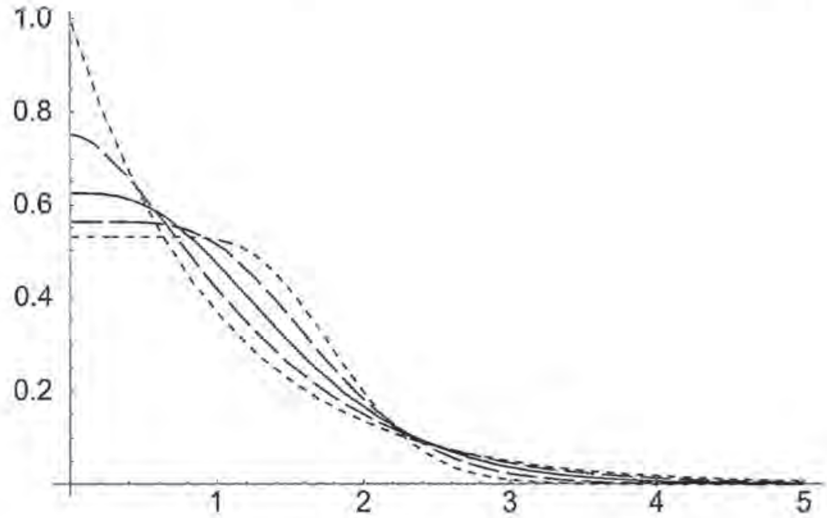


FIGURE 3.5 ma$[\tau, n](t)$ for $n = 1, 2, 4, 8$, and 16, for $\tau = 1$

$n\tau$. This is indicated by the aspect ratio AR, which decreases toward 1 for large n. Each basic EMA operator that is part of the iterated EMA has a range τ, which is much shorter than the range $n\tau$ of the full kernel. Even if the tail of the kernel is still exponential, it decays more quickly due to the small basic EMA range τ.

To further extend our computational toolbox, we build another type of compact kernel by combining iterated EMAs, as shown in the next section. As the iterated EMAs, these combined iterated EMAs have a shorter build-up time interval than a simple EMA of the same range.

3.3.7 Moving Average (MA)

A very convenient moving average is provided by

$$\text{MA}[\tau, n] = \frac{1}{n}\sum_{k=1}^{n} \text{EMA}[\tau', k] \qquad \text{with } \tau' = \frac{2\tau}{n+1} \tag{3.56}$$

The parameter τ' is chosen so that the range of MA$[\tau, n]$ is $R = \tau$, independently of n. This provides a family of more rectangular-shaped kernels, with the relative weight of the distant past controlled by n. Kernels for different values of n and $\tau = 1$ are shown in Figure 3.5. Their analytical form is given by

$$\text{ma}[\tau, n](t) = \frac{n+1}{n}\frac{e^{-t/\tau'}}{2\tau}\sum_{k=0}^{n-1}\frac{1}{k!}\left(\frac{t}{\tau'}\right)^k \tag{3.57}$$

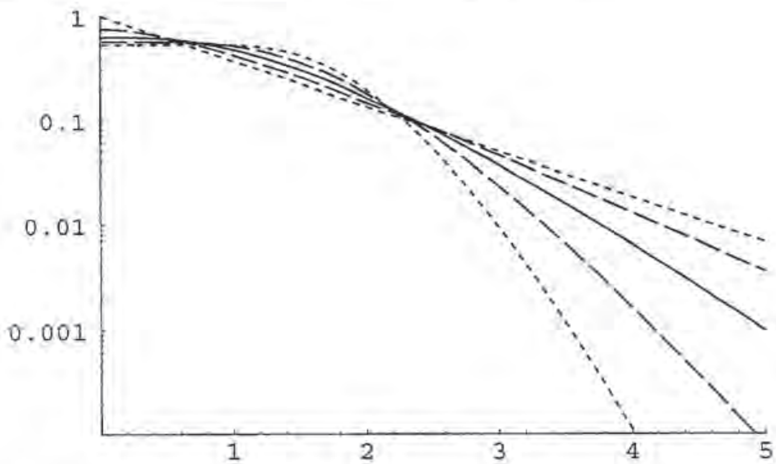


FIGURE 3.6 ma$[\tau, n](t)$ for $n = 1, 2, 4, 8$, and 16, for $\tau = 1$, on a logarithmic scale.

For $n = \infty$, the sum corresponds to the Taylor expansion of $\exp(t/\tau')$, which cancels the term $\exp(-t/\tau')$, making the kernel constant. For finite n, when t/τ' is small enough, the finite sum will be a good approximation of $\exp(t/\tau')$. Small enough means that the largest term in the sum is of order one: $(t/\tau')^n/n! \sim 1$. For large n, the condition $(t/\tau')^n/n! \sim 1$ corresponds to $t \sim 2\tau$ (using Stirling's approximation $n! \sim n^n$). Therefore, for $t \ll 2\tau$,the series approximates well the Taylor expansion of an exponential

$$\sum_{k=0}^{n-1} \frac{1}{k!}\left(\frac{t}{\tau'}\right)^k \to e^{t/\tau'}$$

$$\text{ma} \to \frac{n+1}{n}\,\frac{1}{2\tau}$$

This explains the constant behavior of the kernel for $t < 2\tau$. For $t > 2\tau$ large, the exponential always dominates and the kernel decays to zero. Therefore, for large n, this operator tends to a rectangular moving average for which $AR = 2/\sqrt{3}$. For n values of $n \sim 5$ and higher, the kernel is rectangular-like more than EMA-like; this can be seen in Figure 3.5. These rectangular-like kernels are preferred to the rectangular kernel itself because they fade smoothly rather than abruptly. Abrupt "forgetting" of past events leads to superfluous noise in the operator results.

The decay of MA kernels is also shown in Figure 3.6. The aspect ratio of the MA operator is

$$AR = \sqrt{\frac{4(n+2)}{3(n+1)}} \tag{3.58}$$

Clearly, the larger n, the shorter the build-up.

This family of operators can be extended by "peeling off" some EMAs with small k:

$$\mathrm{MA}[\tau, n_{\mathrm{inf}}, n_{\mathrm{sup}}] = \frac{1}{n_{\mathrm{sup}} - n_{\mathrm{inf}} + 1} \sum_{k=n_{\mathrm{inf}}}^{n_{\mathrm{sup}}} \mathrm{EMA}[\tau', k]$$

with

$$\tau' = \frac{2\tau}{n_{\mathrm{sup}} + n_{\mathrm{inf}}}$$

and with $1 \leq n_{\mathrm{inf}} \leq n_{\mathrm{sup}}$. By choosing such a modified MA with $n_{\mathrm{inf}} > 1$, we can generate a lagged operator with a kernel whose rectangular-like form starts after a lag rather than immediately. At the same time, the kernel loses its abrupt behavior at $t = 0$ and becomes fully continuous, thus reducing noise in the results even further. However, the time delay implied by the lag makes such kernels less attractive for real-time applications.

Almost everywhere, a moving average operator can be used instead of a sample average. The sample average of $z(t)$ is defined by

$$\mathrm{E}[z] = \frac{1}{t_e - t_s} \int_{t_s}^{t_e} dt'\, z(t') \tag{3.59}$$

where the dependency on start-time t_s and end-time t_e is implicit on the left-hand side. This dependency can be made explicit, for example with the notation $\mathrm{E}[t_e - t_s; z](t_e)$, thus demonstrating the parallelism between the sample average and a moving average $\mathrm{MA}[2\tau; z](t)$. The conceptual difference is that when using a sample average, t_s and t_e are fixed, and the sample average is a number (the sample average is a functional from the space of time series to $\mathbb{R}$), whereas the MA operator produces another time series. Keeping this difference in mind, we can replace the sample average $\mathrm{E}[\cdot]$ by a moving average $\mathrm{MA}[\cdot]$. For example, we can construct a standardized time series $\hat{z}$ (as defined in Section 3.3.1), a moving skewness, or a moving correlation (see the various definitions below). Yet be aware that sample averages and MAs can behave differently, for example $\mathrm{E}\left[(z - \mathrm{E}[z])^2\right] = \mathrm{E}\left[z^2\right] - \mathrm{E}[z]^2$, whereas $\mathrm{MA}[(z - \mathrm{MA}[z])^2] \neq \mathrm{MA}[z^2] - \mathrm{MA}[z]^2$.

3.3.8 Moving Norm, Variance, and Standard Deviation

With the efficient moving average operator, we can define the moving norm, moving variance, and moving standard deviation operators:

$$\begin{aligned} \mathrm{MNorm}[\tau, p; z] &= \mathrm{MA}[\tau; |z|^p]^{1/p} \\ \mathrm{MVar}[\tau, p; z] &= \mathrm{MA}[\tau; |z - \mathrm{MA}[\tau; z]|^p] \\ \mathrm{MSD}[\tau, p; z] &= \mathrm{MA}[\tau; |z - \mathrm{MA}[\tau; z]|^p]^{1/p} \end{aligned} \tag{3.60}$$

The norm and standard deviation are homogeneous of degree 1 with respect to z. The p-moment is related to the norm by $\mu_p = \mathrm{MA}[|z|^p] = \mathrm{MNorm}[z]^p$. Usually, $p = 2$ is taken. Lower values for p provide a more robust estimate (see

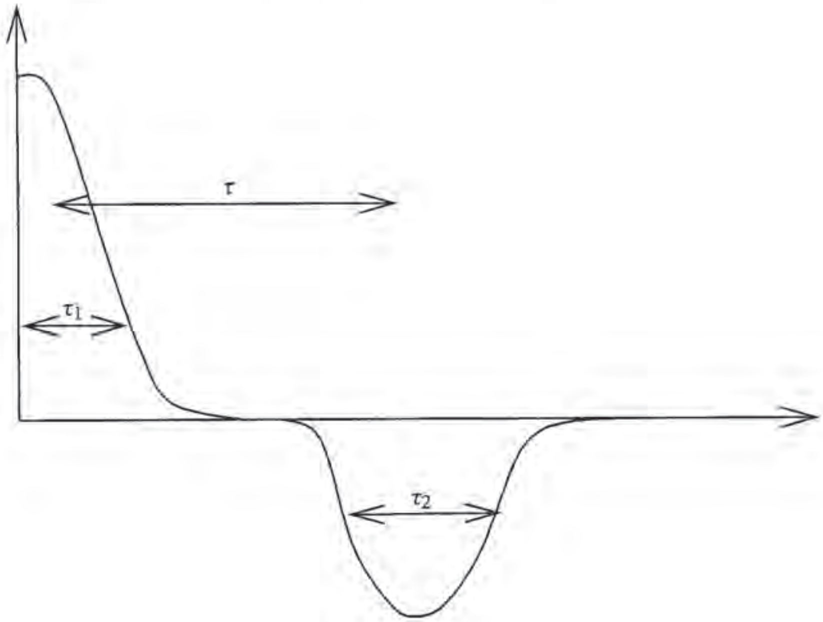


FIGURE 3.7 A schematic differential kernel.

Section 3.3.4), and $p = 1$ is another common choice. Yet even lower values can be used, for example, $p = 1/2$. In the formulae for MVar and MSD, there are two MA operators with the same range τ and the same kernel. This choice is in line with common practice for the calculation of empirical means and variances in the same sample. Yet other choices can be interesting, for example the sample mean can be estimated with a longer time range.

3.3.9 Differential

As argued in the introduction, a low-noise differential operator suitable to stochastic processes should compute an "average differential", namely the difference between an average around time "now" over a time interval τ_1 and an average around time "now $- \tau$" on a time interval τ_2. The kernel may look like that in Figure 3.7. Kernels of a similar kind are used for wavelet transforms. This analogy also applies to other kernel forms and is further discussed in Section 3.3.14.

Usually, τ, τ_1 and τ_2 are related and only the τ parameter appears, with $\tau_1 \sim \tau_2 \sim \tau/2$. The normalization of the differential Δ is chosen so that $\Delta[\tau; c] = 0$ for a constant function $c = c(t) =$ constant, and $\Delta[\tau; t] = \tau$. Note that our point of view is different from that used in continuous-time stochastic analysis. In continuous time, the limit $\tau \to 0$ is taken, leading to the Ito derivative with its subtleties. In our case, we keep the range τ finite in order to be able to analyze

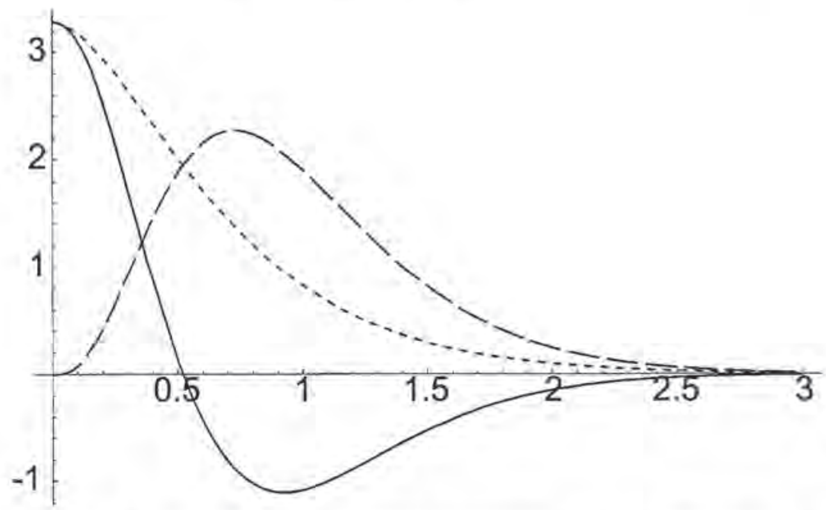


FIGURE 3.8 An example of a differential operator kernel (full line) for $\tau = 1$. The dotted curve corresponds to the first two terms of the operator $\gamma(\text{EMA}[\alpha\tau, 1] + \text{EMA}[\alpha\tau, 2])$, the dashed curve to the last term $2\gamma\ \text{EMA}[\alpha\beta\tau, 4]$.

a process at different time scales (i.e., for different orders of magnitudes of τ). Moreover, for financial data, the limit $\tau \to 0$ cannot be taken because a process is known only on a discrete set of time points (and probably does not exist in continuous time).

The following operator can be selected as a suitable differential operator:

$$\Delta[\tau] = \gamma\ (\text{EMA}[\alpha\tau, 1] + \text{EMA}[\alpha\tau, 2] - 2\ \text{EMA}[\alpha\beta\tau, 4]) \tag{3.61}$$

with $\gamma = 1.22208$, $\beta = 0.65$ and $\alpha^{-1} = \gamma(8\beta - 3)$. This operator has a well-behaving kernel that is plotted in Figure 3.8.

The value of γ is fixed so that the integral of the kernel from the origin to the first zero is one. The value of α is fixed by the normalization condition and the value of β is chosen in order to get a short tail. The tail can be seen in Figure 3.9. This shows that after $t = 3.25\tau$, the kernel is smaller than 10^{-3}, which translates into a small required build-up time of about 4τ.

In finance, the main purpose of a Δ operator is computing returns of a time series of (logarithmic) prices x with a given time interval τ. Returns are normally defined as changes of x over τ; we prefer the alternative return definition $r[\tau] = \Delta[\tau; x]$. This computation requires the evaluation of six EMAs and is therefore efficient, time-wise and memory-wise. An example using our standard week is plotted in Figure 3.10, demonstrating the low noise level of the differential. The conventionally computed return $r[\tau](t) = x(t) - x(t - \tau)$ is very inefficient to evaluate for inhomogeneous time series. The computation of $x(t - \tau)$ requires a high, unbounded number of old t_i, x_i values to be kept in memory, and the t_i

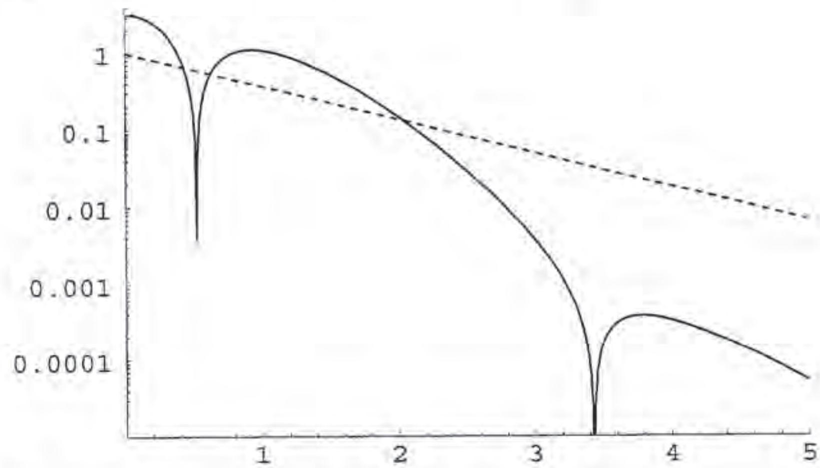


FIGURE 3.9 The absolute value of the kernel of the differential operator (full line), in a logarithmic scale. The dotted line shows a simple EMA with range τ, demonstrating the much faster decay of the differential kernel.

interval bracketing the time $t - \tau$ has to be searched for. This return definition corresponds to a differential operator kernel made of two δ functions (or to the limit $\tau_1, \tau_2 \to 0$ of the kernel in Figure 3.7). The quantity $x(t) - x(t - \tau)$ can be quite noisy, so a further EMA might be taken to smooth it. In this case, the resulting effective differential operator kernel has two discontinuities, at 0 and at τ, and decays exponentially—that is, much slower than the kernel of $\Delta[\tau; x]$. Thus it is cleaner and more efficient to compute returns with the Δ operator of Equation 3.61. Another quantity commonly used in finance is $x - \text{EMA}[\tau; x]$, often called a momentum or an oscillator. This is also a differential with the kernel $\delta(t) - \exp(-t/\tau)/\tau$, with a δ function at $t = 0$. A simple drawing shows that the kernel of Equation 3.61 produces a much less noisy differential. Other appropriate kernels can be designed, depending on the application. In general, there is a trade-off between the averaging property of the kernel and a short response to shocks of the original time series.

3.3.10 Derivative and γ-Derivative

The derivative operator

$$D[\tau] = \frac{\Delta[\tau]}{\tau} \tag{3.62}$$

behaves exactly as the differential operator, except for the normalization $D[\tau; t] = 1$. This derivative can be iterated in order to construct higher order derivatives:

$$D^2[\tau] = D[\tau; D[\tau]] \tag{3.63}$$

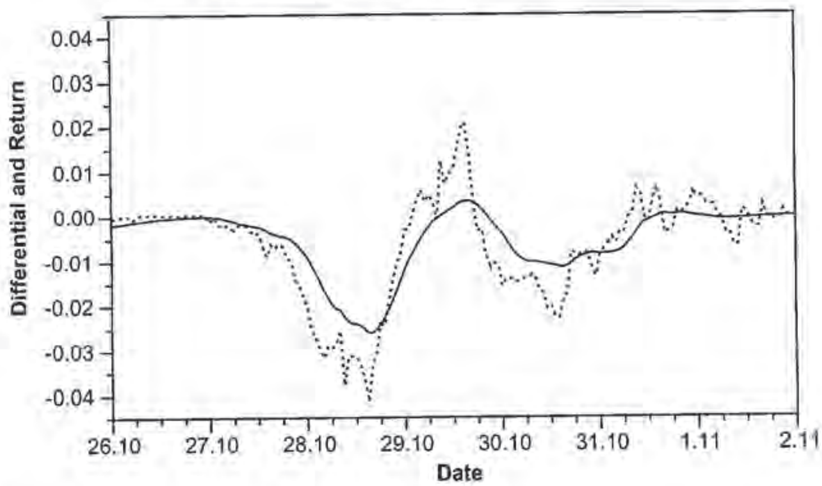


FIGURE 3.10 A comparison between the differential computed using the formula in Equation 3.61 with τ = 24hr (full line) and the pointwise return $x(t) - x(t - 24\text{h})$ (dotted line). The time lag of approximately 4hr between the curves is essentially due to the extent of both the positive part of the kernel ($0 < t < 0.5$) and the tail of the negative part ($t > 1.5$).

The range of the second-order derivative operator is $2\,\tau$. More generally, the n-th order derivative operator D^n, constructed by iterating the derivative operator n times, has a range $n\tau$. As defined, the derivative operator has the dimension of an inverse time. It is easier to work with dimensionless operators and this is done by measuring τ in some units. One year provides a convenient unit, corresponding to an annualized return when $D[\tau]x$ is computed. The choice of unit is denoted by $\tau/1y$, meaning that τ is measured in years, yet other units may also be used. For a random diffusion process, a more meaningful normalization for the derivative is to take $D[\tau] = \Delta[\tau]/\sqrt{\tau/1y}$. For a space of processes as in Section 3.3.3, such that Equation 3.38 holds, the basic scaling behavior with τ is eliminated, namely $\mathrm{E}\left[(D[\tau]z)^2\right] = \sigma^2$. More generally, we can define a γ-derivative as

$$D[\tau, \gamma] = \frac{\Delta[\tau]}{(\tau/1y)^\gamma} \tag{3.64}$$

In particular

$$\begin{aligned} \gamma &= 0 && \text{differential} \\ \gamma &= 0.5 && \text{stochastic diffusion process} \\ \gamma &= 1 && \text{the usual derivative} \end{aligned} \tag{3.65}$$

An empirical probability density function for the derivative is displayed in Figure 3.11. We clearly see that the main part of the scaling with τ is removed when using the γ-derivative with $\gamma = 0.5$.

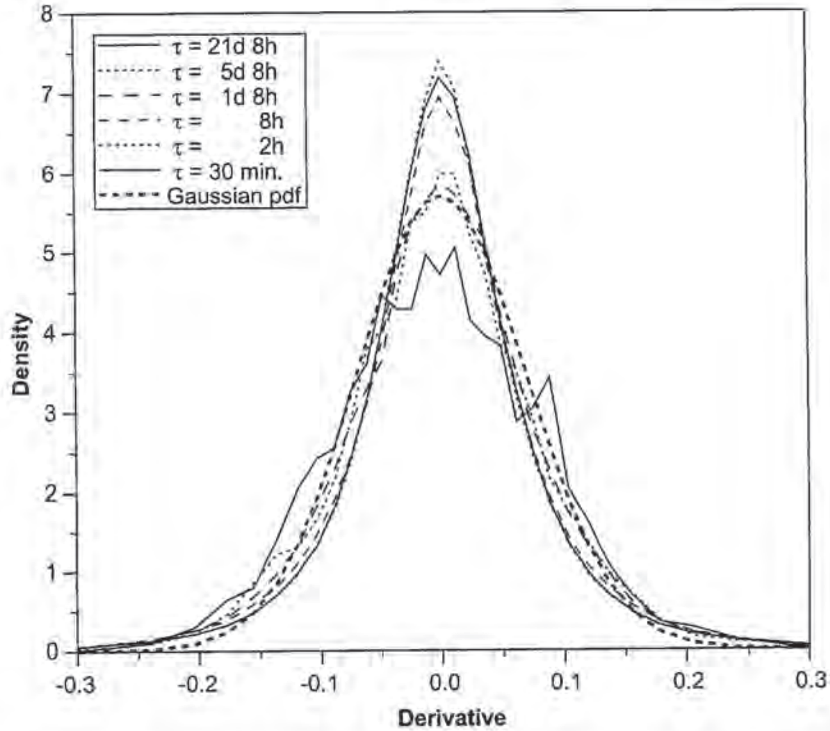


FIGURE 3.11 The annualized derivative $D[\tau, \gamma = 0.5; x]$ for USD-CHF from January 1, 1988 to November 1, 1998. The shortest time intervals τ correspond to the most leptokurtic curves. In order to discard the daily and weekly seasonality, the time scale used is the business time ϑ as explained in Chapter 6 and in Dacorogna *et al.* (1993). The data were sampled every 2 hr (in ϑ-time) to construct the curves. The Gaussian probability density function added for comparison has a standard deviation of $\sigma = 0.07$, similar to that of the other curves.

3.3.11 Volatility

The most common computation of realized or historical volatility is given by Equation 3.8 in Section 3.2.4, based on regularly spaced (e.g., daily) observations. Realized volatility can also be defined and measured with the help of convolution operators.

Volatility is a measure widely used for random processes, quantifying the size and intensity of movements, namely the current "width" of the probability distribution $P(\Delta z)$ of the process increment Δz, where Δ is a difference operator yet to be chosen. Often the volatility of market prices is computed, but volatility is a general operator that can be applied to any time series. There are many ways to turn this idea into a definition, and there is no unique, universally accepted definition of volatility in finance. In our new context, we can reformulate the realized volatility of Equation 3.8 as an L^2 norm,

$$\text{Volatility}[\tau, \tau'; z] = \left(\frac{1}{n} \sum_{i=0}^{n-1} (\delta\ \text{RTS}[\tau'; z])_i^2 \right)^{1/2} \quad \text{with} \quad \tau = n\,\tau' \qquad (3.66)$$

where the operator δ computes the difference between successive values (see Section 3.4.3), τ' is the return interval, and τ is the length of the moving sample. RTS[τ'; z] is an artificial regular time series, spaced by τ', constructed from the irregular time series z. The construction of homogeneous time series was discussed in Sections 3.2.1 and 3.2.2; it is reformulated in Section 3.4.2 in terms of the RTS operator. Realized volatility based on artificially regularized data suffers from several drawbacks:

- For inhomogeneous time series, a synthetic regular time series must be created, which involves an interpolation scheme.
- The difference is computed with a pointwise difference. This implies some noise in the case of stochastic data.
- Only some values at regular time points are used. Information from other points of the series, between the regular sampling points, is thrown away. Resulting from this information loss, the estimator is less accurate than it could be.
- It is based on a rectangular weighting kernel—that is, all points have constant weights of either $1/n$ or 0 as soon as they are excluded from the sample. A continuous kernel with declining weights leads to a better, less disruptive, and less noisy behavior.
- By squaring the returns, this definition puts a large weight on large changes of z and therefore increases the impact of outliers and the tails of $P(z)$. Also, as the fourth moment of the probability distribution of the returns might not be finite (Müller *et al.*, 1998), the volatility of the volatility might not be finite either. In other words, this estimator is not very robust. These are reasons to prefer a realized volatility defined as an L^1 norm:

$$\text{Volatility}[\tau, \tau'; z] = \frac{1}{N} \sum_{i=0}^{N-1} |\Delta[\text{RTS}[\tau'; z]]_i| \quad \text{with} \quad \tau = N\,\tau' \qquad (3.67)$$

There are various ways to introduce better definitions for inhomogeneous time series. These definitions are variations of the following one:

$$\text{Volatility}[\tau, \tau', p; z] = \text{MNorm}[\tau/2, p; \Delta[\tau'; z]] \qquad (3.68)$$

where the moving norm MNorm is defined by Equation 3.60. For Δ, we can take the differential operator of Equation 3.61 or a similar operator. Let us emphasize that no homogeneous time series is needed, and that this definition can be computed simply and efficiently for high-frequency data because it ultimately involves only EMAs. Note the division by 2 in the MNorm of range $\tau/2$. This is to attain an equivalent of Equation 3.66, which is parametrized by the total sample size rather than the range of the (rectangular) kernel.

The volatility defined by Equation 3.68 is still a *realized* volatility although it is now based on inhomogeneous data and operators. The kernel form of the differential operator Δ has a certain influence on the size of the resulting volatility. A "soft" kernel will lead to a lower mean value of volatility than a "hard" kernel whose positive and negative parts are close to delta functions. This has to be accounted for when applying operator-based volatility.

The variations of Equation 3.68 mainly include the following:

- Replacing the norm MNorm by a moving standard deviation MSD as defined by Equation 3.60. By this modification, the empirical sample mean is subtracted from all observations of $\Delta[\tau'; z]$. This leads to a formula analogous to Equation 3.11, whereas Equation 3.68 is analogous to Equation 3.8. Empirically, for most data in finance such as FX, the numerical difference between taking MNorm and MSD is very small.

- Replacing the differential Δ by a γ-derivative $D[\tau, \gamma]$. The advantage of using the gamma derivative is to remove the leading τ dependence, for example by directly computing the annualized volatility, independent of τ. An example is given by Figure 3.12.

Let us emphasize that the realized volatility in Equations 3.66 through 3.68 depends on the two time ranges τ and τ' and, to be unambiguous, both time intervals must be given. Yet, for example, when talking about a daily volatility, the common language is rather ambiguous because only one time interval is specified. Usually, the emphasis is put on τ'. A daily volatility, for example, measures the average size of daily price changes (i.e., $\tau' = 1$ day). The averaging time range τ is chosen as a multiple of τ', of the order $\tau \geq \tau'$ up to $\tau = 1000\tau'$ or more. Larger multiples lead to lower stochastic errors as they average over larger samples, but they are less local and dampen the time variations in the frequent case of nonconstant volatility. In empirical studies, we find that good compromises are in the range from $\tau = 16\tau'$ to $\tau = 32\tau'$.

On other occasions, for example in risk management, one is interested in the conditional daily volatility. Given the prices up to today, we want to produce an estimate or forecast for the size of the price move from today to tomorrow (i.e., the volatility within a small sample of only one day). The actual value of this volatility can be measured one day later; it has $\tau = 1$ day by definition. To measure this value with acceptable precision, we may choose a distinctly smaller τ', perhaps $\tau' = 1$ hr. Clearly, when only one time parameter is given, there is no simple convention to remove the ambiguity.

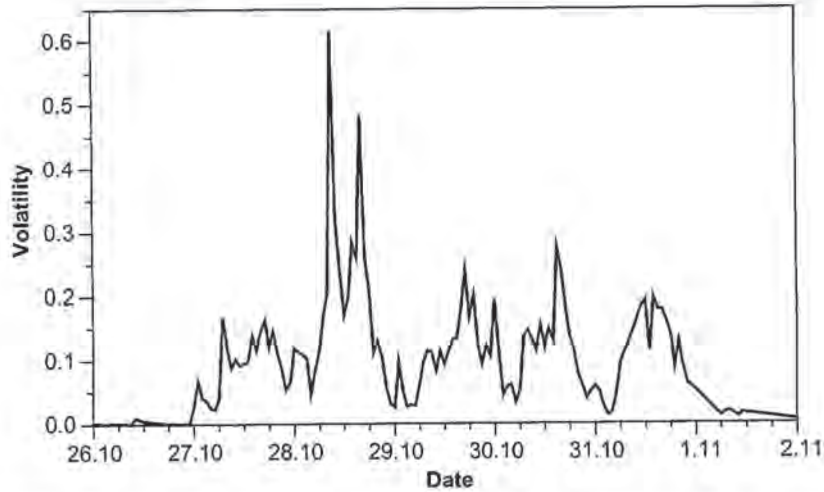


FIGURE 3.12 The annualized volatility computed as $\mathrm{MNorm}[\tau/2; D[\tau/32, \gamma = 0.5; x]]$ with τ = 1hr. The norm is computed with $p = 2$ and $n = 8$. The plotted volatility has five main maxima corresponding to the five working days of the example week. The Tuesday maximum is higher than the others, due to the stock market crash mentioned in the introductory part of Section 3.3.

3.3.12 Standardized Time Series, Moving Skewness, and Kurtosis

From a time series z, we can derive a moving standardized time series:

$$\hat{z}[\tau] = \frac{z - \mathrm{MA}[\tau; z]}{\mathrm{MSD}[\tau; z]} \tag{3.69}$$

In finance, z stands for the price or alternatively for another variable such as the return. Having defined a standardized time series $\hat{z}[\tau]$, the definitions for the moving skewness, and moving kurtosis are straightforward:

$$\begin{aligned} \mathrm{MSkewness}[\tau_1, \tau_2; z] &= \mathrm{MA}[\tau_1; \hat{z}[\tau_2]^3] \\ \mathrm{MKurtosis}[\tau_1, \tau_2; z] &= \mathrm{MA}[\tau_1; \hat{z}[\tau_2]^4] \end{aligned} \tag{3.70}$$

Instead of this kurtosis, the excess kurtosis is often used, whose value for a normal distribution is 0. We obtain the excess kurtosis by subtracting 3 from the MKurtosis value. The three quantities for our sample week are displayed in Figure 3.13.

3.3.13 Moving Correlation

Several definitions of a moving correlation can be constructed for inhomogeneous time series. Generalizing from the statistics textbook definition, we can write two simple definitions:

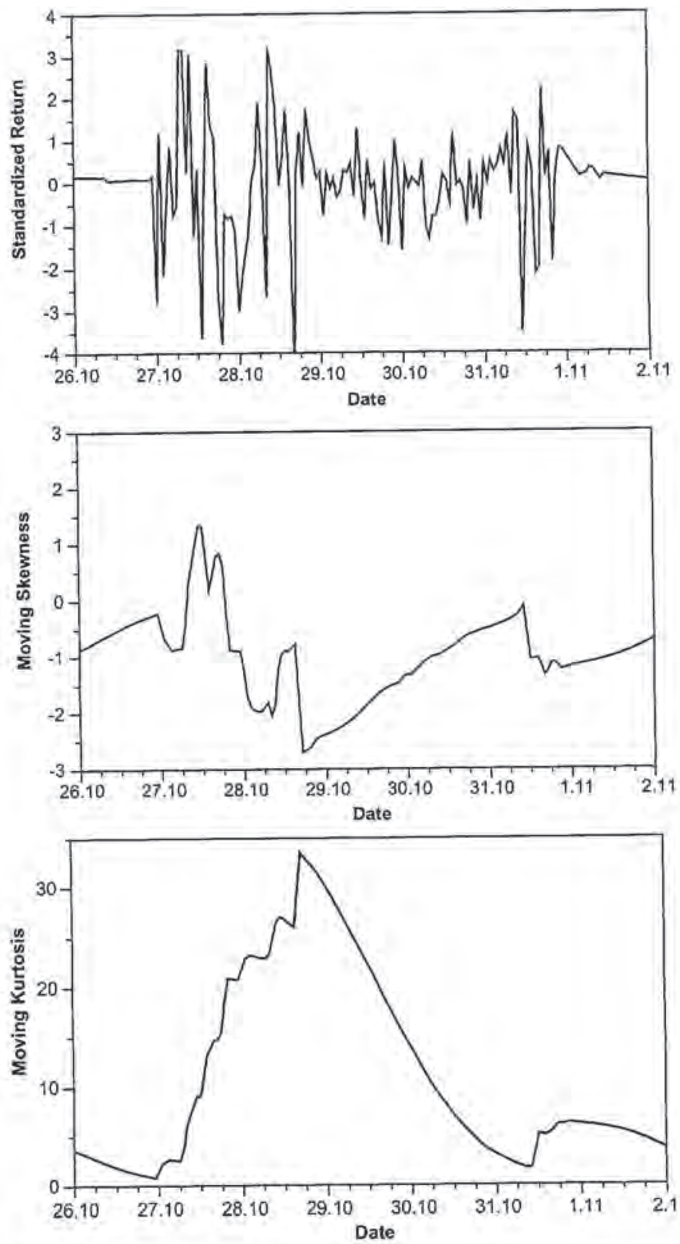


FIGURE 3.13 The standardized return, moving skewness, and moving kurtosis. The returns are computed as $r = D[\tau = 15 \text{ min}; x]$ and standardized with $\tau_1 = \tau_2 = 24\text{hr}$.

$$\text{MCorrelation}_1[\tau; y, z] = \frac{\text{MA}[\,(y - \text{MA}[y])(z - \text{MA}[z])\,]}{\text{MSD}[y]\ \text{MSD}[z]} \tag{3.71}$$

$$\begin{aligned}\text{MCorrelation}_2[\tau; y, z] &= \text{MA}\left[\frac{(y - \text{MA}[y])\ (z - \text{MA}[z])}{\text{MSD}[y]\ \text{MSD}[z]}\right] \\ &= \text{MA}[\,\hat{y}\,\hat{z}\,]\end{aligned} \tag{3.72}$$

where all of the MA and MSD[9] operators on the right-hand sides are taken with the same decay constant τ. These definitions are not equivalent because the MSD operators in the denominator are time series that do not commute with the MA operators. Yet both definitions have their respective advantages. The first definition obeys the inequality $-1 \leq \text{MCorrelation}_1 \leq 1$. This can be proven by noting that $MA[z^2](t)$ for a given t provides a norm on the space of (finite) time series up to t. It happens because the MA operator has a strictly positive kernel that acts as a metric on the space of time series. In this space, the triangle inequality holds $\sqrt{MA[(y+z)^2]} \leq \sqrt{MA[y^2]} + \sqrt{MA[z^2]}$, and, by a standard argument, the inequality on the correlation follows. With the second definition (Equation 3.72), the correlation matrix is bilinear for the standardized time series. Therefore, the rotation that diagonalizes the correlation matrix acts linearly in the space of standardized time series. This property is necessary for multivariate analysis, when a principal component decomposition is used. In risk management, the correlation of two time series of returns, x and y, is usually computed without subtracting the sample means of x and y. This implies a variation of Equations 3.71 and 3.72

$$\text{MCorrelation}'_1[\tau; y, z] = \frac{\text{MA}[\,y\,z\,]}{\text{MNorm}[y]\ \text{MNorm}[z]} \tag{3.73}$$

$$\text{MCorrelation}'_2[\tau; y, z] = \text{MA}\left[\frac{y\,z}{\text{MNorm}[y]\ \text{MNorm}[z]}\right] \tag{3.74}$$

where again the same τ is chosen for all MA operators. In general, any reasonable definition of a moving correlation must obey

$$\lim_{\tau\to\infty} \text{MCorrelation}[\tau; y, z] \to \rho[y, z] \tag{3.75}$$

where $\rho[y, z]$ is the theoretical correlation of the two stationary processes x and y. Generalizing the definition (Equation 3.72), the requirements for the correlation kernel are to construct a causal, time translation invariant, and a linear operator for $\hat{y}$ and $\hat{z}$. This leads to the most general representation

$$\text{MCorrelation}[\hat{y}, \hat{z}](t) = \int_0^\infty \int_0^\infty dt'\, dt''\, c(t', t'')\, \hat{y}(t - t')\, \hat{z}(t - t'') \tag{3.76}$$

[9] See Equation 3.60.

We also require symmetry between the arguments where MCorrelation$[\hat{z}, \hat{y}]$ = MCorrelation$[\hat{y}, \hat{z}]$. Moreover, the correlation must be a generalized average, namely MCorrelation[Const, Const'] = Const Const', or, formulated for the kernel, $\int \int_0^\infty dt'\, dt''\, c(t', t'') = 1$. There is a large choice of possible kernels that obey these requirements. For example, Equation 3.72 is equivalent to the kernel $c(t', t'') = \delta(t' - t'')\, \mathrm{ma}(\frac{t'+t''}{2})$.

3.3.14 Windowed Fourier Transform

In order to study a time series and its volatility at different time scales, we want to have a tool similar to a wavelet transform,[10] which adapts to causal signals. The motivation is to reveal structures of price movements related to certain frequencies. Similar to wavelet transforms, we want a double representation in time and frequency, but we do not require an invertible transformation because our aim is to analyze rather than further process the signal. This gives us more flexibility in the choice of the transformations. A simple causal kernel with such properties is like $\mathrm{ma}[\tau](t)\ \sin(kt/\tau)$, where $\mathrm{ma}[\tau](t)$ is still the MA kernel of Equation 3.57. Essentially, the sine part is (locally) analyzing the signal at a frequency k/τ and the MA part is taking a causal window of range τ. As we want a couple of oscillations in the window 2τ, we choose k between $k \sim \pi$ and $k \sim 5\pi$. Larger k values increase the frequency resolution at the cost of the time resolution. The basic idea is to compute an EMA with a complex τ; this is equivalent to including a sine and cosine part in the kernel. The nice computational iterative property of the moving average is preserved. The first step is to study complex iterated EMAs. The kernel of the complex ema is defined as

$$\mathrm{ema}[\zeta](t) = \frac{e^{-\zeta t}}{\tau} \qquad \text{where} \quad \zeta = \frac{1}{\tau}(1 + ik) \tag{3.77}$$

where ζ is complex but τ is again a real number. The choice of the normalization factor $1/\tau$ is somewhat arbitrary (a factor $|\zeta|$ will produce the same normalization for the real case $k = 0$) but leads to a convenient definition of the windowed Fourier kernel that follows. By using the convolution formula, one can prove iteratively that the kernel of the complex $\mathrm{EMA}[\zeta, n]$ is given by

$$\mathrm{ema}[\zeta, n](t) = \frac{1}{(n-1)!} \left(\frac{t}{\tau}\right)^{n-1} \frac{e^{-\zeta t}}{\tau} \tag{3.78}$$

which is analogous to Equation 3.54. The normalization is such that, for a constant function $c(t) = c$,

$$\mathrm{EMA}[\zeta, n; c] = \frac{c}{(1 + ik)^n} \tag{3.79}$$

[10] An introduction to wavelet methods is studied extensively in Gençay *et al.* (2001b).

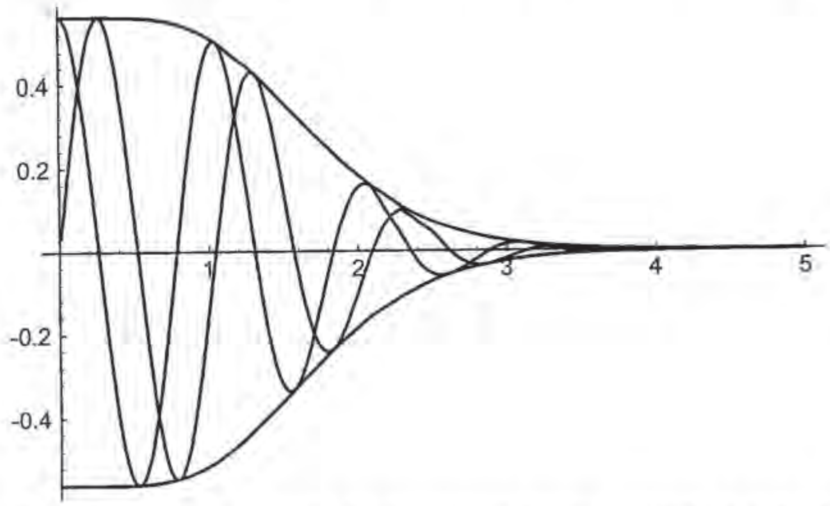


FIGURE 3.14 The kernel wf(t) for the windowed Fourier operator, for $n = 8$ and $k = 6$. Three aspects of the complex kernel are shown: (1) the envelope (= absolute value), (2) the real part (starting on top), and (3) the imaginary part (starting at zero).

Similar to Equation 3.51, we obtain an iterative computational formula for the complex EMA:

$$\begin{aligned} \text{EMA}[\zeta; z](t_n) = & \\ & \mu \, \text{EMA}[\zeta; z](t_{n-1}) + \frac{\nu - \mu}{1 + ik} z_{n-1} + \frac{1 - \nu}{1 + ik} z_n \end{aligned} \tag{3.80}$$

with

$$\begin{aligned} \alpha &= \zeta \, (t_n - t_{n-1}) \\ \mu &= e^{-\alpha} \end{aligned}$$

where ν depends on the chosen interpolation scheme as given by Equation 3.52. We define the (complex) kernel wf(t) of the windowed Fourier transform WF as

$$\begin{aligned} \text{wf}[\tau, k, n](t) &= \text{ma}[\tau, n](t) \, e^{-ikt/\tau} \\ &= \frac{1}{n} \sum_{j=1}^{n} \frac{1}{(j-1)!} \left(\frac{t}{\tau}\right)^{j-1} \frac{e^{-\zeta t}}{\tau} \\ &= \frac{1}{n} \sum_{j=1}^{n} \text{ema}[\zeta, j](t) \end{aligned} \tag{3.81}$$

The kernel is shown in Figure 3.14. Another appropriate name for this operator might be CMA for "complex moving average." The normalization is such that, for a constant function $c(t) = c$.

$$N_{\text{WF}} = \text{WF}[\zeta, n; c] = \frac{c}{n} \sum_{j=1}^{n} \frac{1}{(1+ik)^j}$$

To provide a more convenient real quantity, with the mean of the signal subtracted, we can define a (nonlinear) normed windowed Fourier transform as

$$\text{NormedWF}[\zeta, n; z] = |\text{WF}[\zeta, n; z] - N_{\text{WF}}\ \text{MA}[\tau, n; z]| \qquad (3.82)$$

The normalization is chosen so that

$$\text{NormedWF}[\zeta, n; c] = 0$$

In Equation 3.82, we are only interested in the amplitude of the measured frequency; by taking the absolute value we have lost information on the phase of the oscillations.

Windowed Fourier transforms can be computed for a set of different τ values to obtain a full spectrum. However, there is an upper limit in the range of computable frequencies. Results are reliable if τ clearly exceeds the average time interval between ticks. For τ values smaller than the average tick interval, results become biased and noisy; this sentence applies not only to windowed Fourier transforms but also to most other time series operators.

Figure 3.15 shows an example of the normed windowed Fourier transform for the example week. The stock market crash is again nicely spotted as the peak on Tuesday, October 28.

Using our computational toolbox of operators, other quantities of interest can be easily derived. For example, we can compute the relative share of a certain frequency in the total volatility. This would mean a volatility correction of the normed windowed Fourier transform. A way to achieve this is to divide NormedWF by a suitable volatility, or to replace z by the standardized time series $\hat{z}$ in Equation 3.82.

3.4 MICROSCOPIC OPERATORS

As discussed in Section 3.1, it is in general better to use macroscopic operators because they are well behaved with respect to the sampling frequency. Some microscopic operators allow the extraction of tick-related information at the highest possible frequency. An example of such an operator is the microscopic volatility defined later. The computation of the tick frequency requires (by definition) microscopic operators. We also want to extend to inhomogeneous time series the usual operators applied to homogeneous time series, such as the shift operator.

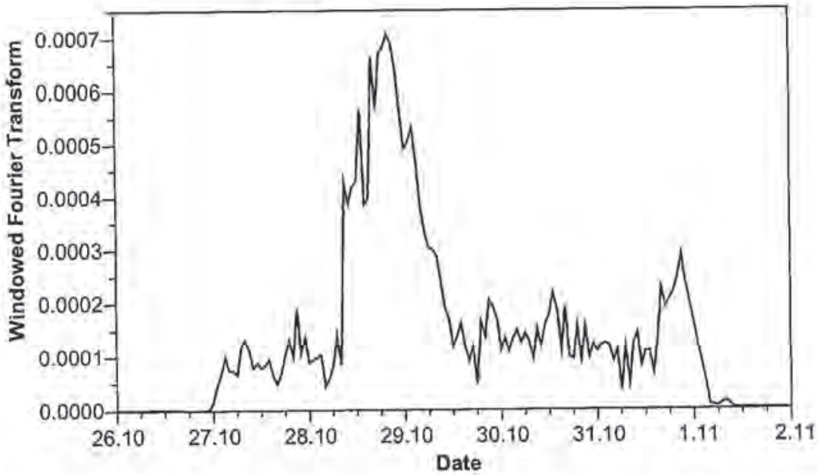


FIGURE 3.15 The normed windowed Fourier transform, with $\tau = 1$ hr, $k = 6$ and $n = 8$.

3.4.1 Backward Shift and Time Translation Operators

The backward shift operator $\mathcal{B}$ shifts the value of the time series by one event backward $\mathcal{B}[z]_i = (t_i, z_{i-1})$, but the time associated to each event is not changed. Some authors use the equivalent *lag* operator L instead. It shifts the time series values but leaves the time part untouched. The inverse operator $\mathcal{B}^{-1}$ will shift the series forward. It is well defined for regular and irregular time series. Only for a homogeneous time series spaced by δt, this operator is equivalent to a time translation by $-\delta t$ (followed by a shift of the time *and* value series by one event with respect to the irrelevant index i).

The operator $\mathcal{T}$ translates the time series by δt forward $\mathcal{T}[\delta t; z]_i = (t_i + \delta t, z_i)$; namely it shifts the time part but leaves the time series values untouched. Note that for an inhomogeneous time series, this operator defines a series with another set of time points.

3.4.2 Regular Time Series Operator

From the time series z, irregularly spaced in time, the operator $RTS[t_0, \delta t]$ constructs an artificial homogeneous time series at times $t_0 + k\delta t$, regularly spaced by δt, rooted at t_0. This involves an interpolation scheme as discussed in Section 3.2.1. Depending on this scheme, the RTS operator can be causal or not. The regular time series can also be constructed as being regular on a given business time scale rather than in physical time.

The RTS operator allows us to move from inhomogeneous to the homogeneous time series as presented in Section 3.2.2. For many computations, it is mandatory to have homogeneous data, for example when modeling financial data with ARMA or GARCH processes. Another example is the computation of empirical probability distributions. Such computations are done with a smooth version of the formula

$$p(z) = \frac{1}{T} \int_0^T dt \; \delta(z - z'(t)) \tag{3.83}$$

with $z'(t)$ the (continuously interpolated) empirical data. In the time integral, a measure can be added, or the integral can be evaluated in business time to account for the seasonalities. The evaluation of the time integral is computationally heavy, and it is much simpler to generate a regular time series and to use the familiar binning procedure to obtain a histogram of z. Note also that a moving probability distribution can be defined by replacing the time integral by an MA operator (see the remark at the end of Section 3.3.7).

3.4.3 Microscopic Return, Difference, and Derivative

From the tick-by-tick price time series, the microscopic return for a quote is defined as $r_j = x_j - x_{j-1}$. This return can be attributed to one quote, even if, strictly speaking, it is related to two subsequent quotes. Note that this is a "microscopic" definition that involves neither a time scale nor an interpolation scheme. Using the backward shift operator $\mathcal{B}$, the return time series can be defined as

$$r = x - \mathcal{B}[x] = (1 - \mathcal{B})\, x = \delta x \tag{3.84}$$

where the microscopic difference operator[11] is $\delta = (1 - \mathcal{B})$. The lag n difference operator is defined by $\delta[n] = (1 - \mathcal{B}^n)$.

The microscopic derivative operator ∂ is defined as

$$\partial[\delta t_0] x_j = \frac{x_j - x_{j-1}}{\delta t_0 + t_j - t_{j-1}} = \left.\frac{\delta x}{\delta t_0 + \delta t}\right|_j \tag{3.85}$$

The constant δt_0 regularizes the expression when $t_i = t_{i-1}$. A reasonable value of δt_0 must be small; the actual choice depends on the application. Similar to the macroscopic γ-derivative, a microscopic γ-derivative can be defined as

$$\partial[\delta t_0, \gamma] x = \frac{\delta x}{(\delta t_0 + \delta t)^\gamma} \tag{3.86}$$

The best parameters should follow a study yet to be done for the random process of x. The constant δt_0 regularizes the expression when $t_j = t_{j-1}$.

These derivatives are potentially very noisy and can be averaged. In general, the macroscopic derivative D (Equation 3.64) seems more relevant for applications to random processes.

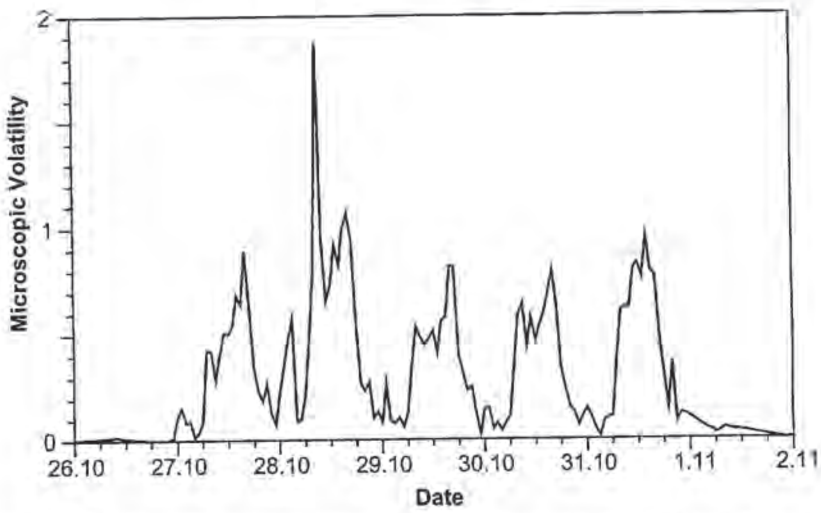


FIGURE 3.16 Microscopic volatility, computed with $\gamma = 0.5$, $\delta t_0 = 0.001$ seconds, the time interval expressed in years (annualized), and $\tau = 1$hr.

3.4.4 Microscopic Volatility

The microscopic volatility is defined as the norm of the microscopic derivative,

$$\text{Microscopic volatility}[\tau; z] = \text{MNorm}[\tau/2; \partial z] \tag{3.87}$$

which also depends on the implicit parameters δt_0 and γ of ∂z. Let us emphasize that this definition does not require a regular time series (and that it is not an MA of the macroscopic definition of volatility). In a way, this definition uses all of the information available on the process z. The constant τ controls the range on which the volatility is computed. The microscopic volatility for our standard example week is displayed in Figure 3.16.

3.4.5 Tick Frequency and Activity

The tick frequency $f(t_j)$ counts the number of ticks per time unit. One definition based on regular time intervals is already given by Equation 3.15 (see also Guillaume *et al.* (1997), for example). In general, the tick frequency at time t_j is defined as

$$f[T](t) \;=\; \frac{1}{T} N\{t_j \mid t_j \in [t - T, t]\} \tag{3.88}$$

where $N\{t_j\}$ counts the number of elements in a set and where T is the sample time interval during which the counting is computed. The tick frequency has

[11] The operator δ should not be confused with the δ function used in Chapter 3.

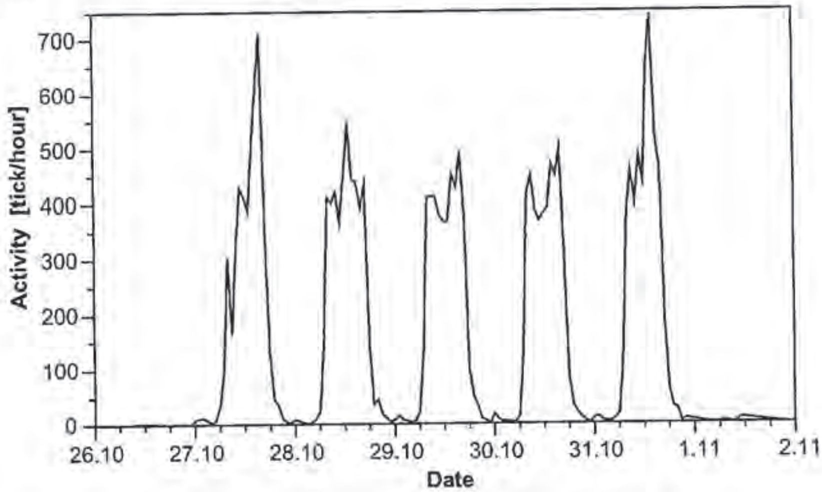


FIGURE 3.17 Tick activity A as defined by Equation 3.90, ticks per hour, computed with τ = 1hr. The five working days of the example week can be clearly seen.

the dimension of an inverse time and is expressed in units such as ticks/minute or ticks/day. This simple definition has some properties that may not always be desired:

- The formula is computationally cumbersome when computed on a moving sample, especially for large T.
- It is an average over a rectangular window. We often prefer moving averages whose kernel (= weighting function) fades more smoothly in the distant past.
- If no quotes are in the interval spanned by T, this definition will give $f = 0$. A related problem is the unusable limit $T \to 0$ if one wants to measure an instantaneous quote rate.

For these reasons, we prefer the definition in Equation 3.90. The tick rate is defined as

$$a[\delta t_0]_j = \frac{1}{\delta t_0 + t_j - t_{j-1}} \tag{3.89}$$

The tick rate has the same dimension as the tick frequency. This definition has the advantage of being related only to the time interval between two subsequent ticks. Following Equation 3.89, an activity can be attributed to one tick, analogous to a return that is attributed to the j^{th} tick by $r_j = x_j - x_{j-1} = (\delta x)_j$. The activity A is the average tick rate during a time interval τ:

$$A[\tau; z] = \mathrm{MA}[\tau/2; a[z]] \tag{3.90}$$

To avoid the spurious singularity when $t_{j-1} = t_j$, the MA operator has to be evaluated with the next-point interpolation (see Equation 3.52). This makes the computation numerically stable even when an extremely small value of δt_0 in Equation 3.89 is chosen.

Figure 3.17 shows the behavior of A in our example week. At first glance, the activity A looks rather different from the tick frequency f. Yet an interesting feature of this definition is to be equivalent to the tick frequency f when the MA operator is a rectangular moving average with $\tau = T/2$ (this is easy to prove by computing the integral of the piecewise constant function a). However, the activity A has some advantages such that it is much simpler to compute on a moving window, and the weighting function of the past can be controlled through the choice of the MA kernel.

4

ADAPTIVE DATA CLEANING

4.1 INTRODUCTION: USING A FILTER TO CLEAN THE DATA

High-frequency data are commercially transmitted as a piece of real-time information to human users, usually traders. These data users are professionals who know the context (e.g., the market state and the likely level of a quoted price). If bad data is transmitted, professional users immediately understand, and implicitly clean the data by using information they have in their personal information set. They do not need additional human or computerized input to check the correctness of the data.

The situation changes if the data users are different, such as researchers investigating historical high-frequency data or computer algorithms that extract real-time information for a given purpose (e.g., a trading algorithm, risk assessment). If bad quotes are used, the results are inevitably bad and totally unusable in the case of aberrant outliers. In the experience of the authors and many other researchers, almost every high-frequency data source contains some bad quotes. Data cleaning is a necessity; it has nothing to do with manipulation or cosmetics.

Data cleaning is a very technical topic. Readers interested in economic results rather than methods and researchers enjoying the privilege of possessing cleaned high-frequency data may skip the remainder of Chapter 4.

A data cleaning methodology requires some criteria to decide on the correctness and possible elimination of quotes. As long as the data set is not too large, human judgment may be a sufficient criterion. In this book, however, we focus on high-frequency data with thousands and millions of observations. Therefore, the criteria have to be formalized through a statistical model that can be implemented as a computer algorithm. Such an algorithm is called a data *filter*. In this chapter, the term "filter" is exclusively used within the context of data cleaning and the term "filtering" is a synonym of "cleaning." Data cleaning is done as a first, independent step of analysis, before applying any time series operator as studied in Chapter 3 and before statistically analyzing the resulting time series. We choose this approach because it is universally applicable, regardless of the type of further analysis. There is a less favorable alternative to prior data filtering: *robust statistics*, where all the data (also outliers) are included in the main statistical analysis. The methods of robust statistics depend on the nature of the analysis and are not universally applicable.

Cleaning a high-frequency time series is a demanding, often underestimated task. It is complicated for several reasons:

- The variety of possible errors and their causes
- The variety of statistical properties of the filtered variables (distribution functions, conditional behavior, structural breaks)
- The variety of data sources and contributors of different reliability
- The irregularity of time intervals (sparse/dense data, sometimes data gaps of long duration)
- The complexity and variety of the quoted data as discussed in Chapter 2: transaction prices, indicative prices, FX forward points (where negative values are allowed), interest rates, figures from derivative markets, transaction volumes, bid-ask quotes versus single-valued quotes
- The necessity of real-time filtering (some applications need instant information before seeing successor quotes)

The data cleaning algorithm presented here is *adaptive* and also presented in Müller (1999). The algorithm learns from the data while sequentially cleaning a time series. It continuously updates its information base in real time.

Further guidelines are needed in a filtering methodology:

- The cause of data errors is rarely known. Therefore the validity of a quote is judged according to its *plausibility* given the *statistical properties* of the series.[1]

[1] We have to distinguish true, plausible movements from spurious movements due to erroneous quotes. Brock and Kleidon (1992) suggest decomposing observed movements in the data according to three causes: (1) erroneous quotes, (2) bid-ask spread dynamics due to the pressure on trading factors, and (3) other economic forces.

- A neighborhood of quotes, called the *filtering window*, is needed to judge the credibility of a quote. Such a data window can grow and shrink according to data quality.
- Quotes with a complex structure (i.e., bid-ask) are split into scalar variables to be filtered separately. The filtered variables are derived from the raw variables (e.g., the logarithm of a bid price or the bid-ask spread). Some special error types may also be analyzed for full quotes before data splitting.
- Numerical methods with convergence problems (such as model estimation and nonlinear minimization) are avoided. The chosen algorithm produces well-defined results in all situations.
- The filter needs to be computationally fast. This requirement excludes algorithms starting from scratch for each new incoming tick. The chosen algorithm is *sequential* and *iterative*. It uses the existing filter information base when a new tick arrives, with a minimum amount of updating.
- The filter has two modes, which are the *real-time* and the *historical* modes. Due to the windowing technique, both modes are supported by the same filter. In historical filtering, the final validation of a quote is delayed until successor quotes have been seen.

4.2 DATA AND DATA ERRORS

4.2.1 Time Series of Ticks

The object of data cleaning is a time series of *ticks*. The term "tick" stands for "quote" in a very general sense: any variable that is quoted, from any origin and for any financial instrument. The time-ordered sequence of ticks is inhomogeneous in the general case where the time intervals between ticks vary in size. Normally, one time series is filtered independently from other series. The multivariate cleaning of several time series together is discussed in Section 4.8.1.

The ticks of the series must be of the same type. They may differ in the origins of the contributors, but should not differ in important parameters such as the maturity (of interest rates, etc.) or the moneyness (of options or implied volatilities). If a data-feed provides bid or ask quotes (or transaction quotes) alternatively in random sequence, we advise splitting the data stream into independent bid and ask streams. Normal bid-ask *pairs*, however, are appropriately handled inside the filter.

The following data structure of ticks is assumed:

1. A time stamp.
2. The tick level(s) of which the data cleaning algorithm supports two kinds:
   (a) Data with one level (a price or transaction volume, etc.), such as a stock index.

(b) Data with two levels: bid-ask pairs, such as foreign exchange (FX) spot rates.

3. Information on the origin of the tick, e.g. an identification code of the contributor (a bank or broker). For some financial instruments, notably those traded at an exchange, this is trivial or not available.

A data feed may provide some other information which is not utilized by the filter.

4.2.2 Data Error Types

A *data error* is a piece of quoted data that does not conform to the real situation of the market. A price quote has to be identified as a data error if it is neither a correctly reported transaction price nor a possible transaction price at the reported time. We have to tolerate some transmission time delays and small deviations especially in the case of indicative prices.

There are many causes for data errors. The errors can be separated into two classes:

1. Human errors: Errors directly caused by human data contributors, for different reasons:
   - Unintentional errors, such as typing errors
   - Intentional errors, such as dummy ticks produced just for technical testing
2. System errors: Errors caused by computer systems, their interactions and failures

Human operators have the ultimate responsibility for system errors. However, the distance between the data error and the responsible person is much larger for system errors than for "human" errors. In many cases, it is impossible to find the exact reason for the data error even if a tick is very aberrant. The task of the filter is to identify such outliers whatever the reason. Sometimes the cause of the error can be guessed from the particular behavior of the bad ticks. This knowledge of the error mechanism can help to improve filtering and, in few cases, allow the correction of bad ticks.

The following error types are so particular that they need special treatment.

1. Decimal errors: Failure to change a "big" decimal digit of the quote. For instance, a bid price of 1.3498 is followed by a true quote 1.3505, but the published, bad quote is 1.3405. This error is most damaging if the quoting software is using a *cache* memory somewhere. The wrong decimal digit may stay in the cache and cause a long series of bad quotes. Around 1988, this was a dominant error type.
2. "Test": Some data contributors sometimes send test ticks to the system, usually at times when the market is not liquid. These test ticks can cause a lot of damage because they may look plausible to the filter, at least initially. Two important examples follow:

- "Early morning test": A contributor sends a bad tick very early in the morning to test whether the connection to the data distributor is operational. If the market is inactive overnight, no trader would take this test tick seriously. For the filter, such a tick may be a major challenge. The filter has to be very critical to first ticks after a data gap.
- Monotonic series: Some contributors test the performance and the time delay of their data connection by sending a long series of linearly increasing ticks at inactive times such as overnight or during a weekend. This is hard for the filter to detect because tick-by-tick returns look plausible. Only the monotonic behavior in the long run can be used to identify the fake nature of this type of data.

3. Repeated ticks: Some contributors let their computers repeat the last tick in more or less regular time intervals. This is harmless if it happens in a moderate way. In some markets with coarse granularity of tick values (such as short-term interest rate futures), repeated tick values are quite natural. However, some contributors repeat old ticks thousands of times with high frequency, thereby obstructing the validation of the few good ticks produced by other, more reasonable contributors.
4. Tick copying: Some contributors employ computers to copy and re-send the ticks of other contributors, as explained in Section 2.2.3. If these ticks are on a correct level, a filter has no reason to care—with one exception. Some contributors run computer programs to produce slightly modified ticks by adding small random corrections. Such slightly varying copy ticks are damaging because they obstruct the clear identification of fake monotonic or repeated series made by other contributors.
5. Scaling problem: Quoting conventions may differ or be officially redefined in some markets. Some contributors may quote the value of 100 units, others the value of 1 unit. Scaling factors are often integer powers of 10, but other values may occur (for stock splits in equity markets). The filter will run into this problem "by surprise" unless a human filter user anticipates all scale changes and preprocesses the data accordingly.

A complete data cleaning tool has to include algorithmic elements to deal with each of these special error types.

4.3 GENERAL OVERVIEW OF THE FILTER

4.3.1 The Functionality of the Filter

The flowcharts in Figure 4.1 illustrate some typical applications of a data cleaning filter in a larger context. Normal users simply want to eliminate "invalid" data from the stream, but the chart on the right-hand side shows that the filter can also deliver more information on the ticks and their quality.

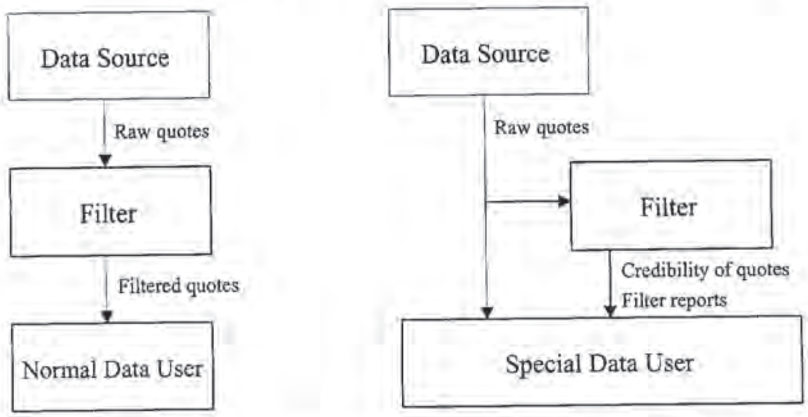


FIGURE 4.1 Data cleaning (filtering): Normal users want to eliminate bad ticks from the application (left chart). In special cases, users want to know filtering results such as the credibility or the reason for rejecting a tick (right chart).

A filter has some configuration parameters depending on the type of instrument, as to be shown later. Once it is created, it performs the following operations:

1. It receives financial ticks in the ordered sequence of their time stamps.
2. It delivers the same ticks in the same ordered sequence, plus the filter results. For each tick, the following results are delivered:
    - Credibility values of the tick and of its individual elements (such as bid, ask, bid-ask spread); the credibility is defined between 0 (totally invalid) and 1 (totally valid)
    - The value(s) of the tick, whose errors can possibly be *corrected* in some cases where the error mechanism is well known
    - The "filtering reason," which is a formalized piece of text explaining why the filter has rejected (or corrected) the tick

Normal users use only those (possibly corrected) ticks with a credibility exceeding a threshold value (which is often chosen to be 0.5). They ignore all invalid ticks and all side results of the filter such as the filter report. The timing of the filter operations is nontrivial. In *real-time* operation, the result of a filter is used right after the tick has entered the filter. In *historical* operation, the user takes the corrected result after the filter has seen a few newer ticks and adapted the credibility of older ticks.

The filter needs a *build-up period* to learn from the data. This is natural for an adaptive filter. If the data cleaning operation starts at the first available tick (beginning of data series), the build-up means to run the filter for a few weeks from this point, storing a set of statistical variables in preparation for restarting the filter from the first available tick. The filter will then be well adapted because it can use the previously stored statistical variables. If the data cleaning operation starts at some later point in time, the natural build-up period is a period immediately preceding the first tick needed.

The filtering algorithm can be seen as one whole block that can be used several times in a data flow such as the following:

- Mixing already filtered data streams from several sources where the mixing result is again filtered. The danger is that the combined filters reject too many quotes, especially in the real-time filtering of fast moves (or price jumps).
- Filtering combined with computational blocks: raw data → filter → computational block → filter → application. Some computational blocks such as cross rate or yield curve computations require filtered input and produce an output that the user may again want to filter.

Repeated filtering of the same time series is rather dangerous because it may lead to too many rejections of quotes. If it cannot be avoided, only one of the filters in the chain should be of the standard type. The other filter(s) should be configured to be weak (i.e., they should eliminate not more than the obviously aberrant outliers).

4.3.2 Overview of the Filtering Algorithm and Its Structure

The filtering algorithm is structured in a hierarchical scheme of subalgorithms. Table 4.1 gives an overview of this structure for a univariate filter for one financial instrument. A higher hierarchy level at the top of Table 4.1 can be added for multivariate filtering, as discussed in Section 4.8.1.

Details of the different algorithmic levels are explained in the next sections. The sequence of these sections follows Table 4.1, *from bottom to top*. Some special filter elements are not treated there, but are briefly described in Section 4.8.

4.4 BASIC FILTERING ELEMENTS AND OPERATIONS

The first element to be discussed in a bottom-to-top specification is the scalar filtering window. Its position in the algorithm is shown in Table 4.1.

The basic filtering operations utilize the quotes in the simplified form of *scalar quotes* consisting of the following:

1. The time stamp
2. One scalar variable value to be filtered (e.g., the logarithm of a bid price), here denoted by x
3. The origin of the quote (as in the full quote of Section 4.2.1)

The basic operations can be divided into two types:

1. Filtering of single scalar quotes: considering the credibility of one scalar quote alone. An important part is the *level filter* where the level of the filtered variable is the criterion.
2. Pair filtering: comparing two scalar quotes. The most important part is the *change filter* that considers the change of the filtered variable from one

TABLE 4.1 Basic structure of the filtering algorithm used for data cleaning. The data cleaning algorithm has three main hierarchy levels, each with its specific functionalities.

Hierarchy level	Level name	Purpose, description
1	Univariate filter	The complete filtering of one time series: Passing incoming ticks to the lower hierarchy levels Collecting the filter results of the lower hierarchy levels and packaging them into the right output format Supporting real-time and historical filtering Supporting one or more filtering hypotheses, each with its own full-tick filtering window
2	Full-tick filtering window	A sequence of recent full ticks (bid-ask), some of them possibly corrected according to a general filtering hypothesis. The tasks are as follows: Tick splitting: splitting a full tick into scalar quotes to be filtered in their own scalar filtering windows Basic validity test (e.g., whether prices are positive) A possible mathematical transformation (e.g., logarithm) All those filtering steps that require full ticks (not just bid or ask ticks alone)
3	Scalar filtering window	A sequence of recent scalar quotes whose credibilities are still being modified. The tasks are as follows: Testing new, incoming scalar quotes Comparing a new scalar quote to all older quotes of the window (using a business time scale and a dependence analysis of quote origins) Computing a first credibility of the new scalar quote; modifying the credibilities of older quotes based on new information Dismissing the oldest scalar quotes when their credibility is finally settled Updating the statistics with good scalar quotes when they are dismissed from the window

quote to another one. Filtering depends on the time interval between the two quotes and the time scale on which this is measured. Pair filtering also includes a comparison of quote origins.

The basic filtering operations and another basic concept of filtering, credibility are presented in the following sections. Their actual application in the larger algorithm is explained later, starting from Section 4.5.

4.4.1 Credibility and Trust Capital

Credibility is a central concept of the filtering algorithm. It is expressed by a variable C taking values between 0 and 1, where 1 indicates validity and 0 invalidity. This number can be interpreted as the probability of a quote being valid according to a given criterion. For two reasons, we avoid the formal introduction of the term "probability." First, the validity of a quote is a fuzzy concept (e.g., slightly deviating quotes of an over-the-counter spot market can perhaps be termed valid even if they are very unlikely to lead to a real transaction). Second, we have no model of

TABLE 4.2 Adding independent credibility values.

The total credibility C_{total} resulting from two independent credibility values C_1 and C_2. The function $C_{total} = C[T(C_1) + T(C_2)]$ defines an addition operator for credibilities, based on Equations 4.1 and 4.2. The values in brackets, (0.5), are in fact indefinite limit values where C_{total} may converge to any value between 0 and 1.

C_{total}	$C_1 =$ 0	0.25	0.5	0.75	1
$C_2 =$					
1	(0.5)	1	1	1	1
0.75	0	0.5	0.75	0.878	1
0.5	0	0.25	0.5	0.75	1
0.25	0	0.122	0.25	0.5	1
0	0	0	0	0	(0.5)

probability even if validity could be exactly defined. Credibility can be understood as a "possibility" in the sense of fuzzy logic as proposed by Zimmermann (1985), for example.

Credibility is not additive; the credibility of a scalar quote gained from two tests is not the sum of the credibilities gained from the individual tests. This follows from the definition of credibility between 0 and 1. The sum of two credibilities of, say, 0.75 would be outside the allowed domain.

For internal credibility computations based on different tests, an additive variable is needed to obtain the joint view of all tests. We define the additive *trust capital* T, which is unlimited in value. There is no theoretical limit for gathering evidence in favor of accepting or rejecting the validity hypothesis. Full validity corresponds to a trust capital of $T = \infty$, full invalidity to $T = -\infty$. We impose a fixed, monotonic relation between the credibility C and the trust capital T of a certain object

$$C(T) = \frac{1}{2} + \frac{T}{2\sqrt{1+T^2}} \quad (4.1)$$

and the inverse relation

$$T(C) = \frac{C - \frac{1}{2}}{\sqrt{C\,(1-C)}} \quad (4.2)$$

There are possible alternatives to this functional relationship. The chosen solution has some advantages in the formulation of the algorithm that will be shown later.

The additivity of trust capitals and Equations 4.1 and 4.2 imply the definition of an addition operator for credibilities. Table 4.2 shows the total credibility resulting from two independent credibility values.

4.4.2 Filtering of Single Scalar Quotes: The Level Filter

There is only one analysis of a single quote called the level filter. Comparisons between quotes (done for a pair of quotes, treated in Section 4.4.3) are often more important in filtering than the analysis of a single quote.

The level filter computes a first credibility of the value of the filtered variable. This only applies to those volatile but mean-reverting time series where the levels as such have a certain credibility in the absolute sense—not only the level changes. Moreover, the timing of the mean reversion should be relatively fast. Interest rates or interest rates futures prices, for example, are mean-reverting only after time intervals of years; they appear to be freely floating within smaller intervals (see Ballocchi, 1996). For those rates and for other prices, level filtering is not suitable.

The obvious example for fast mean reversion and thus for using a level filter is the bid-ask spread, which can be rather volatile from quote to quote but tends to stay within a fixed range of values that varies only very slowly over time. For spreads, an adaptive level filter is at least as important as a pair filter that considers the spread change between two quotes.

The level filter first puts the filtered variable value x into the perspective of its own statistical mean and standard deviation. Following the notation of Section 3.3.8, the standardized variable $\hat{x}$ is defined by

$$\hat{x} = \frac{x - \bar{x}}{\mathrm{MSD}[\,\Delta\vartheta_r, 2;\ x\,]} = \frac{x - \bar{x}}{\sqrt{\mathrm{EMA}[\,\Delta\vartheta_r;\ (x - \bar{x})^2\,]}} \tag{4.3}$$

where the mean value of x is also a moving average:

$$\bar{x} = \mathrm{EMA}[\,\Delta\vartheta_r;\ x\,] \tag{4.4}$$

The time scale used for this computation is called ϑ. Taking a business time scale ϑ as introduced in Section 4.4.6 leads to better data cleaning than taking physical time. The variable $\Delta\vartheta_r$ denotes the configurable range of the kernel of the moving averages and should cover the time frame of the mean reversion of the filtered variable; a reasonable value for bid-ask spreads has to be chosen. The iterative computation of moving averages is explained in Section 3.3.5. Here and for all the moving averages of the filtering algorithm, a simple exponentially weighted moving average (EMA) is used for efficiency reasons.

A small $|\hat{x}|$ value deserves high trust; an extreme $|\hat{x}|$ value indicates an outlier with low credibility and negative trust capital. Before arriving at a formula for the trust capital as a function of $\hat{x}$, the *distribution* of $\hat{x}$ has to be discussed. A symmetric form of the distribution is assumed at least in coarse approximation. This is ensured by the definition of the filtered variable x, which is a mathematically transformed variable. The exact definition of x is deferred to Section 4.6.3 in the chosen structure of this chapter.

The amount of negative trust capital for outliers depends on the tails of the distribution at extreme (positive and negative) $\hat{x}$ values. A reasonable assumption is that the credibility of outliers is approximately the probability of exceeding the outlier value, given the distribution function. This probability is proportional to $\hat{x}^{-\alpha}$ where α is called the tail index. We know that density functions of level-filtered variables such as bid-ask spreads are fat-tailed (see Müller and Sgier, 1992). Determining the distribution and α in a moving sample would be a considerable task, certainly too heavy for filtering software. Therefore, we choose an approximate assumption on α that was found acceptable across many rates, filtered variable types and financial instruments: $\alpha = 4$. This value is also used in the analogous pair filtering tool (e.g., for price changes, and discussed in Section 4.4.3).

For extreme events, the relation between credibility and trust capital, Equation 4.1, can be asymptotically expanded as follows

$$C = \frac{1}{4\,T^2} \quad \text{for } T \ll -1 \tag{4.5}$$

Terms of order higher than $(1/T)^2$ are neglected here. Defining a credibility proportional to $\hat{x}^{-\alpha}$ is thus identical to defining a trust capital proportional to $\hat{x}^{\alpha/2}$. Assuming $\alpha = 4$, we obtain a trust capital proportional to $\hat{x}^2$. For outliers, this trust capital is negative, but for small $\hat{x}$, the trust capital is positive up to a maximum value we define to be 1.

Now, we have the ingredients to come up with a formulation that gives the resulting trust capital of the i^{th} quote according to the level filter:

$$T_{i0} = 1 - \xi_i^2 \tag{4.6}$$

where the index 0 of T_{i0} indicates that this is a result of the level filter only. The variable ξ_i is $\hat{x}$ in a scaled and standardized form:

$$\xi_i = \frac{\hat{x}_i}{\xi_0} \tag{4.7}$$

with a constant ξ_0. Equation 4.6 together with Equation 4.7 is the simplest possible way to obtain the desired maximum and asymptotic behavior. For certain rapidly mean-reverting variables such as hourly or daily trading volumes, this may be enough.

However, the actual implementation for bid-ask spreads has some special properties. Filter tests have shown that these properties have to be taken into account in order to attain satisfactory spread filter results:

- Quoted bid-ask spreads tend to cluster at "even" values (e.g., 10 basis points,) whereas the real spread may be an odd value oscillating in a range below the quoted value. A series of formal, constant spreads can therefore hide some substantial volatility that is not covered by the statistically determined denominator of Equation 4.3. We need an offset $\Delta x_{\min}^2$ to account for the typical hidden volatility in that denominator. A suitable choice is $\Delta x_{\min}^2 = [\text{constant}_1(\bar{x} + \text{constant}_2)]^2$.

- High values of bid-ask spreads are worse in usability than low spreads, by nature. Thus the quote deviations from the mean as defined by Equation 4.3 are judged with bias. Deviations to the high side ($\hat{x}_i > 0$) are penalized by a factor p_{high}, whereas no such penalty is applied against low spreads.
- For some (minor) financial instruments, many quotes are posted with zero spreads (i.e., bid quote = ask quote). This is discussed in Section 4.6.1. In some cases, zero spreads have to be accepted, but we set a penalty against them as in the case of positive $\hat{x}_i$.

We obtain the following refined definition of ξ_i

$$\xi_i = \begin{cases} \dfrac{\hat{x}_i}{\xi_0} & \text{if } \hat{x}_i \leq 0 \text{ and no zero-spread case} \\ p_{high} \dfrac{\hat{x}_i}{\xi_0} & \text{if } \hat{x}_i > 0 \text{ or in a zero-spread case} \end{cases} \tag{4.8}$$

where $\hat{x}_i$ comes from a modified version of Equation 4.3,

$$\hat{x} = \frac{x - \bar{x}}{\sqrt{\mathrm{EMA}[\,\Delta\vartheta_r;\ (x-\bar{x})^2\,] + \Delta x^2_{\mathrm{min}}}} \tag{4.9}$$

The constant ξ_0 determines the size of an $\hat{x}$ that is just large enough to neither increase nor decrease the credibility.

Equation 4.8 is general enough for all mean-reverting filterable variables. If we introduced mean-reverting variables other than the bid-ask spread, a good value for $\Delta x^2_{\mathrm{min}}$ would probably be much smaller or even 0, p_{high} around one and ξ_0 larger (to tolerate volatility increases in absence of a basic volatility level $\Delta x^2_{\mathrm{min}}$).

4.4.3 Pair Filtering: The Credibility of Returns

The pairwise comparison of scalar quotes is a central basic filtering operation. The algorithm makes pairwise comparisons also for quotes that are not neighbors in the series, as explained in Section 4.5.

Pair filtering contains several ingredients, the most important one being the change filter. Its task is to judge the credibility of a variable change (= return if the variable is a price). The time difference between the two quotes plays a role, so the time scale on which it is measured has to be specified. The criterion is *adaptive* to the statistically expected volatility estimate and therefore uses some results from a moving statistical analysis.

The change of the filtered variable x from the j^{th} to the i^{th} quote is

$$\Delta x_{ij} = x_i - x_j \tag{4.10}$$

The variable x may be the result of a transformation in the sense of Section 4.6.3. The time difference of the quotes is $\Delta\vartheta_{ij}$, measured on a time scale to be discussed in Section 4.4.6.

The expected variance $V(\Delta\vartheta)$ of x around zero is determined by the on-line statistics as described in Section 4.4.4. The relative change is defined by

$$\xi_{ij} = \frac{\Delta x_{ij}}{\xi_0 \sqrt{V(\Delta\vartheta_{ij})}} \quad (4.11)$$

with a positive constant ξ_0, which has a value of around 5.5 and is further discussed later. Low $|\xi|$ values deserve high trust, extreme $|\xi|$ values indicate low credibility and negative trust capital; at least one of the two compared quotes must be an outlier.

The remainder of the algorithm is similar to that of the level filter as described in Section 4.4.2, using the relative change ξ_{ij} instead of the scaled standardized variable ξ_i.

The amount of negative trust capital for outliers depends on the density function of changes Δx, especially the tail of the distribution at extreme Δx or ξ values. A reasonable assumption is that the credibility of outliers is approximately the probability of exceeding the outlier value, given the distribution function. This probability is proportional to $\xi^{-\alpha}$, where α is the tail index of a fat-tailed distribution. We know that distributions of high-frequency price changes are indeed fat-tailed (see Dacorogna *et al.*, 2001a). Determining the distribution and α in a moving sample would be a considerable task beyond the scope of filtering software. Therefore, we make a rough assumption on α that is good enough across many rates, filtered variable types and financial instruments. For many price changes, a good value is around $\alpha \approx 3.5$, according to Dacorogna *et al.* (2001a) and Müller *et al.* (1998). As in Section 4.4.2, we generally use $\alpha = 4$ as a realistic, general approximation.

As in Section 4.4.2 and together with Equation 4.5, we argue that the trust capital should asymptotically be proportional to ξ^2 and arrive at a formula that gives the trust capital as a function of ξ:

$$U_{ij} = U(\xi_{ij}^2) = 1 - \xi_{ij}^2 \quad (4.12)$$

which is analogous to Equation 4.6. This trust capital, depending only on ξ, is called U to distinguish it from the final trust capital T that is based on more criteria. At $\xi = 1$, Equation 4.12 yields a zero trust capital, neither increasing nor decreasing the credibility. Intuitively, a variable change of a few standard deviations might correspond to this undecided situation; smaller variable changes lead to positive trust capital, larger ones to negative trust capital. In fact, the parameter ξ_0 of Equation 4.11 should be configured to a high value, leading to a rather tolerant behavior even if the volatility V is slightly underestimated.

The trust capital U_{ij} from Equation 4.12 is a sufficient concept under the best circumstances, independent quotes separated by a small time interval. In the general case, a modified formula is needed to solve the following three special pair filtering problems.

1. Filtering should stay a local concept on the time axis. However, a quote has few close neighbors and many more distant neighbors. When the additive

trust capital of a quote is determined by pairwise comparisons to other quotes as explained in Section 4.5.2, the results from distant quotes must not dominate those from the close neighbors; the interaction range should be limited. This is achieved by defining the trust capital proportional to $(\Delta\vartheta)^{-3}$ (assuming a constant ξ) for asymptotically large quote intervals $\Delta\vartheta$.

2. For large $\Delta\vartheta$, even moderately aberrant quotes would be too easily accepted by Equation 4.12. Therefore, the aforementioned decline of trust capital with growing $\Delta\vartheta$ is particularly important in the case of positive trust capital. Negative trust capital, on the other hand, should stay strongly negative even if $\Delta\vartheta$ is rather large. The new filter needs a selective decline of trust capital with increasing $\Delta\vartheta$: fast for small ξ (positive trust capital), slow for large ξ (negative trust capital). This treatment is essential for data holes or gaps, where there are no (or few) close neighbor quotes.
3. Dependent quotes: if two quotes originate from the same source, their comparison can hardly increase the credibility (but it can reinforce negative trust in the case of a large ξ). In Section 4.4.5, we introduce an independence variable I_{ij} between 0 (totally dependent) and 1 (totally independent).

The two last points imply a certain asymmetry in the trust capital; gathering evidence in favor of accepting a quote is more delicate than evidence in favor of rejecting it.

All of these concerns can be taken into account in an extended version of Equation 4.12. This is the final formula for the trust capital from a change filter:

$$T_{ij} = T(\xi_{ij}^2, \Delta\vartheta_{ij}, I_{ij}) = I_{ij}^* \frac{1-\xi_{ij}^4}{1+\xi_{ij}^2+\left(\frac{d\,\Delta\vartheta_{ij}}{\nu}\right)^3} \tag{4.13}$$

where

$$I_{ij}^* = \begin{cases} I_{ij} & \text{if } \xi_{ij}^2 < 1 \\ 1 & \text{if } \xi_{ij}^2 \geq 1 \end{cases} \tag{4.14}$$

The independence I_{ij} is always between 0 and 1 and is computed by Equation 4.23. The variable d is a quote density explained in Section 4.4.4. The configurable constant ν determines a sort of filtering interaction range in units of the typical quote interval ($\approx 1/d$).

Table 4.3 shows the behavior of the trust capital according to Equation 4.13. The trust capital converges to zero with an increasing quote interval $\Delta\vartheta$ much more rapidly for small variable changes $|\xi|$ than for large ones. For small $\Delta\vartheta_{ij}$ and $I_{ij} = 1$, Equation 4.13 converges to Equation 4.12.

The approach of Equation 4.13 has been tested for almost all available types of financial data, not only FX. We find that it works for all data types with the same values of the parameters.

TABLE 4.3 Trust capital as a function of two variables.

The trust capital T resulting from a comparison of two independent ($I^* = 1$) scalar quotes, depending on two variables: the relative variable change ξ and the time interval $\Delta\vartheta$ between the quotes. ξ is defined by Equation 4.11, and d and ν are explained in the text.

T	$d\,\Delta\vartheta\,/\,\nu =$ 0	0.5	1	2	4
$\lvert\xi\rvert =$					
4	-15.0	-14.9	-14.2	-10.2	-3.2
2	-3.0	-2.9	-2.5	-1.2	-0.22
1	0	0	0	0	0
0.5	0.75	0.68	0.42	0.10	0.014
0	1	0.89	0.50	0.11	0.015

4.4.4 Computing the Expected Volatility

The expected volatility is a function of the size of the time interval between the quotes and thus requires a larger computational effort than other statistical variables. Only credible scalar quotes should be used in the computation. The updates of all statistics are therefore managed by another part of the algorithm that knows about final credibilities as explained in Section 4.5.5.

Choosing an appropriate business time scale ϑ is important for measuring the time intervals between quotes and for all other computations of this section. This is explained in Section 4.4.6.

Although the expected volatility computation can be implemented with various methods of different degrees of sophistication, we adopt a simple method. The first variable needed is the quote density

$$d = \mathrm{EMA}\left[\Delta\vartheta_r;\ \frac{c_d}{\delta\vartheta}\right] \tag{4.15}$$

This is a moving average in the notation of Section 3.3.5; $\delta\vartheta$ is the time interval between two "valid" (as defined on a higher level) neighbor quotes on the chosen time scale. $\Delta\vartheta_r$ is the configurable range of the kernel of the moving average. The variable c_d is the weight of the quote, which normally has a value of $c_d = 1$ and is lower only in the case of repeated quote values. The iterative computation of moving averages is explained in Section 3.3.5. The value $1/\delta\vartheta$ has to be assumed for the whole quote interval, which implies using the "next point" interpolation. It can be shown that a zero value of $\delta\vartheta$ does not lead to a singularity of the EMA.

An annualized squared "micro"-volatility is defined as a variance in the form of a moving average

$$v = \mathrm{EMA}\left[\Delta\vartheta_r;\ \frac{(\delta x)^2}{\delta\vartheta + \delta\vartheta_0}\right] \tag{4.16}$$

where the notation follows Sections 3.3.5 and 3.4.3 and the range $\Delta\vartheta_r$ is the same as in Equation 4.15. δx is the change of the filtered variable between (sufficiently credible) neighbor quotes. There is a small time interval offset

$$\delta\vartheta_0 = \max\left(\frac{d_0}{d},\ \delta\vartheta_{\min}\right) \tag{4.17}$$

The small positive term $\delta\vartheta_0$ accounts for some known short-term behaviors of markets: (1) certain asynchronicities in the quote transmissions, (2) some temporary market level inconsistencies that need time to be arbitraged out, (3) a negative autocorrelation of many market prices over short time lags (see Section 5.2.1). However, $\delta\vartheta_0$ is not needed to avoid singularities of v; even a zero value of both $\delta\vartheta$ and $\delta\vartheta_0$ would not lead to a singularity of the EMA. The "next point" interpolation is again appropriate in the EMA computation.

Strictly speaking, v can be called annualized only if ϑ is measured in years, but the choice of this unit does not matter in our algorithm. The exponent of the annualization is not too important because the different values of $\delta\vartheta$ share the same order of magnitude.

Experience shows that the volatility measure of the filter should not rely only on one variance v as defined here. It is more stable to use three such volatilities: v_{fast}, v and v_{slow}. All of them are computed by Equation 4.16, but they differ in their ranges, $\Delta\vartheta_r$, where v_{fast} has a short range, v a medium-sized range, and v_{slow} a long range. Our expected volatility is defined to be the maximum of the three:

$$v_{\exp} = \max(v_{\text{fast}}, v, v_{\text{slow}}) \tag{4.18}$$

This is superior to taking only v. In case of a market shock, the rapid growth of v_{fast} allows for a quick adaptation of the filter, whereas the inertia of v_{slow} prevents the filter from forgetting volatile events too rapidly in a quiet market phase.

From the annualized $v_{\exp}$, we obtain the expected squared change as a function of the time interval $\Delta\vartheta$ between two quotes. At this point, the filter needs a special element to prevent the filter from easily accepting price changes over large data gaps, time periods with no quotes. Data gaps are characterized by a large value of $\Delta\vartheta$ and very few quotes within this interval. In case of data gaps, an upper limit of $\Delta\vartheta$ is enforced:

$$\Delta\vartheta_{\text{corr}} = \min\left[\frac{2.5\,Q}{d},\ \max\left(\frac{0.1\,Q}{d},\ \Delta\vartheta\right)\right] \tag{4.19}$$

where d is taken from Equation 4.15 and Q is the number of valid quotes in the interval between the two quotes; this is explained in Section 4.5.2. Equation 4.19

also sets a lower limit of $\Delta\vartheta_{\mathrm{corr}}$ in case of a very high frequency of valid quotes. It is important to validate fast trends with many quotes.

The corrected quote interval $\Delta\vartheta_{\mathrm{corr}}$ is now used to compute the expected squared change V

$$V = V(\Delta\vartheta_{\mathrm{corr}}) = (\Delta\vartheta_{\mathrm{corr}} + \delta\vartheta_0)\, v_{\mathrm{exp}} + V_0 \tag{4.20}$$

This function $V(\Delta\vartheta_{\mathrm{corr}})$ is needed in the trust capital calculation of Section 4.4.3 and inserted in Equation 4.11. The positive offset V_0 is small and could be omitted in many cases with no loss of filter quality. However, a small $V_0 > 0$ is desirable. Some quotes are quoted in coarse granularity (i.e., the minimum step between two possible quote values is rather large as compared to the volatility). This is the case in some interest rate futures and also for bid-ask spreads (in FX markets), which often have a rounded size of 5, 10, or 20 basis points with rarely a value in between. Quotes with coarse granularity have a *hidden* volatility such that a series of identical quotes may hide a movement of a size smaller than the typical granule. The term V_0 thus represents the hidden volatility:

$$V_0 = 0.25\, g^2 + \varepsilon_0^2 \tag{4.21}$$

where the granule size g is also determined by adaptive methods. (The granularity analysis is also needed in the analysis of repeated ticks, which is not explained here.) The extremely small term ε_0^2 just has the numerical task to keep $V_0 > 0$.

The term ε_0^2, however, plays a special role if the scalar variable to be filtered is a bid-ask spread. The spread filter is the least important filter, but leads to the highest number of rejections of FX quotes if it is configured similar to the filter of other scalars. This fact is not accepted by typical filter users who want a more tolerant spread filter. A closer inspection shows that different contributors of bid-ask quotes often have different spread quoting policies. They are often interested only in the bid or ask side of the quote and tend to push the other side off the real market by choosing a spread too large. Thus the spreads of neighbor quotes may have different sizes even in quiet markets. In some minor FX markets, some contributors even mix retail quotes with very large spreads into the stream of interbank quotes. In order not to reject too many quotes for spread reasons, we have to raise the tolerance for fast spread changes and reject only extreme jumps in spreads. This means raising ε_0^2: $\varepsilon_0 = \mathrm{constant}_1\,(\bar{x} + \mathrm{constant}_2)$, where $\bar{x}$ is defined by Equation 4.4. This choice of ε_0 can be inferred from the mapping of the bid-ask spread in Equation 4.45. When a filter is initialized, we set $V_0 = \varepsilon_0^2$ and replace this by Equation 4.21 as soon as the granule size estimate g is available, based on statistics from valid quotes.

4.4.5 Pair Filtering: Comparing Quote Origins

Pair filtering results can add some credibility to the two quotes only if these are independent. Two identical quotes from the same contributor do not add substantial confidence to the quoted level—the fact that an automated quoting system sends the same quote twice does not make this quote more reliable. Two nonidentical quotes from the same contributor may imply that the second quote has been produced to correct a bad first one. Another interpretation might be that an automated quoting system has a random generator to send a sequence of slightly varying quotes to mark presence on the information system. Different quotes from entirely different contributors are the most reliable case for pair filtering.

The basic tool is a function to compare the origins of the two quotes, considering the main source (the information provider), the contributor ID (bank name), and the location information. This implies that available information on contributors has a value in data cleaning and should be collected rather than ignored. An "unknown" origin is treated just like another origin name. The resulting independence measure I'_{ij} is confined between 0 for identical origins and 1 for clearly different origins. In some cases (e.g., same bank but different subsidiary), a value between 0 and 1 can be chosen.

I'_{ij} is not yet the final formulation but has to be put in relation with the general origin *diversity* of the time series. An analysis of data from only one or very few origins must be different from that of data with a rich variety of origins. The general diversity D can be defined as a moving average of the $I'_{i\,i-1}$ of valid neighbor quotes,

$$D = \text{EMA}[\,\text{tick-time},\ R;\ I'_{i\,i-1}\,] \tag{4.22}$$

where R is the range (center of gravity) of the kernel. The "tick-time" is a time scale that is incremented by one at each new quote. The "next point" interpolation is again appropriate in the EMA computation. Only "valid" quotes are used; this is possible on a higher level of the algorithm (see Section 4.5.5). By doing so, we prevent D from being lowered by bad mass quotes from a single computerized source. Thus we are protected against a difficult filtering problem. The high number of bad mass quotes from a single contributor will not force the filter to accept the bad level.

The use of D makes the independence variable I_{ij} adaptive through the following formula:

$$I_{ij} = I'_{ij} + f(D)\,(1 - I'_{ij}) \tag{4.23}$$

with

$$f(D) = \frac{0.0005 + (1 - D)^8}{2.001} \tag{4.24}$$

If the diversity is very low (e.g., in a single-contributor source), this formula (reluctantly) raises the independence estimate I_{ij} to allow for some positive trust capital to build up. For a strictly uniform source ($I' = D = 0$), I_{ij} will reach 0.5, which is one half of the I_{ij} value of truly independent quotes in a multicontributor series.

The output variable I_{ij} resulting from Equation 4.14 is always confined between 0 and 1 and is generally used in Equation 4.14. Some special cases need a special discussion:

- Repeated quotes. Rarely, the raw data contains long series of repeated quotes from the same contributor, and the obtained value of I_{ij} may still be too high. A solution would be a special filtering element focused on repeated ticks.
- High-quality data. The collected data may be mixed with old, historical, commercially available daily data that were of distinctly higher quality than the data from a single, average-quality contributor. When comparing two quotes from this historical daily data, we may force $I'_{ij} = 1$ although these quotes come from the same "contributor." This special filtering element is necessary only if there are huge, proven quality differences between contributors.
- In multivariate filtering (see Section 4.8.1), artificial quotes that might be injected by a multivariate covariance analysis should have $I'_{ij} = 1$ when compared to each other or to any other quote.

4.4.6 A Time Scale for Filtering

Time plays a role in the adaptive elements of the level filter as well as in almost all parts of the change filter. Value changes are tolerated more easily when separated by a large time interval between the time stamps. When using the term "time interval," we need to specify the time scale to be used.

The algorithm works with any time scale, but some are more suitable than others. If our tolerance for quote level changes is as large over weekends as over working hours, we have to accept almost any bad quote from the few weekend contributors. These weekend quotes are sometimes test quotes or other outliers in the absence of a liquid market. Our solution is a time scale that compresses the weekends and other inactive periods and thus leads to a lower tolerance.

Accounting for the low weekend activity is vital, but the exact treatment of typical volatility patterns during working days is less important. Therefore, we cannot accept using only physical time (= calendar/clock time), but the following solutions are possible:

1. A very simple business time with two states: active (working days) and inactive (weekend from Friday 21:00:00 GMT to Sunday 21:00:00 GMT, plus the most important and general holidays). The speed of this business time as compared to physical time would be either 1.4 (in active state) or 0.01 (in inactive state).
2. An adaptively weighted mean of three simple, generic business time scales ϑ with smoothly varying weights according to built-in statistics. This solution suits those filter developers that prefer to avoid the complex ϑ technology of Chapter 6.

TABLE 4.4 Active periods of the three generic markets.

Daytimes limiting the active periods of three generic, continent-wide markets; in Greenwich Mean Time (GMT). The scheme is coarse, modeling just the main structure of worldwide financial markets. The active periods differ according to local time zones and business hours. The Asian market starts on the day before from the viewpoint of the GMT time zone.

Market	k	$t_{\mathrm{start},k}$	$t_{\mathrm{end},k}$
East Asia	1	21:00	7:00
Europe	2	6:00	16:00
America	3	11:00	21:00

3. An adaptively weighted mean of three generic business time scales ϑ as defined by Chapter 6 or Dacorogna *et al.* (1993).

The second solution differs from the third one only in the definition of the basic ϑ-time scales. The adaptivity mechanism is the same for both solutions.

Three generic ϑ-times are used, based on typical volatility patterns of three main markets: Asia, Europe, and America. In the second solution, these ϑ times are defined as follows:

$$\frac{\mathrm{d}\vartheta_k}{\mathrm{d}t} = \begin{cases} 3.4 & \text{if } t_{\mathrm{start},k} \leq t_d < t_{\mathrm{end},k} \text{ on a working day} \\ 0.01 & \text{otherwise (inactive times, weekends, holidays)} \end{cases} \tag{4.25}$$

where t_d is the daytime in Greenwich Mean Time (GMT) and the generic start and end times of the working-daily activity periods are given by Table 4.4. They correspond to typical observations in several markets. The active periods of exchange-traded instruments are subsets of the active periods of Table 4.4. The time scales ϑ_k are time integrals of $\mathrm{d}\vartheta_k/\mathrm{d}t$ from Equation 4.25. Thus the time ϑ_k flows either rapidly in active market periods or very slowly in inactive periods. Its long-term average speed is similar to physical time. The implementation of Equation 4.25 requires some knowledge about holidays. The database of holidays to be applied may be rudimentary (e.g., Christmas holidays) or more elaborate to cover all main holidays of the financial centers on the three continents. The effect of daylight saving time is neglected here as the market activity model is coarse.

If the three ϑ_k-times are chosen as defined by Chapter 6 (the third solution of the list), effects like daylight saving time and local holidays (i.e., characteristic for one continent) are also covered. The activity in the morning of the geographical markets is higher than in the afternoon—a typical behavior of FX rates and, even more so, interest rates, interest rate futures, and other exchange-traded markets.

Once the three scales ϑ_k are defined (by the integrals of Equation 4.25 in our suggestion), their adaptively weighted mean is constructed and used as the time scale ϑ for filtering. This ϑ-time is able to approximately capture the daily and weekly seasonality and the low volatility of holidays. High precision is not required as ϑ is only one among many ingredients of the data cleaning algorithm, many of which are based on rather coarse approximations. This is the definition of ϑ-time:

$$\vartheta = \sum_{\text{all } k} w_k \, \vartheta_k \tag{4.26}$$

with

$$\sum_{\text{all } k} w_k = 1 \tag{4.27}$$

where "all k" means "all markets." This is three in our case, but the algorithm also works for any other number of generic markets. The weights w_k are adaptive to the actual behavior of the volatility. A high w_k reflects a high fitness of ϑ_k, which implies that the volatility measured in ϑ_k has low seasonal variations.

The determination of the w_k might be done with methods such as the maximum likelihood estimation of a volatility model. However, this would be unreliable given the local convergence issues and the existing modeling limitations of Equation 4.26. The proposed heuristic method always returns an unambiguous solution. The volatility of changes of the filtered variable is measured on all ϑ_k-scales in terms of a variance similar to Equation 4.16:

$$\sigma_k = \sqrt{\text{EMA}\left[\, \Delta\vartheta_{\text{smooth}};\ \frac{(\delta x)^2}{\delta\vartheta_k + \delta\vartheta_0} \,\right]} \tag{4.28}$$

where $\delta\vartheta_k$ is the interval between validated neighbor quotes in ϑ_k-time, δx is the corresponding change of the filtered variable, $\delta\vartheta_0$ is defined by Equation 4.17 and the time scale of the EMA is ϑ_k-time. The notation is as in Sections 3.3.5 and 3.4.3. Smoothing with a short range $\Delta\vartheta_{\text{smooth}}$ is necessary to diminish the influence of quote-to-quote noise. The EMA computation assumes a constant value of $(\delta x)^2/(\delta\vartheta_k + \delta\vartheta_0)$ for the whole quote interval. This means the "next point" interpolation of Equation 3.52.

The fluctuations of the variable σ_k indicate the badness of the ϑ_k model. In the case of a bad fit, σ_k is often very low (when the ϑ_k-scale expands time) and sometimes very high (when the ϑ_k-scale compresses time). The fluctuations are quantified in terms of the variance F_k,

$$\begin{aligned} F_k &= \text{EMA}[\, \Delta\vartheta_r;\ (\, \sigma_k - \text{EMA}[\, \Delta\vartheta_r;\ \sigma_k \,]\,)^2 \,] \\ &= \text{MVar}[\, \Delta\vartheta_r,\ 2;\ \sigma_k \,] \end{aligned} \tag{4.29}$$

where the time scale is ϑ_k-time; the MVar operator is explained in Section 3.3.8.

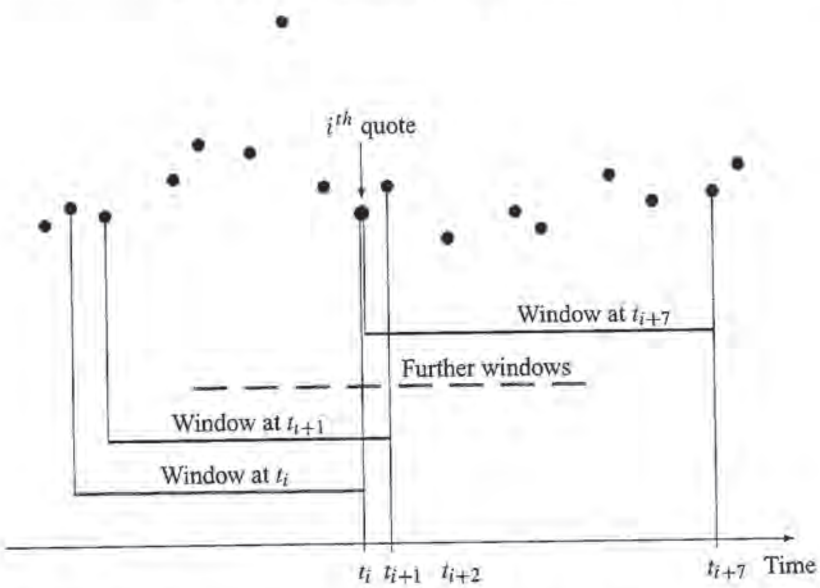


FIGURE 4.2 The scalar filtering window moves forward in time by including new scalar quotes and dismissing old ones.

The range $\Delta\vartheta_r$ has to be suitably chosen. In our approximation, the fluctuations directly define the weight of the k^{th} market:

$$w_k = \frac{1}{F_k \sum_{\text{all } k'} \frac{1}{F_{k'}}} \tag{4.30}$$

which satisfies Equation 4.27 and can be inserted into Equation 4.26.

4.5 THE SCALAR FILTERING WINDOW

The scalar filtering window is located at the bottom of the hierarchical structure of the algorithm as shown in Table 4.1. It covers the set of all recent scalar quotes contained in a time interval. This neighborhood of quotes is used to judge the credibility of new incoming scalar quotes. In the course of the analysis, these new quotes are included and old quotes are dismissed at the back end of the window following a certain rule. Thus the window is moving forward in time. This mechanism is illustrated by Figure 4.2.

All the scalar quotes within the window have a provisional credibility value, which is modified with new incoming quotes. When the quotes leave the window, their credibilities are regarded as finally determined. Sufficiently credible quotes are then used to update the statistics needed for adaptivity.

At the initialization of a filter from scratch, the window is empty. When the first scalar quote enters, it cannot be filtered by pair filtering yet, only the level filter applies.

4.5.1 Entering a New Quote in the Scalar Filtering Window

Whenever a new scalar quote enters the window, an analysis is made based on earlier results and the new quote.

There are two possible ways in which a new quote enters the scalar filtering window:

1. The normal update. A new scalar quote from the data source enters, is analyzed, and finally becomes the newest member of the scalar filtering window. The window variables are updated accordingly. These operations are described by Sections 4.5.2 through 4.5.6.
2. A filter test. A new scalar quote from any source is merely tested. It is analyzed as in a normal update, but it does not become a member of the window. No window variable is changed by this test. Thus we execute the steps of Section 4.5.2 and avoid those of Sections 4.5.3 through 4.5.6. The resulting trust capital of the new scalar quote is returned.

4.5.2 The Trust Capital of a New Scalar Quote

The algorithm of the filtering window is organized in an iterative way. Whenever a new quote enters the window, an update is made based on earlier results and an analysis of the new quote.

When the new, i^{th} scalar quote arrives, it already satisfies certain basic validity criteria (e.g., a price is not negative) and has possibly been transformed to a logarithmic value. This is ensured by the higher-level quote splitting algorithm explained in Section 4.6. The following filtering operations are done with the incoming i^{th} scalar quote:

1. The base trust capital T_{i0} is computed as the result of the level filter, Equation 4.6, if the scalar quote is a bid-ask spread. Otherwise, $T_{i0} = 0$. The resulting T_{i0} of Equation 4.6 is multiplied by a configured constant c_{level} that determines the importance of level filtering.
2. The new quote is compared to all old quotes of the window through pair filtering steps as described in Section 4.4.3. The trust capitals T_{ij} resulting from Equation 4.13 determine the trust capital T_i of the new quote and also affect the trust capitals T_j of the old quotes.

For computing T_{ij}, we need the expected squared value change V from Equation 4.20 and $\Delta\vartheta_{\text{corr}}$ from Equation 4.19 and therefore the number Q of valid quotes in the time interval from quote j to quote i. For this, we use the valid-quote age Q_j of the old quotes

$$Q = Q_j + 1 \qquad (4.31)$$

The increment by 1 stands for the new quote, which is not yet included in the window. The computation of Q_j is explained at the end of Section 4.5.3. The resulting value of Q is inserted in Equation 4.19.

The trust capital of the new, i^{th} quote is computed additively as follows

$$T_i' = c_{\text{level}}\, T_{i0} + \sum_{j=i-n}^{i-1} C_j\, T_{ij} \tag{4.32}$$

T_i' is not termed T_i because it is not yet the final trust capital in some cases. Equation 4.32 is a weighted sum with weights $C_j = C(T_j)$ from Equation 4.1, which are the current credibilities of the n other quotes of the window.

The number n of quotes used for comparison to the i^{th} quote has an influence on the trust capital and thus the credibility. The higher the value of n, the higher the trust capital according to Equation 4.32 (provided that we are in a series of good data). This effect reflects the fact that the more comparisons to other quotes, the more certain our judgment on credibility. However, the effect of increasing n by adding more and more remote quotes is marginal. The remoteness of quotes implies a high term proportional to $(\Delta\vartheta_{ij})^3$ in the denominator of Equation 4.13, so the resulting T_{ij} values are close to zero. The choice of n is further discussed in Section 4.5.4.

Equation 4.32 is a conservative concept insofar as it judges the credibility of a new quote in the light of the previously obtained credibilities C_j of the earlier quotes. In the case of an unusually large real move or price jump, new quotes on a new level might be rejected for a prolonged time. To prevent this, there is special treatment of "after-jump" situations, which may lead to a correction of the resulting trust capital T_i and a quicker acceptance of a new level after a jump.

The first step of the after-jump algorithm is to identify the location of a possible real jump within the scalar filtering window. This is done during the computation of Equation 4.32. At every j, we test whether the incomplete sum of that equation

$$T'_{i,\text{at } j} = T_{i0} + \sum_{j'=i-n}^{j-1} C_{j'}\, T_{ij'} \tag{4.33}$$

is less than the critical value T_{crit}

$$T_{\text{crit}} = \mu\, c_{\text{level}}\, T_{i0} - 1 \tag{4.34}$$

(where μ is defined below). At the same time, we test $T_{ij} > 0$ (this indicates having reached a new, stable level after the jump rather than an outlier). At the first j where *both* conditions are satisfied, we conclude that a value jump must have taken place somewhere before quote $j-1$. Although this jump certainly happened before quote j, we define $j_{\text{jump}} = j$ because this is the index of the first quote where we have a reason to believe that the jump was real. In order to validate this possible real value jump, we initialize an *alternative* trust capital T_i''

$$T''_{i,\text{at } j_{\text{jump}}} = T_{\text{crit}} - 0.5 + \mu\,(T'_{i,\text{at } j} - T_{\text{crit}}) \tag{4.35}$$

We *dilute* the normal trust capital $T'_{i,\text{at } j}$ by a small dilution factor μ. When the filter is initialized (before seeing some 10 acceptable quotes), we choose a slightly larger μ value in order to prevent the filter from being trapped by an initial outlier. The offset term -0.5 in Equation 4.35 prevents the alternative hypothesis from being too easily accepted. For all values of $j \geq j_{\text{jump}}$, we set

$$T''_j = \mu\,T_j \tag{4.36}$$

and insert these diluted trust capitals T''_j of old quotes in Equation 4.1. The resulting credibilities C''_j are used to complete the computation of the alternative trust capital T''_i:

$$T''_i = T''_{i,\text{at } j_{\text{jump}}} + \sum_{j=j_{\text{jump}}}^{i-1} C''_j\,T_{ij} \tag{4.37}$$

analogous to Equation 4.32.

Now, we decide whether to take the normal, conservative trust capital T'_i or the alternative T''_i. The resulting, final trust capital is

$$T_i = \begin{cases} T''_i & \text{if } T''_i > T'_i \text{ and } T''_i > 0 \\ T'_i & \text{otherwise} \end{cases} \tag{4.38}$$

The alternative solution prevails if its trust capital exceeds 0 and the trust capital of the conservative solution. The trust capital T_i of the new quote is the end result of a pure filter test. In the case of a normal update, the window has to be updated.

4.5.3 Updating the Scalar Window

A new quote affects the trust capitals of the old quotes of the window. The most dramatic change happens in the case of accepting the alternative hypothesis according to Equation 4.38. In this case, a real value jump is acknowledged, which leads to a major reassessment of the old quotes. First, the pairwise trust capital of quote comparisons across the jump is diluted

$$T_{\text{corr},ij} = \begin{cases} \mu\,T_{ij} & \text{for } j < j_{\text{jump}} \\ T_{ij} & \text{otherwise} \end{cases} \tag{4.39}$$

In the normal case with no jump, $T_{\text{corr},ij} = T_{ij}$. Afterward, the quotes after the newly detected jump get a new opportunity

$$T_{j,\text{new}} = \begin{cases} \mu\,T_j & \text{if } j \geq j_{\text{jump}} \text{ and } T_j < 0 \\ T_j & \text{otherwise} \end{cases} \tag{4.40}$$

In the case of a jump, this new value $T_{j,\text{new}}$ replaces T_j.

In every case, whether there is a jump or not, the trust capitals of all quotes are finally updated additively following Equation 4.32

$$T_{j,\text{new}} = T_j + C_i \, T_{\text{corr},ij} \, , \quad \text{for } j = i-n \ldots i-1 \tag{4.41}$$

where $C_i = C(T_i)$ follows from Equation 4.1 by substituting T_i from Equation 4.38. The result $T_{j,\text{new}}$ of Equation 4.41 is replacing the old value T_j. It should also be clarified that the diluted values T_j'' from Equation 4.36 are never directly used to modify the trust capitals T_j.

In historical filtering, Equations 4.39 through 4.41 may lead to the rehabilitation of an initially rejected old quote. Even in real-time filtering, the corrected trust capital of an old quote indirectly contributes to the filtering of new quotes through Equation 4.32 and through the use of only sufficiently credible old quotes in the statistics of adaptive filtering.

The valid-quote age Q_j of all the old quotes is also updated

$$Q_{j,\text{new}} = Q_j + C_i \, , \quad \text{for } j = i-n \ldots i-1 \tag{4.42}$$

where $C_i = C(T_i)$. The more credible the new quote, the higher the increment of the valid-quote age Q_j.

After all these updates, the new quote with index i and with its newly computed trust capital T_i is inserted in the window as its newest member, with the valid-quote age Q_i initialized to zero.

4.5.4 Dismissing Quotes from the Scalar Window

The window does not grow infinitely. At the end of a normal update as described in Section 4.5.3, a rule for dismissing scalar quotes is applied. There are three criteria for obtaining a properly sized window: (1) a sufficient time interval, (2) a sufficient number of quotes, and (3) a sufficient *overall* credibility of all scalar quotes. These criteria are listed here in the sequence of increasing importance.

In our general quote dismissal rule, we use the product of the criteria. At the end of an update with a new quote, the following condition for dismissing the oldest quote (with index $i-n$) is evaluated:

$$(\vartheta_i - \vartheta_{i-n+1}) \, n^2 \left(\sum_{j=0}^{n-1} C_{i-j} \right)^6 \geq W \tag{4.43}$$

The sum of credibilities, the overall credibility, is the most important criterion and is therefore raised to the sixth power. This exponent is a parameter as many others; the value 6 has been found optimal in tests of samples of different data frequencies and qualities. The configuration parameter W defines the sufficient size of the window and has the dimension of a time. The parameter W is somehow related to the parameter ν of Equation 4.13, which determines a filtering range. Choosing a very large W when ν is limited does not add value because the distant quotes have a negligible weight in this case.

A few considerations may illustrate the behavior of Equation 4.43. If the data in the window are of good quality, the window is of small size. As soon as a cluster of low-quality or doubtful data enters the window, it will grow (sometimes to a very large size) until the situation becomes clearer and most old quotes can be dismissed again. In the case of a sparse time series, the window may contain few quotes but these quotes will extend further in time than for a dense time series. After dismissing the oldest quote when Equation 4.43 is fulfilled, the whole quote dismissal procedure is repeated as long as the remaining window still satisfies Equation 4.43.

In very rare cases, the window grows to a very large size and the filtering algorithm becomes slow. This problem and its solution are discussed in Section 4.5.6. Aside from this, another safety measure is taken where a quote older than 300 days is dismissed from the window even if Equation 4.43 is not fulfilled, as long as the remaining window still has at least two quotes.

Dismissed scalar quotes are also reported to the higher level of the filtering algorithm. This is necessary in the case of historical filtering for producing the final filtering results.

4.5.5 Updating the Statistics with Credible Scalar Quotes

When a scalar quote is dismissed from the window, its credibility C_i has reached a final value that will no longer be changed where $C_i = C(T_i)$ results from Equation 4.1. This is the right moment to update all the statistics needed for the adaptivity of the filter.

Invalid quotes are excluded from these statistics and they are simply ignored when updating the statistical variables. We set a critical credibility level C_{crit} where only quotes with credibility values above C_{crit} are used for updating the statistics. However, we should not be too rigid when excluding quotes. The filter has to adapt to unexpected events such as sudden volatility increases, but this requires including also some mildly rejected quotes. In fact, tests have shown that only the totally invalid quotes should be excluded here. We choose a low critical credibility level. In the initial phase, right after a filter starts from initialization (before having seen 10 acceptable quotes), we take a larger, more cautious value.

If a dismissed quote has a credibility $C_i > C_{\text{crit}}$, we update all the statistics. These updates typically imply the computation of moving average iteration formulas, and the statistics are explained in Sections 4.4.2 through 4.4.6.

4.5.6 A Second Scalar Window for Old Valid Quotes

The quote dismissal rule of Equation 4.43 makes sure that the scalar window stays reasonably small—except in the case of a very long series of bad quotes. Such long, rarely occurring series usually consist of computerized quotes (e.g., repeated or monotonic quotes). The filtering window technology as described so far is well able to handle this case, but the computation time of the filter grows very much in the case of a very large window. In real time, this does not really matter, but historical filtering becomes slow.

For efficiency reasons, the filter therefore supports a second queue of old valid quotes. The normal scalar window size is strictly limited to a maximum number of quotes, but an old quote dismissed from the normal window is stored in a second scalar window if its credibility exceeds a low threshold value. Otherwise, the dismissed quote is treated as any dismissed quote, as explained before, including the updating of statistics and the final reporting of its credibility.

This second scalar window of old valid quotes is normally empty. As soon as one or more dismissed quotes are in this window, it is treated as a part of the normal scalar window in all computations. The trust capital computation of Equation 4.32, for example, has a sum over both scalar windows, starting at the window of old valid quotes. The window of old valid quotes stays and possibly grows as long as the quote dismissal condition (applied to the two joined scalar windows) is not fulfilled. When the condition is fulfilled, the oldest quote of the scalar window of old valid quotes is deleted. After deleting all of its quotes, the second window is again empty and filtering is back to the normal mode.

The concept of a second scalar filtering window for old valid quotes adds quite some complexity to the filter and is motivated only by computational efficiency.

4.6 THE FULL-QUOTE FILTERING WINDOW

The full-quote filtering window is managed on hierarchy level 2 of Table 4.1. It is basically a sequence of recent full quotes plus a set of algorithmic methods of managing and processing this sequence. The full-quote filtering window has the following tasks:

- Splitting the quotes into scalar quotes that can be used in the filtering operations of Section 4.4.
- A first basic validity test for the filtered variables. This is usually a domain test (e.g., rejecting negative prices). Rejected scalar quotes are marked as invalid ($C_i = 0$) and eliminated from all further tests. They do not enter a scalar filtering window.
- In many cases, a transformation of the quoted level such as taking logarithms of prices instead of raw price values.
- Creating independent filtering environments for all types of scalar quotes, each with its own scalar filtering window.
- Storing the credibility of dismissed scalar quotes until all the other scalar quotes belonging to the same full quote have also been dismissed. (The spread filter may dismiss quotes before the bid price filter, for example.)
- Storing the full quotes as long as two or more filtering hypotheses coexist, until one of them wins. This is decided by the next higher hierarchy level (see Section 4.7). The decision between filtering hypotheses can also be made fast enough to make this point superfluous.

- When a full quote is finally dismissed, reporting it, together with its filtering results, to the higher level (needed only in historical filtering).

In principle, the full-quote filtering window also offers the opportunity of analyzing those data errors that affect full quotes in a way that cannot be analyzed when just looking at scalar quotes after splitting. In our experience, we have never found a good reason to implement this (aside from the filtering hypotheses discussed in Section 4.7).

The full quotes may enter a full-quote filtering window in a form already corrected by a filtering hypothesis. This fact plays no role here since the algorithm of the full-quote window does not care about quote corrections. This is managed on a higher level. The most important task of the full-quote filtering window is *quote splitting*.

4.6.1 Quote Splitting Depending on the Instrument Type

Quotes can have complex structures as explained in Section 4.2.1. The filter follows the guideline of quote splitting, which is motivated by the goals of modularity and transparency. Instead of trying to formulate complex algorithms for complex data structures, we split the quotes into scalar quotes that are individually filtered, wherever possible. Some filtering operations are done on a higher level before splitting as explained in Section 4.7.

The quote splitting unit has the task of splitting the stream of full quotes into streams of different filtered variables, each with its scalar quotes that are used in the filtering operations of Section 4.4.

Some quotes, such as bid-ask or open/high/low/close quotes, are splittable by nature. Only the bid-ask case is discussed here as many instruments come in the form of bid-ask quotes. Other instruments have single-valued quotes. Bid-ask quotes are split into three scalar quotes:

- Bid quote
- Ask quote
- Bid-ask spread

Other instruments have single-valued quotes, which are "split" into one scalar quote:

- The "level" quote

This is not as trivial as it looks because quote splitting is coupled with two other operations: basic validity testing and mathematical transformations, as will be explained. The user of the filter has to know whether an instrument has single-valued or bid-ask quotes and has to select or configure the filter accordingly.

4.6.2 The Basic Validity Test

Many quotes have a natural lower limit in a predetermined *domain*. This instrument-dependent information has an impact on quote splitting and needs to be configured by the user. The lower limit of the allowed domain is called $p_{\min}$. For some instruments, there is no limit (or $p_{\min} = -\infty$). The choice of the lower limit is rather obvious for most instruments. A list of important examples is presented here:

Prices. Genuine asset prices of whatever kind, including FX and equity prices, are never negative. This means: $p_{\min} = 0$.

FX forward premiums/discounts. As explained in Section 2.3.2, the "forward points" can be positive or negative. There is no lower limit ($p_{\min} = -\infty$).

Interest rates. These can be *slightly* negative in extreme cases such as the JPY case discussed in Section 2.3.1 (but these negative interest rates were above -1%). Some theories rely on interest rates staying always positive, but a filter is not allowed to reject slightly negative interest rates if these are posted by reasonable contributors. The filter should use a moderately negative value of $p_{\min}$ here, e.g. -5%.

Short-term interest-rate futures. These can normally be handled as ordinary prices (where $p_{\min} = 0$), but are in fact defined by Equation 2.3 to have no lower limit (but an upper one). In practice, futures prices are quite far from 0, so it does not matter whether we assume a lower limit of 0 or none.

The choice of the lower limit is important for the further treatment. The following errors lead to complete invalidity:

- Quotes that violate the monotonic sequence of time stamps (i.e., quotes with a time stamp before the previously treated quote). In some software environments, this is an impossible error.
- A domain error. An illegal level p of the filtered variable (i.e., $p < p_{\min}$) as opposed to a merely implausible level.

Invalid scalar quotes with an error of this kind do not enter a scalar filtering window and are completely ignored in all further filtering steps. We mark them by setting $C_i = 0$. This is a fundamentally stronger statement than merely giving a very low credibility as a result of the scalar filtering window.

In the case of bid-ask quotes, the three resulting scalar quotes are tested individually:

- Bid quote. Domain error if bid quote $p_{\text{bid}} < p_{\min}$.
- Ask quote. Domain error if ask quote $p_{\text{ask}} < p_{\min}$.
- Bid-ask spread. Domain error if $p_{\text{ask}} < p_{\text{bid}}$.

Thus it is possible that the same quote leads to a valid bid quote passed to the scalar filtering window of bid prices and an invalid ask quote that is rejected.

The domain test of bid-ask spreads needs to be further discussed. First, we might interpret bad values ($p_{\text{ask}} < p_{\text{bid}}$) as the result of a sequence error. In other words, if the contributor typed ask-bid instead of bid-ask, this would be an error that could be corrected by the filter. This interpretation, although being true in many cases, is dangerous as a general rule. We prefer to reject all ask quotes that are less than the bid quote.

On the other hand, a more rigid test might also reject *zero* spreads. However, there are some quote contributors to minor markets interested only in either bid or ask or middle quotes. These contributors often produce formal quotes with $p_{\mathrm{bid}} = p_{\mathrm{ask}}$. In some markets, such quotes are the rule rather than the exception. A filter that rejects all of those quotes is throwing away some valuable and irreplaceable information.

The solution looks as follows. First, there is a filtering option of generally rejecting zero spreads (i.e., the case $p_{\mathrm{bid}} = p_{\mathrm{ask}}$). If the user chooses this option, the quote splitting algorithm will act accordingly. Otherwise, zero spreads can be accepted, but they have low credibilities in a market dominated by positive spreads. This is further explained in the next section.

4.6.3 Transforming the Filtered Variable

The filtered variable is mathematically transformed in order to reach two goals:

1. A simpler (e.g., more symmetric) density function. The basic filtering operations (e.g., Equation 4.6), assume a roughly symmetric distribution of the scalar quote values (and their changes). Some variables, mainly the bid-ask spread, have a skewed distribution. The filtering method contains no full-fledged analysis to determine the exact nature of the distribution. This would be too much for an efficient filter algorithm. The idea of the transformation is that the mathematically transformed variable has a more symmetric distribution than the raw form. For the logarithm of bid-ask spreads, this has been demonstrated in Müller and Sgier (1992).
2. The transformed variable should not depend on units such as Japanese Yens per U.S. Dollar. *Relative* changes have the advantage of being comparable between different financial instruments and different time periods of the same instrument. They do not depend on the units in which the different rates are expressed. A usual way to work with relative, unit-free variables is to take the logarithm of the raw variables such as prices.

The rules of the mathematical transformation are closely related to the validity tests of Section 4.6.2. The transformation never fails because all illegal quotes have already been removed by the domain tests. The transformed quote value is denoted by x and used in many formulas of Section 4.4.

For single-valued quotes, bid quotes, and ask quotes, the following transformation is made:

$$x = \begin{cases} \log(p - p_{\mathrm{min}}) & \text{if } p_{\mathrm{min}} > -\infty \text{ exists} \\ p & \text{otherwise} \end{cases} \tag{4.44}$$

For bid-ask spreads, the transformation is

$$x = x_{\mathrm{spread}} = 45.564 \sqrt{x_{\mathrm{ask}} - x_{\mathrm{bid}}} \tag{4.45}$$

where x_{bid} and x_{ask} are results from Equation 4.44. Equation 4.45 has been chosen to return a value similar to $\log(x_{ask}-x_{bid})+\text{constant}$ for a wide range of arguments $x_{ask} - x_{bid}$ of typically occurring sizes. Indeed, a logarithmic transformation of spread values would be a natural choice. The reason to use Equation 4.45 rather than a logarithmic transformation is related to zero spreads.[2] A logarithmic transformation would make zero spreads impossible (as $\log(0) = -\infty$). When inserting a zero spread in Equation 4.45, we obtain the legal result $x = 0$. This value is far away from typical ranges of values obtained for positive spreads, so its credibility is likely to be low in normal situations. When zero spreads become a usual event, the filter will start to accept them.

4.7 UNIVARIATE FILTERING

Univariate filtering is the top level of the filter. All the main filtering functions are managed here. The full-quote filtering window with its quote splitting algorithm of Section 4.6.1 is on a lower hierarchy level (see Table 4.1). Thus the univariate filter sees full quotes before they are split; it has access to all components of a full quote in their raw form (with no transformation).

The tasks of univariate filtering are as follows:

- Serving as the main configuration of a filter.
- Analyzing those data errors that affect not only individual quotes but a whole continuous sequence of quotes. The presence (or absence) of such a general error defines the *filtering hypothesis*. Two such cases were found in financial data and are therefore covered by the filter:

  1. Decimal errors. A wrong decimal digit of the quote, corresponding to a constant offset from the true quote.
  2. Scaling factor. The quote deviates from the true level by a constant factor, often a power of 10.

  Both cases are further discussed here.
- Creating a new full-quote filtering window for a newly detected filtering hypothesis.
- Managing filtering hypotheses and their full-quote filtering windows during their lifetimes, selecting the winning hypothesis.
- In the case of an error hypothesis, correcting the error of new incoming quotes according to the hypothesis and passing the corrected quotes to the full-quote filtering window.
- Packaging the filtering results to be accessed by the user.
- Recommending a suitable build-up period of the filter prior to the desired start date of the filtering result production, based on the filter configuration. Typical sizes are from weeks to months.

[2] The treatment of zero spreads is discussed at the end of Section 4.6.2.

The errors affecting a continuous sequence of quotes cannot be sufficiently filtered by the means described in the previous sections; they pose a special challenge to filtering. The danger is that the continuous stream of false quotes is accepted to be valid after a while because this false series appears *internally* consistent.

A filtering hypothesis is characterized by one general assumption on an error affecting all its quotes. This can lead to another unusual property. Sometimes the cause of the error is so clear and the size of the error so obvious that quotes can be *corrected*. In these cases, the filter produces not only credibilities and filtering reasons but also corrected quotes that can be used in further applications. This will discussed further.

The errors leading to a filter hypothesis are rare. Before discussing the details, we should evaluate the relevance of this filtering element in general. Such an evaluation may lead to the conclusion that the filtering hypothesis algorithm is not necessary in a new implementation of the filter.

Decimal errors have been the dominant error type in the page-based data feed from Reuters in 1987–1989. In later years, they have become rare; they hardly exist in modern data feeds. The few remaining decimal errors in the 1990s often were of short duration so they could successfully be filtered also through the standard data filter. Thus there is no convincing case for adding a decimal error filter algorithm to a filter of modern data. A decimal error filter is needed if old, historical data have to be cleaned.

The scaling filter is also superfluous if the user of the filter has a good organization of raw data. If a currency is rescaled (e.g., 1000 old units = 1 new unit as in the case of the Russian Ruble), a company with good data handling rules will not need the data cleaning filter to detect this; this rescaling will be appropriately handled before the data is passed to the filter. Rescaled currencies (or equity quotes after a stock split) can be treated as a *new* time series. However, the transition between the two definitions may not be abrupt, and there may be a mixture of quotes of both scaling types for a while. A scaling analysis within the filter can serve as an additional element of safety to treat this case and detect unexpected scale changes.

There is the possibility of having coexisting hypotheses, for example, the hypothesis of having a decimal error and the hypothesis of having none. If an immediate decision in favor of one hypothesis is always made, there is no need to store two coexisting hypotheses. Note that the filtering hypothesis algorithms are executed for each new quote before quote splitting.

4.7.1 The Results of Univariate Filtering

The output of the univariate filter consists of several parts. For every quote entered, the following filtering results are available:

1. The credibility of the quote
2. The value(s) of the quote, possibly corrected according to a filtering hypothesis such as a scaling factor or a decimal error as explained in 4.2.2

3. The filtering reason, explaining why the filter has rejected a quote
4. Individual credibilities of scalar quotes (bid, ask, spread)

Users may only want a minimum of results, perhaps just a yes/no decision on using or not using the quote. This can be obtained by simply checking whether the credibility of the quote is above or below a threshold value, which is usually chosen to be 0.5.

In the case of bid-ask data, the credibility C of the full quote has to be determined from the credibilities of the scalar quotes, usually applying the following formula:

$$C = \min(C_{\text{bid}}, C_{\text{ask}}, C_{\text{spread}}) \tag{4.46}$$

This formula is conservative and safe; valid quotes are meant to be valid in every respect. The timing of the univariate filtering output depends on whether it is in a historical or real-time mode.

4.7.2 Filtering in Historical and Real-Time Modes

The terms "historical" and "real-time" are defined from the perspective of filtering here. A filter in real-time mode may be applied in a historical test. The two modes differ in their timing:

- In the real-time mode, the credibilities of a newly included quote resulting from Equations 4.38 and 4.1 are immediately passed to the univariate filtering unit. If there is only one filtering hypothesis, these credibilities are directly accessible to the user. If there are several hypotheses, the hypothesis with the highest overall credibility will be chosen.
- In the case of historical filtering, the initially produced credibilities are modified by the advent of new quotes. Only those quotes are output whose credibilities are finally determined. At that time, the quotes leave the full-quote filtering window and this implies that their components have also left the corresponding scalar filtering windows. If several filtering hypotheses coexist, their full-quote windows do not dismiss any quotes and so we get filtering results only when conflicts between filtering hypotheses are finally resolved in favor of one winning hypothesis.

Although these modes are different, their implementation and selection is easy. In the historical mode, we retrieve the oldest member of the full-quote window only after a test on whether this oldest quote and its results are ready. In the real-time mode, we pick the newest member of the same full-quote window. Thus it is possible to get both modes from the same filter run.

A special option of historical filtering should be available by obtaining the last quotes and their results when the analysis reaches the most recent available quote. It should be possible to output the full-quote window (of the dominant filtering hypothesis) for that purpose, even if the credibilities of its newest quotes are not finally corrected.

This leads to another timing mode that might frequently occur in practice. A real-time filter might be started from historical data. In this case, we start the filter in historical mode, flush the full-quote window as soon as the filter time reaches real time, and then continue in real-time mode. This can be implemented as a special mode if such applications are likely.

4.7.3 Choosing the Filter Parameters

The filter algorithm as a whole depends on many configuration parameters. Table 4.5 summarizes the definitions and explanations. The parameters are listed in the sequence of their appearance in Chapter 4. Some less important parameters have no symbol and appear directly as numbers in the text; nevertheless they have been included in Table 4.5. The same parameter values can be chosen for the different financial markets. Tests have shown that we need no parameter adjustments because the adaptive algorithm successfully adjusts to different financial instruments.

Filter users may choose the parameter values in order to obtain a filter with properties suited to their needs. A higher value of ξ_0 in Equation 4.11, for instance, will lead to a more tolerant filter. For a sensitivity test, we define different filters, for example, a weak (tolerant) filter and a strong (fussy) filter. This is explained in Section 4.9.

4.8 SPECIAL FILTER ELEMENTS

The filter described so far is flexible enough for most cases, but not for some of the special error types presented at the end of Section 4.2.2. These errors can be identified by additional algorithmic elements, which are discussed by Müller (1999). Moreover, there can be disruptive events such as the redefinition of financial instruments that pose some additional problems. For these rare cases, the data cleaning environment should provide the possibility of human intervention.

4.8.1 Multivariate Filtering: Filtering Sparse Data

Multivariate filtering is a concept that has not been used in the empirical results of this book, and univariate filtering as described in Section 4.7 remains the highest algorithmic level. Multivariate filtering requires a more complex and less modular software than univariate filtering—but it seems the only way to filter very sparse time series with unreliable quotes. Some concepts of a possible implementation are presented here.

In the financial markets, there is a quite stable structure of only slowly varying correlations between financial instruments. In risk management software packages, a large, regularly updated covariance matrix is used to keep track of these correlations. Covariance matrices between financial instruments can also be applied in the data cleaning of sparse quotes. Although univariate filtering methods work well for dense quotes, they lose a large part of their power when the density

TABLE 4.5 List of filter parameters.

Description of parameter	Symbol	Equation number
Range of mean x	$\Delta\vartheta_r$	4.3, 4.4
Parameters of Δx^2_{min} used in the level filter		(after Equation 4.7)
Critical deviation from mean x	ξ_0	4.8
Critical size of value change	ξ_0	4.11
Interaction range in change filter (normal value, special value for bid-ask spread)	ν	4.13
Range of quote density	$\Delta\vartheta_r$	4.15
Weight of new quote in quote density (normal value, special value for repeated quotes)	c_d	4.15
Range of short-term, standard and long-term volatility (v_{fast}, v, v_{slow})	$\Delta\vartheta_r$	4.16
Relative time interval offset for volatility	d_0	4.17
Absolute time interval offset for volatility	$\delta\vartheta_{\text{min}}$	4.17
Relative limits of quote interval $\Delta\vartheta$ (upper, lower)		4.19
Weight of squared granule in volatility offset		4.21
Parameters used for volatility offset ε_0 for bid-ask spreads		(after Equation 4.21)
Range (memory) of the quote diversity analysis	R	4.22
All parameters of the impact of quote diversity		4.24
Activity of active periods, for ϑ_k		4.25
Activity of inactive periods, for ϑ_k		4.25
Range of short-term volatility used for ϑ	$\Delta\vartheta_{\text{smooth}}$	4.28
Range of the variance of volatility fluctuations used for ϑ	$\Delta\vartheta_r$	4.29
Weight of the level filter	c_{level}	4.32
Trust capital dilution factor (normal value, special value at initialization from scratch)	μ	4.34 – 4.36
Window size parameter	W	4.43
Critical credibility for statistics update (normal value, special value at initialization from scratch)	C_{crit}	(Section 4.5.5)
Lower limit of allowed domain (prices, FX forwards, interest rates)	p_{min}	4.44 (and Section 4.6.2)
Factor in transformation of bid-ask spreads		4.45
Standard credibility threshold for accepting a quote		(Section 4.7.1)

of quotes is low. When a new quote of a sparse series comes in, there are only few quotes to compare and these quotes can be quite old and thus not ideal for filtering. This is the place where some additional information from the covariance matrix becomes useful. This can technically be done in several ways.

The only method outlined here is the *artificial quote* method. If the sparse rate (e.g., in form of a middle price) is included in a covariance matrix that also covers some denser rates, we can generate some artificial quotes of the sparse series by exploiting the most recent quotes of the denser series and the covariance matrix. The expectation maximization (EM) algorithm of Morgan Guaranty (1996) is a method to produce such artificial quotes; there are also some alternative methods. Results are good if all the series included in the generation of artificial quotes are highly correlated or anticorrelated to the sparse series.

Artificial quotes may suffer from three uncertainties: (1) they have a stochastic error in the value because they are estimated, (2) there is an uncertainty in time due to asynchronicities in the quotes of the different financial instruments (Low *et al.*, 1996), and (3) only a part of the full quote is estimated from the covariance matrix (e.g., the middle price, whereas the bid-ask spread has to be coarsely estimated as an average of past values). Therefore, an additional rule may be helpful by using artificial quotes only if they are not too close to good quotes of the sparse series.

In some cases, we can simply use arbitrage conditions to construct an artificial quote, such as the triangular arbitrage of FX cross rates explained in Section 2.2.2. The following algorithmic steps are done in the artificial quote method:

- Define a basket of high-frequency time series which are fairly well correlated or anticorrelated to the sparse series.
- Generate artificial quotes from the correlation matrix and mix them with the normal quotes of the sparse series, thus reinforcing the power of the univariate filtering algorithm.
- Eliminate the artificial quotes from the *final* output of the filter (because a filter is not a gap-filler).

This algorithm has the advantage of leaving the univariate filtering algorithm almost unchanged. The multivariate element only enters in the technical form of additional quotes. Quotes are the usual input of univariate filtering.

4.9 BEHAVIOR AND EFFECTS OF THE DATA FILTER

Data cleaning is a necessity because unfiltered outliers would spoil almost any data application. However, there is a legitimate concern about unwanted side effects caused by data cleaning. Are too many ticks rejected? Does filtering open a door to arbitrary data manipulation?

The rejection rates are low as shown by the typical examples presented in Table 4.6. The investigated data filter is a standard filter developed and used by Olsen & Associates (O&A), following the guidelines of Chapter 4. A proper build-up time is essential for such an adaptive filter as explained in Section 4.3.1. In all

TABLE 4.6 Data cleaning: Rejection rates.

Percentage of ticks rejected by a standard data cleaning filter of Olsen & Associates, for different financial markets. The analyzed test samples always consist of irregularly spaced high-frequency data over a period of one year. The reported rejection rates originate from the filter working in real-time mode.

Market	Financial instrument	Analyzed time period	Number of all ticks in period	Rejected outlier ticks	All rejected ticks
Major FX rates	EUR-USD	Mar 99 – Feb 00	3,457,116	0.07%	0.30%
	USD-JPY	Jan 89 – Dec 89	683,555	0.24%	0.49%
	USD-JPY	Jan 99 – Dec 99	1,324,421	0.06%	0.48%
Minor FX rates	USD-MYR	Jan 99 – Dec 99	1,950	7.59%	8.41%
	USD-MXP	Jan 99 – Dec 99	55,227	1.14%	1.66%
Spot interest rates	GBP (3 months)	Jan 99 – Dec 99	10,471	0.08%	50.27%
Short-term interest rate futures	CHF (Mar 00, LIFFE)	Jan 99 – Dec 99	34,561	8.54%	8.54%

examples, the build-up period was the 3 months preceding the analyzed period. All the examined raw data have been collected from the Reuters real-time data feed. Two rejection rates are indicated: (1) the rejection rate of "classical" outliers only, and (2) the rate of all rejected ticks, including those monotonically drifting or excessively repeated ticks identified by special parts of the cleaning algorithm. These "nonclassical" data errors are explained in Section 4.2.2 and can directly or indirectly lead to bad data quality, as the normal outliers. Therefore, they are eliminated by a good data filter.

For frequently quoted, major financial instruments, less than 0.5% of the ticks are rejected, as indicated by examples of major FX rates (EUR-USD and USD-JPY) in Table 4.6. The two analyzed USD-JPY samples are separated in time by 10 years. The percentage of outliers has clearly decreased over these 10 years. Data quality seems to have improved. However, the percentage of *all* rejected ticks has remained almost stable, due to an increase of monotonically drifting and excessively repeated ticks. These bad ticks are generated by improper computerized quoting, which has obviously become more widespread over the years. Minor FX rates such as USD-MYR and USD-MXP in Table 4.6 typically have higher rejection rates, which may exceed 5%. In less liquid markets, the competitive pressure to publish high-quality data seems to be lower. The spot interest rate of GBP with a maturity of 3 months in Table 4.6 has the high rejection rate of 50%, but there are just 0.1% true outliers. The high number of 50% is solely due to the quoting habit of one single bank that excessively repeated few quotes at high frequency over long periods. This behavior is also found for other, similar financial instruments. Market data from exchanges are often more reliable because of the centralized data generation. The percentage of outliers is smaller, and there are no monotonically drifting or excessively repeated quotes. This latter observation can be made for the Swiss Franc (CHF) interest rate futures of Table 4.6, where all the rejected ticks are true outliers. However, the outlier rate is rather high, about 8.5%. A closer look shows that most of these rejected ticks are empty ticks with formally quoted values of zero. Whatever the reason of the data supplier to post these empty ticks, the filter rightly rejects them as outliers. The rejection rates of Table 4.6 have been computed for the filter running in real-time mode. The corresponding rejection rates of historical filtering (see Section 4.7.2) are similar—usually slightly lower. A data filter needs a testing environment to analyze its statistical behavior. Table 4.6 presents a simple example of results produced by such a testing environment.

A good general method to test the effects of filtering in practice is a *sensitivity* analysis of the following kind. The data application, whatever it is, is implemented twice, using two *different* filters. Both filters may follow the same algorithm, but one of them is weak with more tolerant parameters, leading to a lower rejection rate, perhaps only half the rejection rate of the other filter. Then the results of both applications are compared to each other. The deviations between analogous results directly reflect the sensitivity or robustness of the analysis against changes in the data cleaning algorithm, and indirectly the possible degree of distortion by the filter.

This has been done, for example, in the case of an extreme value study of FX returns—a type of analysis very sensitive to outliers (which naturally lead to extreme return observations). Fortunately, the results for both filters are very similar, which means that both filters successfully eliminate the true outliers. The doubtful ticks that are accepted by one filter and rejected by the other one have little influence on the final results.

5

BASIC STYLIZED FACTS

5.1 INTRODUCTION

Gathering basic stylized facts on the behavior of financial assets and their returns is an important research activity. Without such facts it is not possible to design models that can explain the data. High-frequency data opened up a whole new field of exploration and brought to light some behaviors that could not be observed at lower frequencies. In this chapter we review the main stylized facts for foreign exchange (FX) rates, interbank money market rates, and Eurofutures contracts.

These stylized facts can be grouped under four main headings: autocorrelation of return, distributional issues, scaling properties, and seasonality. We find a remarkable similarity between the different asset types. Hence, we shall examine each of the properties first for FX rates and then show how they are present or modified for the others. FX rates have been the subject of many studies. However, these studies do not present a unified framework of the return distributions of the data-generating process. Most of the earlier literature analyzed daily time series, but, more and more, recent publications deal with intraday prices. They essentially confirm the findings of this chapter. Here, we use a set of intraday time series covering a worldwide 24 hr market,[1] and we present a study of fundamental statistical

[1] For a full description of the data, we refer the reader to Chapter 2.

properties of the intraday data. More specifically, this chapter demonstrates the following:

- At the highest frequency, the middle price is subject to microstructure effects (e.g., the bouncing of prices between the bid and ask levels). The price formation process plays an important role and overshadows some of the properties encountered at lower frequencies.
- The distributions of returns are increasingly fat-tailed as data frequency increases (smaller interval sizes) and are hence distinctly unstable. The second moments of the distributions most probably exist while the fourth moments tend to diverge.
- Scaling laws describe mean absolute returns and mean squared returns as functions of their time intervals (varying from a few minutes to one or more years). We find that these quantities are proportional to a power of the interval size.
- There is evidence of seasonal heteroskedasticity in the form of distinct daily and weekly clusters of volatility. This effect may partly explain the fat-tailedness of the returns and should be taken into account in the study of the return distributions. Daily and weekly patterns also exist in quote frequency.
- Daily and weekly patterns are also found for the average bid-ask spread, which is negatively correlated to the volatility. The trading activity in terms of price quoting frequency has a positive correlation to the volatility and a negative one to the spread. These findings imply that the trading volume is also positively correlated to the volatility. The daily patterns of all these variables may be explained by the behavior of three main markets—America, Europe, and East Asia—whose active periods partially overlap. Our intraday and intraweek analysis shows that there are systematic variations of volatility, even within what are generally considered business hours.

The literature presents a number of views regarding the distributions of FX returns and the corresponding data-generating process. Some papers claim FX returns to be close to Paretian stable ones, for instance, (McFarland *et al.*, 1982; Westerfield, 1997); some to Student distributions that are not stable (Rogalski and Vinso, 1978; Boothe and Glassman, 1987); some reject any single distribution (Calderon-Rossel and Ben-Horim, 1982). Most researchers now agree that a better description of the data generating process is in the form of a conditional heteroskedastic model rather than being from an unconditional distribution. Among the earliest to propose this for the FX rates were Friedmann and Vandersteel (1982); Wasserfallen and Zimmermann (1985); Tucker and Scott (1987) and Diebold (1988). On distributional issues, the only agreement seems to be that daily returns are fat-tailed and that there are substantial deviations from a Gaussian random walk model. Moreover, all of the literature on GARCH agrees that the distribution is not stable. Many of the studies of the late 1980s have been limited to daily or even weekly data except for

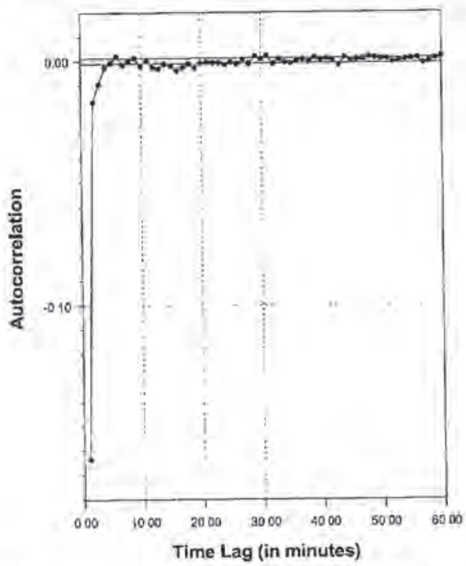


FIGURE 5.1 The autocorrelation function for the USD-DEM returns is plotted for different time lags in minutes up to 60 min. The returns are computed with prices interpolated using the previous tick interpolation method (see Chapter 3). The two horizontal lines represent the 95% confidence interval of an i.i.d. Gaussian process. The sampling period runs from January 5, 1987, to January 5, 1993. The autocorrelation is significantly negative up to a time lag of 4 min.

Wasserfallen and Zimmermann (1985); Feinstone (1987); Ito and Roley (1987), and Wasserfallen (1989). These papers analyze intradaily samples restricted to particular local markets and their local business hours. Recently the group of Barndorff-Nielsen has come up with the normal inverse Gaussian distribution that seems to capture some of the features we describe here, as observed in Eberlein *et al.* (1998); Barndorff-Nielsen (1998), and Barndorff-Nielsen and Prause (1999).

5.2 PRICE FORMATION PROCESS

The following three facts pertain to the short-term (less than 10 min) behavior of the foreign exchange intradaily returns. They highlight the difficulties inherent in tick-by-tick analysis.

5.2.1 Negative First-Order Autocorrelation of Returns

Goodhart (1989) and Goodhart and Figliuoli (1991) first reported the existence of negative first-order autocorrelation of returns at the highest frequencies, which disappears once the price formation process is over. In Figure 5.1, the autocorrelation function of returns measured at a 1 min interval is plotted against its lags. The returns are computed using the previous tick interpolation. There is significant autocorrelation up to a lag of 4 min. For longer lags, the autocorrelations mainly lie within the 95% confidence interval of an identical and independent (i.i.d.) Gaussian distribution. Goodhart (1989) also demonstrated that this negative autocorrelation is not affected by the presence (or absence) of major news announcements. Finally, Goodhart and Figliuoli (1992) showed that the resulting oscillations of the prices are not caused by bouncing prices between different geographical areas with different information sets. In Figure 5.1, negative autocorrelation is observed not only at the first lag (1 min) but also at further lags up to about 3 or 4 min. This is due to irregular spacing of ticks. If *tick time* is taken (i.e., an artificial time scale that moves by one unit with every tick), the negative autocorrelation is observed only at the first lag and rarely at larger lags, thus justifying the term "first-order." This behavior is characteristic if individual ticks randomly deviate from the market average while return clusters of longer duration are absent.

A first explanation of this fact is that traders have *diverging opinions* about the impact of news on the direction of prices—contrary to the conventional assumption that the FX market is composed of homogeneous traders who would share the same views about the effect of news so that no negative correlation of the returns would be observed. A second—and complementary—explanation for this negative autocorrelation is the tendency of market makers to skew the spread in a particular direction when they have order imbalances (Bollerslev and Domowitz, 1993; Flood, 1994). A third explanation is that even without order imbalances or diverging opinions on the price, certain banks systematically publish higher bid-ask spreads. This could also cause the prices to bounce back and forth between banks (Bollerslev and Domowitz, 1993). An early model for this bid/ask bounce was proposed by Roll (1984) in modeling transaction data in the stock market. The idea is that the two prices, bid and ask, can be hit randomly according to the number of buyers and sellers in the market. If the number of buyers is equal the number of sellers, which is the case most of the time in the market without exogeneous news, this model will produce a negative autocorrelation of transaction returns at high-frequency.

This negative autocorrelation is also seen in FX-rate transaction prices (Goodhart *et al.*, 1995) and in Eurofutures contracts (Ballocchi *et al.*, 1999b). For some stock indices such as the S&P 500, Bouchaud and Potters (2000) finds the autocorrelation of returns to be positive while it is not found in stock returns themselves or in futures contracts on indices (Ahn *et al.*, 2000). The explanation for the positive autocorrelation of stock indices is that some of them are constructed from equities that have very different liquidity. The model is called the *lagged adjustment model* (Ahn *et al.*, 2000). In this model one group of stocks reacts more slowly to aggregate information than another group of stocks. Because the autocovariance of a well-diversified portfolio is just the average cross-covariance of the stocks that make up the portfolio, this results in positive autocorrelations. In any case, the autocorrelation of returns is directly related to microstructure effects

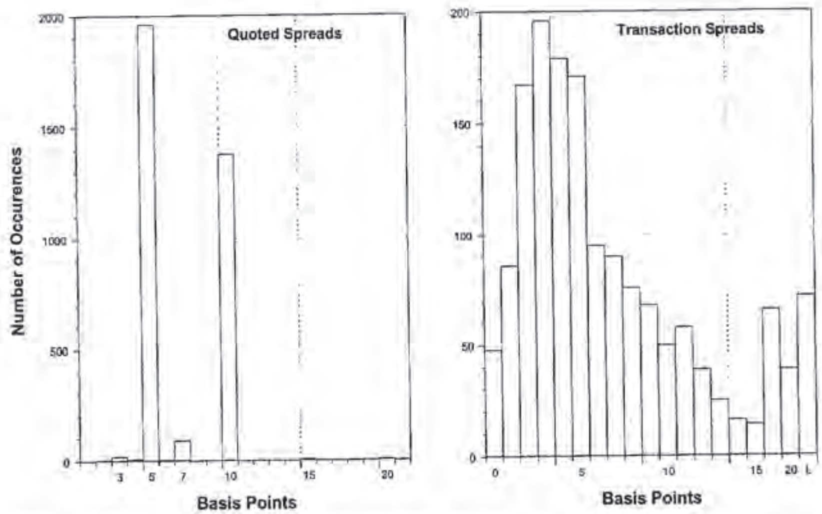


FIGURE 5.2 The figure on the left presents the spread size frequencies for USD-DEM *quotes* during June 16, 1993, collected from Reuters FXFX page. The figure on the right presents the spread size frequencies for USD-DEM *transactions* during June 16, 1993, from an analysis of Reuters Dealing 2000-2 by Goodhart *et al.* (1995).

in the market and should be carefully considered before using data at very high frequency.

The negative first-order autocorrelation can be seen as unwanted noise to be removed in a further study. An *effective price* can be defined in a way to eliminate the negative autocorrelation, as already discussed at the end of Section 3.2.2.

5.2.2 Discreteness of Quoted Spreads

Bid-ask spreads have discrete values. For studying this, we use the spread in its raw form, defined as ask price minus bid price, rather than the relative spread defined by Equation 3.12. In the example of Figure 5.2, bid-ask spreads of FX quotes are discretely distributed with the major peak at 5 basis points, followed by peaks at 10 and 7 basis points. A basis point is the smallest quoted decimal digit, which is 0.0001 German Marks per U.S. Dollar in the case of USD-DEM. In other, longer sampling periods and for other FX rates, we additionally observe spreads of 3, 8, 15, and 20 basis points with noticeable frequency. In a sample investigated by Bollerslev and Melvin (1994), the peaks at 5, 7, 10, and 15 basis points account for more than 97% of the distribution. These conventional spread values have evolved over the years, depending on the markets. For USD-DEM and some other major FX rates, the highest spread frequency peak shifted from 10 to 5 basis points during the 1990s, partly because the price levels became lower (Müller and Sgier, 1992). As explained in Section 3.2.5, spreads mainly depend on the cost structure of the market making banks and the habits of the market. Goodhart and Curcio (1991) have shown that individual banks usually quote two or three different spreads. Market makers who want to attract buyers more than sellers, or the other way around, tend to publish a skewed quote where only either the bid or the ask price is competitive and the other price is pushed away by a spread of conventional size, often 5 or 7 basis points. When they are uncertain about the direction the price should take, they may quote larger spreads with conventional values such as 10 or 15 (Lyons, 1998). Because different banks have different conventions and market situations change over time, the distribution of spreads has 4 or 5 peaks instead of 2 or 3.

A possible way to approximately *model* the *real* spreads, that is the difference between *traded* bid and ask prices, could be an analysis of the market *microstructure* which is discussed in detail by Flood (1991). Such a real spread model would analyze the microoscillations of (almost) simultaneous prices from different market makers. The *effective spread* would be something like the difference between the lowest ask and the highest bid prices currently quoted by any market maker—a model that would complement the effective price model proposed in Section 5.2.1. We did not try to set up such a subtle model, although we think that this would be the only way to overcome the limitations of quoted spreads. In a recent paper, Hasbrouck (1998) precisely proposes a market microstructure model for the clustering of the spreads based on a similar idea of a latent continuous efficient price.

Although the distribution of the spreads is discrete and consistent with theory (Admati and Pfleiderer, 1988; Subrahmanyam, 1991) market makers will cover themselves by conventional larger spreads in periods of higher risk such as in the release of important news (Goodhart, 1989), the closing or opening of markets (Bollerslev and Domowitz, 1993) and lunch breaks, (Müller *et al.*, 1990). More generally, the size of the spread is inversely related to market activity as measured by the tick frequency or the mean hourly volatility (Müller *et al.*, 1990). The size of the spread is directly related to the (instantaneous) volatility, which also measures the risk (Bollerslev and Domowitz, 1993).

In Figure 5.2, very different pictures emerge from the quoted spreads, which are only *indicative*, and the spreads as obtained from the electronic dealing system Reuters Dealing 2000-2. The different behaviors of the spread constitute the most pronounced difference between quoted prices and transaction prices. In Figure 5.2, the spread of actual transaction prices is uniformly distributed as one would expect. In their paper, Goodhart *et al.* (1995) note that, contrary to spreads, the volatility of middle prices does not exhibit substantial differences when transaction prices are used instead of quotes.

In the case of exchange-traded instruments such as Eurofutures (IR futures), there is no well-defined spread because the bid and the ask quotes are not synchronized and, depending on the market state, there may be only bid quotes or only ask quotes for a while. Nevertheless, a spread can be computed from bid and ask quotes that are few seconds apart. This effective spread is usually very small, typically less than one basis point on the Eurofutures market (Ballocchi *et al.*, 1999b), which represents relative spreads of the order of 10^{-4} according to the definition in Equation 3.12. Similar values are found for bond futures traded on the Deutsche Termin-Börse (DTB) (Franke and Hess, 1997).

5.2.3 Short-Term Triangular Arbitrage

The extremely short-term dynamics of price processes is also reflected in the significant predictive power of the USD-DEM in contrast to the other currencies (Goodhart and Figliuoli, 1991). A short delay is needed before traders in smaller currencies adjust themselves to the patterns of the two leading currencies. It is an effect comparable to the one we described in Section 5.2.1 for the positive autocorrelation of high-frequency returns of stock indices. Eben (1994) also finds evidence of triangular arbitrage opportunities at very high frequencies arising from very short-term trend reversals between two USD-rates, which are not yet reflected in the quoted cross rates. Although the detection of triangular arbitrage opportunities is rather easy and quick with a unique vehicle currency, it takes more time when the rates between two vehicles (e.g., USD and DEM) change (Suvanto, 1993; Hartmann, 1998).

Triangular arbitrage opportunities detected in quoted data do not necessarily reflect riskless profit-taking opportunities in real markets. The transaction costs may exceed the profits and the transaction prices may adjust more quickly than the quotes.

5.3 INSTITUTIONAL STRUCTURE AND EXOGENEOUS IMPACTS

5.3.1 Institutional Framework

An example of an institutional framework is the European Monetary System (EMS) introduced in the 1990s to keep some intra-European FX rates within certain bands. An intradaily analysis of FX rates within the EMS gives some insights into the distinct characteristics of this monetary system at a time when the bands were still quite narrow. As illustrated in Figure 5.3 (b), the EMS achieved a smaller *drift exponent* of the scaling law.

The scaling law relates the mean absolute return $E[|r|]$ observed over time intervals of a certain size to the size Δt of these intervals: $E[|r|] = \text{const}\,(\Delta t)^D$. The exponent D is called the drift exponent and empirically estimated using data samples. Low drift exponents indicate that the EMS successfully reduced the size of returns over large time intervals as compared to the volatility of short-term returns. A further, detailed discussion of drift exponents and their empirical estimation can be found in Section 5.5.

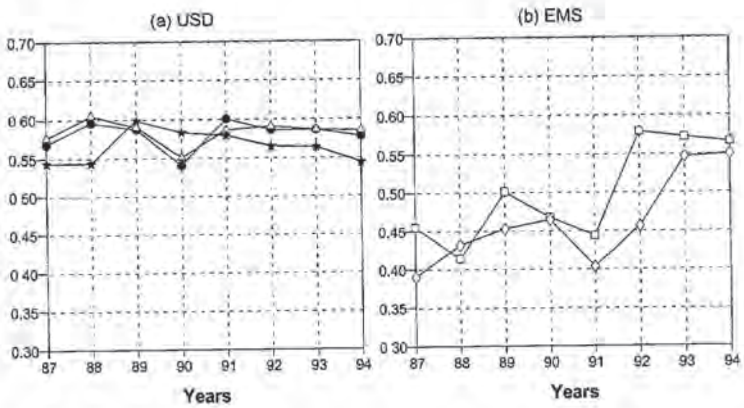


FIGURE 5.3 Drift exponents of the scaling law as a function of time, empirically estimated for yearly samples. (a): Drift exponents of freely floating rates against the USD, DEM (•), FRF (△), JPY (⋆). (b): Drift exponents of EMS rates against the DEM, ITL (□) and FRF (◇).

When the Italian Lira (ITL) left the EMS in 1992 and the EMS bands of the French Franc (FRF) were broadened in 1993, the values of the drift exponents went up and approached those of freely floating rates, as can be seen in Figure 5.3.

The drift exponent and the long-term volatility under the EMS were reduced, however, at the cost of a larger probability of extreme events. This is further explained in Section 5.4.2. The statistical analysis of EMS rates shows that institutional setups such as the EMS can be distinguished from freely floating markets by purely statistical criteria in a robust way, independent of assumptions on the generating-process.[2]

Another effect of the market framework can be seen in other financial high-frequency data such as interbank spot interest rates. In this market, money market quotes coming from East Asia are systematically higher than those from Europe or America. Figure 5.4 clearly indicates that for USD 3-month money market rates (spot interest rates collected from Telerate) the last bid quote before 2 a.m. GMT (Greenwich Mean Time) almost always exceeds the last bid quote before 8 p.m. GMT. On average, the early quote is larger by one-eight of a percent. The interest rate intraday seasonality is caused by a geographical market segmentation between East Asia on one side and Europe and America on the other side. This segmentation is justified by market practitioners as being due to institutional constraints and credit risks, making it less appealing on average for a European bank to place a deposit with an East Asian counterparty than with a European counterparty. The temporary difficulties in the Japanese banking sector were a likely cause of the segmentation. The segmentation became very pronounced in the last half of 1995 and again during the "Asian crisis" in 1998 (with interest rate deviations of about

[2] See Svensson (1992) for a review of the literature on the modeling of target zones, and in particular, of the European Monetary System.

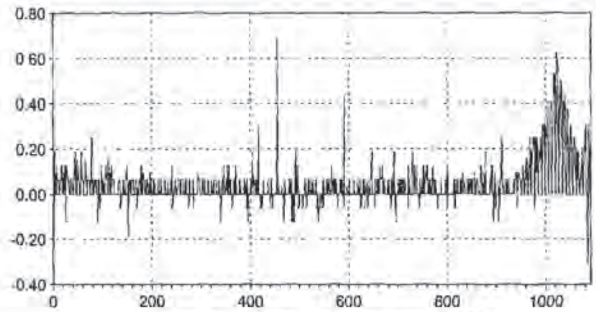


FIGURE 5.4 Daily difference between the last bid quote before 2 a.m. GMT and the last bid quote before 8 p.m. GMT. The early quote is systematically higher than the later one. Data sample: Quotes of the USD 3-month interbank money market rates published by Telerate. The analysis runs from January 5, 1993, to January 31, 1995.

0.5%). As explained in Section 2.3.1, the segmentation even caused negative JPY interest rates in the European and American markets.

5.3.2 Positive Impact of Official Interventions

One special type of trader is the central banks, as the time and the size of their interventions can be measured on an intradaily basis. Central banks may operate either directly through officially announced interventions or indirectly through unannounced interventions. Official interventions operate essentially as signals given to the markets and are therefore difficult to measure, see Edison (1993) for a review of the literature on central bank interventions. Some evidence is given in Goodhart and Hesse (1993) of the positive effects in the long run of official interventions, although they may result in short-term losses for the central bankers. One could, however, easily extend the analysis to any other long-term trader. A trader who can afford to keep a large open position for a long time will have some impact on the market through his reputation, even if he doesn't have a large share of the market. This is the case of some hedging funds, for example. Peiers (1997) shows the positive impact of unannounced interventions and interventions of a central bank, the Bundesbank, through the biggest player on the market, namely the Deutsche Bank.

5.3.3 Mixed Effect of News

News is a very broad concept covering a phone call of a customer who wants to make a large FX transaction (due to inventory imbalances, for instance), a conversation with a colleague, price forecasts and histories when used in technical analysis programs or the economic forecasts of the research department of a bank, general economic and political news, and major economic news announcements.

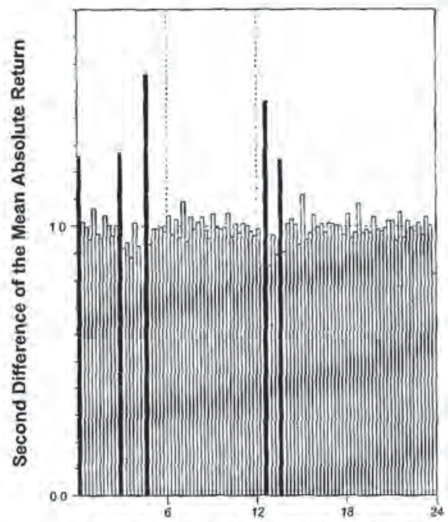


FIGURE 5.5 Intraday distribution of 15-min mean *changes* for the absolute returns (Equation 5.1 for the USD-DEM). Sudden peaks are darkened. The values are averages over all weeks of a sampling period of many years.

News is therefore difficult to quantify. Goodhart (1989) first tried to quantify news by looking at the "news" pages of Reuters. General economic and political news was displayed on the AAMM page (until February 1997). Goodhart (1989) found that "small" news does not have a significant effect on the behavior of the foreign exchange rates. Distinct and relatively large price movements unrelated to any news are indeed apparent. The price formation process seems to prevail, notwithstanding the presence or absence of news. In contrast, major economic news announcements such as trade, unemployment, budget deficit, or gross domestic product growth have significant impact (Goodhart, 1989). Economic news announcements along with the market expectations and the effect of the previous announcement were displayed in Reuters' FXNB page. Effective news—that is, the difference between the market expectation and the actual figure that is released—increases the volatility as the dispersion of traders' views on the impact of the effective news widens.

In Figure 5.5 we study the systematic effect of news release over a full day. We plot as a function of daytime (in GMT) a quantity that reflects the *change* of volatility by examining its relation to the neighboring values. This quantity is defined as

$$h_i = \exp\left[\ln|r_i| - \frac{1}{2}\ln|r_{i-1}r_{i+1}|\right] \tag{5.1}$$

where the index i represents the time of day (in steps of 15-*min*), and the h_i are averaged over all working days of a long sampling period. The three right peaks in Figure 5.5 show the clear-cut effect of news release in New York and Japan. News is not released every working day, but when it is, this happens at the typical daytimes indicated by peaks. The two peaks for the United States are separated by 1 hr and reflect the change of daylight saving time, which does not exist in Japan. The first two peaks on the left correspond to the beginning of the Japanese trading session and to the time just after the Japanese lunch. Goodhart *et al.* (1993) further show that major economic news announcements, such as release of the U.S. trade figures or changes in the U.K. base interest rate, have a significant impact on the return process. This effect however extends over 3 or 4 days as markets eventually incorporate the effects of the news. Moreover, the direction of the effect on the level of the price is difficult to predict. This can be explained by the highly nonlinear dynamics of the FX rates (Guillaume, 1994).

An alternative way to quantify the impact of news is with the mixture of distribution hypothesis (Clark, 1973; Tauchen and Pitts, 1983; Andersen, 1996). In this framework, the clustering of the volatility results from the clustering of the news arrival process. Because the news arrival process is an unobserved variable, proxies for the market activity such as the volume of trade are used (volume is not available in the FX markets). Moreover, as shown in Jones *et al.* (1994), volume can be rather noisy. Therefore, empirical studies in the FX intradaily markets use the tick frequency or the spread as proxies for the level of activity. Although a certain correlation between these variables and the volatility is obvious from the simple inspection of Figure 5.12, severe limitations harm the use of these variable as noted earlier. Moreover, Davé (1993) shows that tick frequency can only be a good approximation of the volume when markets are analyzed as separate geographical entities. Thus, there is no overlap between markets and the data are not disaggregated by individual bank subsidiary. Goodhart (1989) also shows that tick frequency does not specifically rise when news is released. Therefore, empirical evidence in favor of this mixture of distribution hypothesis is only partial, (Demos and Goodhart, 1992; Bollerslev and Domowitz, 1993). A more recent paper by Melvin and Yin (2000) provides new support for the link between news arrival frequency and quote frequency.

In another study, Almeida *et al.* (1998) are able to quantify the effect of news to a short-lived response of 15 min on average, confirming the results of Figure 5.5. The peaks of that figure disappear if longer time intervals are examined. This is also confirmed in a study by Franke and Hess (1998) on other very liquid markets such as the U.S. treasury bond market and the German Bund futures market. By studying the effect of scheduled U.S. macroeconomic news releases, these authors were able to detect an increase of volatility of the U.S. Treasury bond futures contracts. This anomalous volatility would last from few minutes to a maximum of an hour. Moreover, they show that the futures Bund price reacts significantly to an American news announcements. They attribute this reaction to the increasing integration of the German bond market. It is only with the use of more sophisticated indicators, rather than purely examining the returns, that it is possible to detect a significant impact of the news. Recently, Zumbach *et al.* (2000) have developed a scale of market shocks by integrating different volatility measures and relating a shock to its probability of occurrence. This measure is able to clearly identify turbulences on the market as well as to quantify the effect of news (Zumbach *et al.*, 2000).

5.4 DISTRIBUTIONAL PROPERTIES OF RETURNS

We mentioned in the introduction to this chapter the variety of opinions about the distributions of FX returns and the corresponding data-generating process. In this section, we do not want to propose a new model for the probability distribution function, but rather examine empirically what type of behavior is observed when returns are measured at different frequencies. We shall first present general results on the entire distribution and note that they are fat-tailed. Then, instead of looking at the center of the distribution, we shall present an alternative way to characterize the distribution by looking at the behavior of the tails.

There are many possible models of distribution functions, but this variety is greatly reduced when considering the tails of the distributions. The tail of a distribution can be described by using only one parameter, the tail index α. The empirical estimation of the tail index is difficult and requires large numbers of observations. The availability of high-frequency data makes this possible in practice. The methods, the empirical results, and their interpretation are presented in Section 5.4.2.

5.4.1 Finite Variance, Symmetry and Decreasing Fat-Tailedness

In this subsection, we analyze the probability distribution of returns of financial assets. The probability distribution associates each movement size with a certain probability of occurrence. In the case of empirical data, the domain of possible return values is divided into boxes, and one counts the frequency of occurrence in each box. One important issue in the case of tick-by-tick data is that this data are irregularly spaced in time, t_j. We have already discussed in Chapters 2 and 3 the different ways of constructing a homogeneous time series. Here we chose to take linearly interpolated prices. This is the appropriate method for interpolating in a series with independent random increments for most types of analyses. An alternative method discussed earlier, taking the last valid price before the gap as representative for the gap interval, must be avoided in a study of distributions as it would lead to a spurious large return from the last valid price within the gap to the first real price after the gap.

In Tables 5.1 and 5.2, we present the empirically computed moments of the distributions for the major FX rates against the USD and the major FX rates against the DEM.[3] The means are close to zero, as compared to the standard deviations,

[3] At least three of these cross rates have disappeared with the introduction of the Euro. Nevertheless, we think it is still interesting to report the results for them because they show the convergence of those

TABLE 5.1 Moments of return distributions for USD FX rates

This table gives an empirical estimation of the first 4 moments of the unconditional return distribution at different time intervals for the major currencies against the USD for the period from January 1, 1987, to December 31, 1993. The term kurtosis refers to the excess kurtosis, so a normal distribution has a kurtosis value of zero.

Rate	Time interval	Mean	Variance	Skewness	Kurtosis
USD-DEM	10 min	$-2.73 \cdot 10^{-7}$	$2.62 \cdot 10^{-7}$	0.17	35.10
	1 hr	$-1.63 \cdot 10^{-6}$	$1.45 \cdot 10^{-6}$	0.26	23.55
	6 hr	$-9.84 \cdot 10^{-6}$	$9.20 \cdot 10^{-6}$	0.24	9.44
	24 hr	$-4.00 \cdot 10^{-5}$	$3.81 \cdot 10^{-5}$	0.08	3.33
	1 week	$-2.97 \cdot 10^{-4}$	$2.64 \cdot 10^{-4}$	0.18	0.71
USD-JPY	10 min	$-9.42 \cdot 10^{-7}$	$2.27 \cdot 10^{-7}$	−0.18	26.40
	1 hr	$-5.67 \cdot 10^{-6}$	$1.27 \cdot 10^{-6}$	−0.09	25.16
	6 hr	$-3.40 \cdot 10^{-5}$	$7.63 \cdot 10^{-6}$	−0.05	11.65
	24 hr	$-1.37 \cdot 10^{-4}$	$3.07 \cdot 10^{-5}$	−0.15	4.81
	1 week	$-9.61 \cdot 10^{-4}$	$2.27 \cdot 10^{-4}$	−0.27	1.30
GBP-USD	10 min	$-6.91 \cdot 10^{-9}$	$2.38 \cdot 10^{-7}$	0.02	27.46
	1 hr	$7.61 \cdot 10^{-7}$	$1.40 \cdot 10^{-6}$	−0.23	21.53
	6 hr	$4.63 \cdot 10^{-6}$	$8.85 \cdot 10^{-6}$	−0.34	10.09
	24 hr	$1.72 \cdot 10^{-5}$	$3.60 \cdot 10^{-5}$	−0.26	4.41
	1 week	$6.99 \cdot 10^{-5}$	$2.72 \cdot 10^{-4}$	−0.66	2.77
USD-CHF	10 min	$-2.28 \cdot 10^{-7}$	$3.07 \cdot 10^{-7}$	−0.04	23.85
	1 hr	$-1.37 \cdot 10^{-6}$	$1.75 \cdot 10^{-6}$	0.05	18.28
	6 hr	$-8.23 \cdot 10^{-6}$	$1.11 \cdot 10^{-5}$	0.05	7.73
	24 hr	$-3.38 \cdot 10^{-5}$	$4.51 \cdot 10^{-5}$	−0.04	2.81
	1 week	$-2.58 \cdot 10^{-4}$	$3.16 \cdot 10^{-4}$	0.09	0.34
USD-FRF	10 min	$-1.98 \cdot 10^{-7}$	$2.08 \cdot 10^{-7}$	0.35	43.31
	1 hr	$-1.18 \cdot 10^{-6}$	$1.28 \cdot 10^{-6}$	0.47	28.35
	6 hr	$-7.13 \cdot 10^{-6}$	$8.29 \cdot 10^{-6}$	0.23	9.69
	24 hr	$-2.91 \cdot 10^{-5}$	$3.40 \cdot 10^{-5}$	0.06	3.22
	1 week	$-2.32 \cdot 10^{-4}$	$2.44 \cdot 10^{-4}$	0.16	0.88

and the absolute values of the skewness are, except in very few cases, significantly smaller than 1. We can conclude from these facts that the empirical distribution is almost symmetric. The mean values are slightly negative (except for GBP-USD where the currencies are inverted) because during this period (from January 1, 1987, to December 31, 1993) we have experienced an overall decline of the USD. For all time horizons, the empirically determined (excess) kurtosis exceeds the value 0, which is the theoretical value for a Gaussian distribution. For the shortest

currencies to the Euro by exhibiting lower variances than the others. They present a good example of the influence of external factors on the statistical behavior of financial asset prices.

TABLE 5.2 Moments of return distributions for DEM FX rates

This table gives an empirical estimation of the first 4 moments of the unconditional return distribution at different time intervals for the major currencies against the DEM for the period from January 1, 1987, to December 31, 1993. The term kurtosis refers to the excess kurtosis, so a normal distribution has a kurtosis value of zero.

Rate	Time interval	Mean	Variance	Skewness	Kurtosis
DEM-FRF	10 min	$9.84 \cdot 10^{-8}$	$1.91 \cdot 10^{-8}$	0.54	86.29
	1 hr	$5.89 \cdot 10^{-7}$	$1.14 \cdot 10^{-7}$	0.79	69.70
	6 hr	$3.53 \cdot 10^{-6}$	$6.53 \cdot 10^{-7}$	1.41	36.87
	24 hr	$1.07 \cdot 10^{-5}$	$2.84 \cdot 10^{-6}$	1.15	24.26
	1 week	$8.94 \cdot 10^{-5}$	$1.93 \cdot 10^{-6}$	1.92	3.95
DEM-NLG	10 min	$-5.19 \cdot 10^{-8}$	$1.42 \cdot 10^{-9}$	−5.68	9640.85
	1 hr	$-3.11 \cdot 10^{-7}$	$7.54 \cdot 10^{-9}$	2.76	4248.12
	6 hr	$-1.86 \cdot 10^{-6}$	$2.48 \cdot 10^{-8}$	0.74	124.35
	24 hr	$-7.80 \cdot 10^{-6}$	$9.66 \cdot 10^{-8}$	−0.30	30.02
	1 week	$-4.57 \cdot 10^{-5}$	$6.63 \cdot 10^{-7}$	0.03	0.06
DEM-ITL	10 min	$1.07 \cdot 10^{-6}$	$1.75 \cdot 10^{-7}$	0.86	64.03
	1 hr	$6.46 \cdot 10^{-6}$	$1.24 \cdot 10^{-6}$	1.83	89.92
	6 hr	$3.88 \cdot 10^{-5}$	$7.16 \cdot 10^{-6}$	1.03	37.26
	24 hr	$1.18 \cdot 10^{-4}$	$2.53 \cdot 10^{-5}$	−0.51	13.08
	1 week	$9.42 \cdot 10^{-4}$	$1.37 \cdot 10^{-4}$	−0.25	0.17
GBP-DEM	10 min	$4.53 \cdot 10^{-7}$	$9.86 \cdot 10^{-8}$	−0.32	25.97
	1 hr	$2.69 \cdot 10^{-6}$	$7.12 \cdot 10^{-7}$	−0.34	16.90
	6 hr	$1.56 \cdot 10^{-5}$	$4.62 \cdot 10^{-6}$	−0.02	7.48
	24 hr	$7.04 \cdot 10^{-5}$	$1.79 \cdot 10^{-5}$	0.27	3.15
	1 week	$1.17 \cdot 10^{-4}$	$1.29 \cdot 10^{-4}$	0.07	0.59
DEM-JPY	10 min	$-3.39 \cdot 10^{-6}$	$2.21 \cdot 10^{-7}$	−0.09	12.35
	1 hr	$-2.03 \cdot 10^{-5}$	$1.46 \cdot 10^{-6}$	−0.03	88.58
	6 hr	$-1.21 \cdot 10^{-4}$	$9.12 \cdot 10^{-6}$	−0.04	6.57
	24 hr	$-4.85 \cdot 10^{-4}$	$3.56 \cdot 10^{-5}$	0.12	2.52
	1 week	$-3.15 \cdot 10^{-3}$	$2.67 \cdot 10^{-4}$	−0.07	0.03

time intervals, the kurtosis values are extremely high. Another interesting feature is that all of the rates show the same general behavior, a decreasing kurtosis with increasing time intervals. At intervals of around 1 week, the kurtosis is rather close to the Gaussian value.

Tables 5.1 and 5.2 suggest that the variance and the third moment are finite in the large-sample limit and that the fourth moment may not be finite. Some solid evidence in favor of these hypotheses is added by the tail index studies that follow, mainly the results of Table 5.3. Indeed, the larger the number of observations, the larger the empirically computed kurtosis. At frequencies higher than 10 min, there seems to be some contradiction between the work of Goodhart and Figliuoli (1991), which claims that the fat tails start to decrease at these frequencies, and the paper of Bollerslev and Domowitz (1993), which gives some evidence of a still increasing fat-tailedness. One can show, however, that both results hold depending on whether one uses the linear interpolation method or the previous tick to obtain price values at fixed time intervals at such frequencies. This is an example of the difficulty of making reliable analyses of quoted prices at frequencies higher than 10 min. The divergence of the fourth moment explains why absolute values of the returns are often found to be the best choice of a definition of the volatility (i.e., the one that exhibits the strongest structures).[4] Indeed, because the fourth moment of the distribution enters the computation of the autocorrelation function of the variance, the autocorrelation values will systematically decrease with a growing number of observations.

To complement Tables 5.1 and 5.2, we plot on Figure 5.6 the cumulative frequency of USD-JPY for returns measured at 10 min, 1 day, and 1 week on the scale of the cumulative Gaussian probability distribution. Normal distributions have the form of a straight line, which is approximately the case for the weekly returns with a moderate (excess) kurtosis of approximately 1.3. The distribution of 10-min returns, however, has a distinctly fat-tailed form and its kurtosis in Table 5.1 is very high. If the data-generating process was a random walk with increments from a stable distribution, which is defined by the law that *scaled* returns $r/(\Delta t)^{\gamma}$ for a certain γ have the same distribution irrespective of the measurement interval Δt, we would obtain a uniform distribution with identical moments within the significance limits.[5] Considering all the presented results, this is clearly not the case. This instability of distributions was also found by other authors. McFarland *et al.* (1982) and Boothe and Glassman (1987) suggest that distributions are composed of reactions to different flows of information. Calderon-Rossel and Ben-Horim (1982) are in agreement with our findings and claim that the returns cannot be accurately described by a unique type of stable distribution.

5.4.2 The Tail Index of Return Distributions

The tails of all possible distributions can be classified into three categories:[6]

i. Thin-tailed distributions for which all moments are finite and whose cumulative distribution function declines exponentially in the tails

ii. Fat-tailed distributions whose cumulative distribution function declines with a power in the tails

iii. Bounded distributions which have no tails

[4] We shall see some evidence of this in Section 5.6.1 and in Chapter 7.

[5] Here there is no need to further characterize stable distributions in addition to the described scaling behavior. Section 5.5.2 has a definition and discussion of stable distributions.

[6] The interested reader will find the full development of the theory in Leadbetter *et al.* (1983), and de Haan (1990).

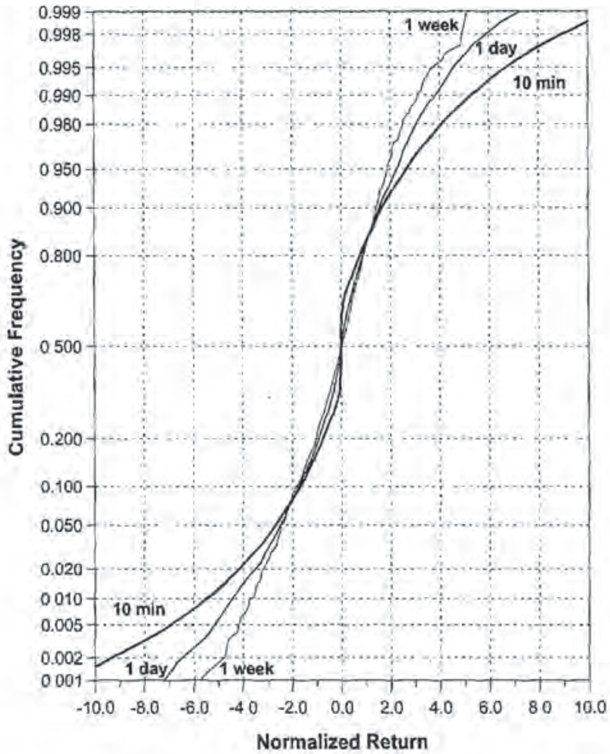


FIGURE 5.6 The cumulative distributions for 10-min, 1-day, and 1-week USD-JPY returns shown against the Gaussian probability on the y-axis. On the x-axis the returns normalized to their mean absolute value are shown. The mean absolute return for 10 min is 2.62×10^{-4}, for 1 day 3.76×10^{-3}, and for 1 week 1.14×10^{-2}. The three curves are S-shaped as typical of fat-tailed distributions. The S-shapes of the three curves are very differently pronounced.

A nice result is that these categories can be distinguished by the use of only one parameter, the tail index α with $\alpha = \infty$ for distributions of category (i), $\alpha > 0$ for category (ii), and $\alpha < 0$ for category (iii). The empirical estimation of the tail index and its variance crucially depends on the size of the sample (Pictet *et al.*, 1998). Only a well chosen set of the most extreme observations should be used. The very large sample size available for intradaily data ensures that enough "tail observations" are present in the sample. An important result is that the tails of a fat-tailed distribution are invariant under addition although the distribution as a whole may vary according to temporal aggregation (Feller, 1971). That is, if weekly returns are Student-t identically and independently distributed, then monthly returns are not Student-t distributed.[7] Yet the tails of the monthly return

[7] This is an implication of the central limit theorem.

distribution are like the tails of the weekly returns, with the same exponent α, but the *real tail* might be very far out and not even seen in data samples of limited size.[8] Another important result in the case of fat-tailed distributions concerns the finiteness of the moments of the distribution. From

$$\mathrm{E}\left[X^k\right] = M_0 + c\int_s^{\infty} x^{k-\alpha-1}dx + o\left(x^{k-\alpha}\right) \qquad (5.2)$$

where X is the observed variable, M_0 is the part of the moment due to the center of the distribution (up to s), c is a scale variable and α is the tail index. It is easily seen that only the first k-moments, $k < \alpha$, are bounded.

How heavy are the tails of financial asset returns? The answer to this question is not only the key to evaluating risk in financial markets but also to accurately modeling the process of price formation. Evidence of heavy tails presence in financial asset return distributions is plentiful (Koedijk *et al.*, 1990; Hols and De Vries, 1991; Loretan and Phillips, 1994; Ghose and Kroner, 1995; Müller *et al.*, 1998) ever since the seminal work of Mandelbrot on cotton prices (Mandelbrot, 1963). He advanced the hypothesis of a stable distribution on the basis of an observed invariance of the return distribution across different frequencies and the apparent heavy tails of the distribution. A controversy has long been going on in the financial research community as to whether the second moment of the distribution of returns converges. This question is central to many models in finance, which heavily rely on the finiteness of the variance of returns. The risk in financial markets has often been associated with the variance of returns since portfolio theory was developed. From option pricing models (Black and Scholes, 1973) to the Sharpe ratio (Sharpe, 1994) used for measuring portfolio performance, the volatility variable is always present.

Another important motivation of this study is the need to evaluate extreme risks in financial markets. Recently, the problem of risk in these markets has become topical following few unexpected big losses like in the case of Barings or Daiwa. The Bank for International Settlements has set rules to be followed by banks to control their risks, but most of the current models for assessing risks are based on the assumption that financial assets are distributed according to a normal distribution. In the Gaussian model the evaluation of extreme risks is directly related to the variance, but in the case of fat-tailed distributions this is no longer the case.

Computing the tail index is a demanding task but with the help of high frequency data it is possible to achieve reasonable accuracy (see Pictet *et al.*, 1998; Dacorogna *et al.*, 2001a), where a theorem is proved which explicitly shows that more data improve the estimation of the tail index. Here, we present the main framework[9].

[8] See, for instance, the simulations done in Pictet *et al.* (1998) where for a high enough aggregation level, it is not possible to recover the theoretical tail index for Student-t distributions even if one can use 128 years of 10-min data.

[9] The interested reader can find the details in two recent papers by Pictet *et al.* (1998) and Dacorogna *et al.* (2001a).

Let $X_1, X_2, \ldots, X_n$ be a sequence of n observations drawn from a stationary i.i.d. process whose probability distribution function F is unknown. We assume that the distribution is fat-tailed—that is, the tail index α is finite.[10] Let us define $X_{(1)} \geq X_{(2)} \geq \ldots \geq X_{(n)}$ as the descending order statistics from $X_1, X_2, \ldots, X_n$.

Extreme value theory states that the extreme value distribution of the ordered data must belong to one of just three possible general families, regardless of the original distribution function F (Leadbetter *et al.*, 1983). Besides, if the original distribution is fat-tailed, there is only one general family it can belong to

$$G(x) = \begin{cases} 0 & x \leq 0 \\ \exp(-x^{-\alpha}) & x > 0, \quad \alpha > 0 \end{cases} \tag{5.3}$$

where $G(x)$ is the probability that $X_{(1)}$ exceeds x. There is only one parameter to estimate, α, which is called the tail index. The stable distributions (excluding the Gaussian distribution), the Student-t model, and the unconditional distribution of the ARCH-process all fall in the domain of attraction of this type of distribution.

To give more intuition to these statements, we plot the logarithm of the order statistics m as a function of the difference between the logarithms of the most extreme observation, $\ln X_{(1)}$, and the m^{th} observation in the ordered sequence, $\ln X_{(m)}$. Such a plot is shown on Figure 5.7 for the case of a Student-t distribution with 4 degrees of freedom. Because we are in the domain of attraction of $\exp(-x^{-\alpha})$, it is trivial to see that the problem of estimating α becomes the problem of estimating the slope of the tangent at $m \to 0$ of the curve shown in Figure 5.7. We see that a straight line with a slope equal to 4 is indeed a good tangent to the curve, as it should be because the theoretical tail index of the Student-t distribution is equal to the number of degrees of freedom. Although the behavior of $\ln X_{(m)}$ is quite regular on Figure 5.7 because we took the average values over 10 Monte Carlo simulations, it is not always so and the problem of how to choose the number of points that are far in the tail is not trivial. One needs a more formal way to estimate the tail index. The estimator we present here is a way of estimating the slope of the tangent shown in Figure 5.7. There are other ways of studying the tail index by directly fitting the distribution of ordered data to some known distribution.[11]

We concentrate our efforts on the estimator first proposed by Hill (1975)

$$\hat{\gamma}^{H}_{n,m} = \frac{1}{m-1} \sum_{i=1}^{m-1} \ln X_{(i)} - \ln X_{(m)} \quad \text{where } m > 1 \tag{5.4}$$

This estimator was proven to be a consistent estimator of $\gamma = 1/\alpha$ for fat-tailed distributions in Mason (1982). From Hall (1982) and Goldie and Smith (1987), it follows that $(\gamma_{n,m} - \gamma)m^{1/2}$ is asymptotically normally distributed with mean zero

[10] A good review of the definitions used in this chapter can be found in Leadbetter *et al.* (1983).

[11] A good reference to learn about these methods is the book by Embrechts *et al.* (1997).

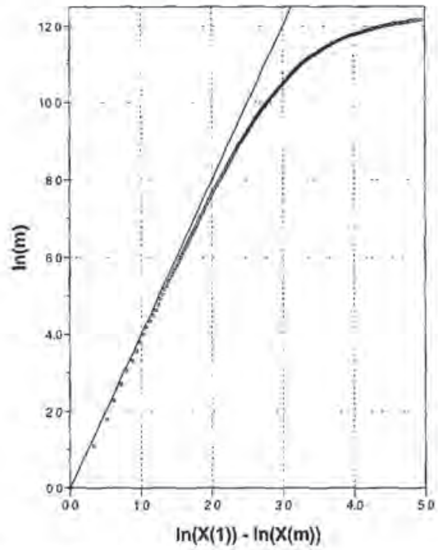


FIGURE 5.7 The logarithm of the order statistics m is plotted as a function of the difference between the logarithm of the most extreme observation and the logarithm of the ordered random observations. The data are drawn from a Student-t distribution with 4 degrees of freedom averaged over 10 replications of a Monte-Carlo simulation. The straight line represents the theoretical tangent to this curve.

and variance γ^2. In fact the Hill estimator is the maximum likelihood estimator of γ and $\alpha = 1/\gamma$ holds for the tail index. For finite samples, however, the expected value of the Hill estimator is biased. As long as this bias is unknown, the practical application of the Hill estimator to empirical samples is difficult. A related problem is that $\hat{\gamma}_{n,m}$ depends on m, the number of order statistics, and there is no easy way to determine which is the best value of m. Extending a bootstrap estimation method proposed by Hall (1990), Danielsson *et al.* (1997) solved the problem by means of a subsample bootstrap procedure, which is described and discussed by Pictet *et al.* (1998). Many independent subsamples (or resamples) are drawn from the full sample and their tail behaviors are statistically analyzed, which leads to the best choice of m. For such a statistical analysis, the subsamples have to be distinctly smaller than the full sample. On the other hand, the subsamples should still be large enough to contain some representative tail observations, so the method greatly benefits from a large sample size to begin with.

Tail index values of some FX rates have been estimated by a subsample bootstrap method and are presented in Table 5.3. The confidence ranges indicated for all values are standard errors times 1.96. Assuming a normally distributed error, this corresponds to 95% confidence. The standard errors have been obtained through the jackknife method, which can be characterized as follows. The data

TABLE 5.3 Estimated tail indices of FX rates

Estimated tail index values α and their 95% confidence ranges, for main FX rates against the USD, gold (XAU), and silver (XAG) and some of the main (computed) cross rates against the DEM, Müller *et al.* (1998). The tail index values are based on the subsample bootstrap method using the Hill estimator, the confidence ranges result from the jackknife method. Computed cross rates are obtained via the two bilateral rates against the USD, see Equation 2.1. The estimations are performed on samples from January 1, 1987, to June 30, 1996. The time intervals are measured in ϑ-time (see Chapter 6).

Rate	30 min	1 hr	2 hr	6 hr	1 day
USD-DEM	3.18 ±0.42	3.24 ±0.57	3.57 ±0.90	4.19 ±1.82	5.70 ±4.39
USD-JPY	3.19 ±0.48	3.65 ±0.79	3.80 ±1.08	4.40 ±2.13	4.42 ±2.98
GBP-USD	3.58 ±0.53	3.55 ±0.65	3.72 ±1.00	4.58 ±2.34	5.23 ±3.77
USD-CHF	3.46 ±0.49	3.67 ±0.77	3.70 ±1.09	4.13 ±1.77	5.65 ±4.21
USD-FRF	3.43 ±0.52	3.67 ±0.84	3.54 ±0.97	4.27 ±1.94	5.60 ±4.25
USD-ITL	3.36 ±0.45	3.08 ±0.49	3.27 ±0.79	3.57 ±1.35	4.18 ±2.44
USD-NLG	3.55 ±0.57	3.43 ±0.62	3.36 ±0.92	4.34 ±1.95	6.29 ±4.96
DEM-JPY	3.84 ±0.59	3.69 ±0.87	4.28 ±1.49	4.15 ±2.20	5.33 ±3.74
GBP-DEM	3.33 ±0.46	3.67 ±0.70	3.76 ±1.17	3.73 ±1.59	3.66 ±1.70
GBP-JPY	3.59 ±0.63	3.44 ±0.70	4.15 ±1.32	4.35 ±2.27	5.44 ±4.12
DEM-CHF	3.54 ±0.54	3.28 ±0.54	3.44 ±0.82	4.29 ±1.84	4.21 ±2.43
GBP-FRF	3.19 ±0.46	3.33 ±0.62	3.37 ±0.90	3.41 ±1.27	3.34 ±1.65
XAU-USD	4.47 ±1.15	3.96 ±1.13	4.36 ±1.82	4.13 ±2.22	4.40 ±2.98
XAG-USD	5.37 ±1.55	4.73 ±1.93	3.70 ±1.52	3.45 ±1.35	3.46 ±1.97

sample is modified in 10 different ways, each time removing one-tenth of the total sample. The tail index is separately computed for each of the 10 modified samples. An analysis of the deviations between the 10 results yields an estimate of the standard error, which is realistic because it is based on the data rather than restrictive theoretical assumptions. The methodology is explained by Pictet *et al.* (1998).

All the FX rates against the USD as well as the presented cross rates have a tail index between 3.1 and 3.9 (roughly around 3.5). These values are found in the 30-minute column of Table 5.3. The other columns are affected by lower numbers of observations and thus wider confidence ranges. The chosen cross rates are computed from USD rates according to Equation 2.1. None of them was part of the European Monetary System (EMS), except GBP-DEM for a period much shorter than the analyzed sample. Gold (XAU-USD) and silver (XAG-USD) have higher tail index values above 4. These markets differ from FX. Their volatilities were very high in the 1980s, followed by a much calmer behavior in the 1990s, a structural change that may have affected the tail statistics.

TABLE 5.4 Estimated tail indices of cross rates.

Estimated tail index values and their 95% confidence ranges, for cross FX rates. The tail index values are based on the subsample bootstrap method using the Hill estimator, and the confidence ranges result from the jackknife method. All the cross rates of the lower part were subject to the regulations of the European Monetary System (EMS). The computed cross rates are obtained via the two bilateral rates against the USD (see Equation 2.1). The estimations are performed on samples from January 1988 to June 1994 ($6\frac{1}{2}$ years). The time intervals are measured in ϑ-time (see Chapter 6).

Rate	30 min	1 hr	2 hr	6 hr
DEM-JPY	4.17 ±1.13	4.22 ±1.48	5.06 ±1.40	4.73 ±2.19
GBP-DEM	3.63 ±0.46	4.09 ±1.98	4.78 ±1.60	3.22 ±0.72
GBP-JPY	3.93 ±1.16	4.48 ±1.20	4.67 ±1.94	5.60 ±2.56
DEM-CHF	3.76 ±0.79	3.64 ±0.71	4.02 ±1.52	6.02 ±2.91
GBP-FRF	3.30 ±0.41	3.42 ±0.97	3.80 ±1.34	3.48 ±1.75
FRF-DEM	2.56 ±0.34	2.41 ±0.14	2.36 ±0.27	3.66 ±1.17
DEM-ITL	2.93 ±1.01	2.60 ±0.66	3.17 ±1.28	2.76 ±1.49
DEM-NLG	2.45 ±0.20	2.19 ±0.13	3.14 ±0.95	3.24 ±0.87
FRF-ITL	2.89 ±0.34	2.73 ±0.49	2.56 ±0.41	2.34 ±0.66

In Section 5.3.1, we have seen that cross rates behaved differently when both exchanged currencies were members of the EMS in the 1990s. A difference is also found when considering the tail behavior, as shown in Table 5.4. The sample was chosen accordingly, during the lifetime of the EMS. The upper block of Table 5.4 has non-EMS cross rates for comparison, the lower block has EMS cross rates. The 1-day column is missing in Table 5.4 as the sample size is smaller than that of Table 5.3. The tail index values of EMS cross rates are around 2.7, distinctly lower than the typical value of 3.5 found for other cross rates in the upper part of Table 5.4 and other FX rates in Table 5.3. The 30-min columns of both tables should mainly be considered, but the values of the other columns confirm the same fact that EMS cross rates have fatter tails.

The cross rates are computed from USD rates according to Equation 2.1. As compared to direct quotes, these computed cross rates have larger spreads and an artificially generated volatility (i.e., noise due to the price uncertainty within the spread and asynchronous fluctuations of the used USD rates). Therefore we have also analyzed direct cross rate quotes in the limited sample (since October 1992) where they are available. These direct quotes have less short-term noise and lead to slightly but systematically lower tail index values than those of computed cross rates. The small tail index of EMS cross rates indicates that the reduced volatility induced by the EMS setup is at the cost of a larger probability of extreme events, which may lead to realignments of the system. This is an argument against the credibility of institutional setups such as the EMS.

Like the drift exponent as a function of time, the tail index reflects the institutional setup and, in a further interpretation, the way different agents on the markets interact. The tail index can therefore be considered as another empirical measure of market regulation and market efficiency. A large tail index indicates free interactions between agents with different time horizons, with a low degree of regulation and thus a smooth adjustment to external shocks. A small tail index indicates the opposite.

With very few exceptions, the estimated tail indices are between 2 and 4. A few confidence ranges extend to values outside this range, but this is due to the limited number of observations mainly for the longer return measurement intervals. Using Equation 5.2, we conclude that the second moment of the return distribution is finite and the fourth moment usually diverges. In Section 5.6.1, this fact leads us to preferring the autocorrelation of absolute returns to that of squared returns, which relies on the finiteness of the fourth moment.

Tables 5.3 and 5.4 indicate that the distribution of FX returns belongs to the class of fat-tailed nonstable distributions that have a finite tail index larger than 2. Furthermore, the very high values of the kurtosis in Tables 5.1 and 5.2 and the growth of these values with increasing sample size provide additional evidence in favor of this hypothesis.[12] From Tables 5.3 and 5.4, one can also verify the invariance of the tail index under aggregation, except for the longest intervals, where the small number of observations becomes a problem in getting significant estimates of α. The smaller number of data for large intervals forces the estimation algorithm to use a larger fraction of this data, closer to the center of the distribution. Thus the empirically measured tail properties become distorted by properties of the center of the distribution, which, for $\alpha > 2$ and under aggregation, approaches the normal distribution (with $\alpha = \infty$) as a consequence of the central limit theorem.

In Table 5.5, we present the results of α estimations of interbank money market cash interest rates for five different currencies and two maturities. Although generally exhibiting lower α's, the results are close to those of the FX rates. The message seems to be the same: fat tails, finite second moment,[13] and nonconverging fourth moment. . We also find a relative stability of the tail index with time aggregation. The estimations for daily returns give more consistent values than in the case of the FX rates. Yet the estimations are more noisy as one would expect from data of lower frequency, and this is reflected in the high errors displayed in Table 5.5. The tail index estimation is quite consistent but can significantly jump from one time interval to the next as is the case for GBP and CHF 6 months. This market is much less liquid than the FX market. Interest rate markets with higher liquidity can be studied in terms of interest rate derivatives, which are traded in

[12] Simulations in Gielens *et al* (1996) and McCulloch (1997) show that one cannot univocally distinguish between a fat-tailed nonstable and a thin-tailed distribution only on the basis of low estimated values of the tail index. However, the confidence intervals around the estimated tail index values and the diverging behavior of the kurtosis are strong evidence in favor of the fat-tail hypothesis.

[13] In most of the cases, except perhaps the 6-month interest rate for JPY, α is significantly larger than 2. In the JPY case, if the first 2 years are removed, we get back to values for α around 3.

TABLE 5.5 Estimated tail index for cash interest rates.

Estimated tail index values of cash interest rates and their 95% confidence ranges, for five different currencies and two maturities. The tail index values are based on the subsample bootstrap method using the Hill estimator, and the confidence ranges result from the jackknife method. The time intervals are measured in ϑ-time (see Chapter 6). The estimations are performed on samples going from January 2, 1979, to June 30, 1996; "m" refers to a month.

Currency	Maturity	1 day	1 week	Maturity	1 day	1 week
USD	3m	4.03 ±2.99	3.53 ±3.46	6m	4.10 ±2.84	3.50 ±3.07
DEM	3m	2.54 ±0.73	2.88 ±1.63	6m	2.39 ±0.76	2.62 ±1.82
JPY	3m	3.16 ±2.07	3.43 ±3.01	6m	2.03 ±0.85	3.60 ±3.53
GBP	3m	2.61 ±0.84	3.86 ±3.78	6m	4.04 ±2.64	6.65 ±7.53
CHF	3m	3.69 ±2.41	5.24 ±5.13	6m	3.02 ±1.26	7.46 ±7.31

futures markets such as London International Financial Futures Exchange (LIFFE) in London or Singapore International Monetary Exchange (SIMEX) in Singapore.

To study the stability of the tail index under aggregation, we observe how the estimates change with varying sample size. We do not know any theoretical tail index for empirical data, but we compare the estimates with the estimation done on the "best" sample we have, 30-min returns from January 1987 to June 1996. To study the tail index of daily returns, we use an extended sample of daily data from July 1, 1978, to June 30, 1996. The small-sample bias can be studied by comparing the results to the averages of the results from two smaller samples: one from June 1, 1978, to June 30, 1987, and another one from July 1, 1987, to June 30, 1996. These two short samples together cover the same period as the large sample. The results are given in Table 5.6.[14] When going from the short samples to the long 18-year sample, we see a general decrease of the estimated α toward the values reached for 30-min returns. The small-sample bias is thus reduced, but probably not completely eliminated. We conclude that, at least for the FX rates against the USD, the α estimates from daily data are not accurate enough even if the sample covers up to 18 years. The case of gold and silver is different because the huge fluctuations of the early 1980s have disappeared since then. The picture in this case is blurred by the changing market conditions.

A similar analysis of 30-min returns reinforces the obtained conclusions. In this case, the two shorter samples are from July 1, 1988, to June 30, 1992, and from July 1, 1992, to June 30, 1996. The large sample is again the union of the two shorter samples. A certain small-sample bias is found also for the 30-min returns of most rates comparing the two last columns of Table 5.3, but this bias is rather small. This is an expected result because the number of 30-min observations is much larger than that of the daily observations.

[14] For the short samples we do not give the errors because we present only the average of both samples. The errors are larger for the short samples than for the long sample.

TABLE 5.6 Estimated tail index for different samples sizes.

Estimated tail index for the main FX rates, gold (XAU) and silver (XAG) on different samples for both daily and 30-min returns. The time intervals are measured in ϑ-time (see Chapter 6).

FX rates	Daily returns		30-min returns	
	Short samples	7/1978–6/1996	7/1988–6/1996	Short samples
USD-DEM	4.84	4.34 ±2.46	3.27 ±0.50	3.29
USD-JPY	7.81	5.69 ±3.94	3.86 ±0.71	3.94
GBP-USD	4.79	4.35 ±3.02	3.37 ±0.53	3.57
USD-CHF	5.24	4.15 ±2.71	3.63 ±0.55	3.61
USD-FRF	4.48	4.37 ±2.85	3.52 ±0.54	3.59
USD-ITL	3.82	3.97 ±1.94	3.38 ±0.44	3.56
USD-NLG	4.17	4.05 ±1.98	3.56 ±0.66	3.57
XAU-USD	3.65	3.88 ±2.53	4.24 ±0.99	4.00
XAG-USD	3.94	3.40 ±1.92	4.12 ±0.75	3.54

5.4.3 Extreme Risks in Financial Markets

From the practitioners' point of view, one of the most interesting questions that tail studies can answer is what are the extreme movements that can be expected in financial markets? Have we already seen the largest ones or are we going to experience even larger movements? Are there theoretical processes that can model the type of fat tails that come out of our empirical analysis? The answers to such questions are essential for good risk management of financial exposures. It turns out that we can partially answer them here. Once we know the tail index, we can apply extreme value theory *outside* our sample to consider possible extreme movements *that have not yet been observed historically*. This can be achieved by a computation of the quantiles with exceedance probabilities.[15] Although this chapter focuses on stylized facts, it is interesting to show an example of the application of some of these empirical studies, which is very topical to risk management. There is a debate going on to design the best hedging strategy against extreme risks. Some researchers suggest using a dynamic method by utilizing conditional distributions (McNeil and Frey, 2000). We think that for practical purposes the hedge against extreme risk must be decided on the basis of the *unconditional* distribution. For a large portfolio, it would be impossible to find counterparties to hedge in very turbulent states of the market. Like in the case of earthquakes, hedging this type of risk needs to be planned far in advance.

Let us consider the expansion of the asymptotic cumulative distribution function from which the X_i observations are drawn as

$$F(x) \;=\; 1 - a\,x^{-\alpha}\left[1 + b\,x^{-\beta}\right] \qquad (5.5)$$

[15] We follow here the approach developed in Dacorogna *et al.* (2001a).

We denote by x_p and x_t quantiles with respective exceedance probabilities p and t. Let n be the sample size and choose $p < 1/n < t$; that is, x_t is inside the sample, while x_p is not observed. By definition (we concentrate on the positive tail),

$$p = ax_p^{-\alpha}\left[1 + bx_p^{-\beta}\right], \quad t = ax_t^{-\alpha}\left[1 + bx_t^{-\beta}\right] \tag{5.6}$$

Division of the two exceedance probabilities and rearrangement yields

$$x_p = x_t\left(\frac{t}{p}\right)^{1/\alpha}\left(\frac{1 + bx_p^{-\beta}}{1 + bx_t^{-\beta}}\right)^{1/\alpha} \tag{5.7}$$

Given that t is inside the sample, we can replace t by its empirical counterpart m/n, say; that is, m equals the number of order statistics X_t, which are greater than X_t. An estimator for x_p is then as follows

$$\hat{x}_p = X_m\left(\frac{m}{np}\right)^{\hat{\gamma}} \tag{5.8}$$

where m equals the $\hat{m}$ obtained from the tail estimation corresponding to $\hat{\gamma}$. To write this estimator we ignore the last factor on the right-hand side of Equation 5.7. This would be entirely justified in the case of the Pareto law when $b = 0$. Thus $\hat{x}_p$ is based on the same philosophy as the Hill estimator. For an m sufficiently small relative to n, the tails of Equation 5.5 are well approximated by those of the Pareto law, and hence Equation 5.8 is expected to do well. In fact, it is possible to prove (de Haan *et al.*, 1994) that, for the law in Equation 5.5

$$\frac{\sqrt{m}}{\ln\frac{m/n}{p}}\left(\frac{\hat{x}_p}{x_p} - 1\right) \tag{5.9}$$

has the same limiting normal distribution as the Hill estimator. Equation 5.9 gives us a way to estimate the error of our quantile computation.

Table 5.7 shows the result of a study of extreme risk using Equation 5.8 to estimate the quantiles for returns over 6 hr. This time interval is somewhat shorter than an overnight position (in ϑ-time) but is a compromise between the accuracy of the tail estimation and the length of the interval needed by risk managers. The first part of the table is produced by Monte Carlo simulations of synthetic data where the process was first fitted to the 30-min returns of the USD-DEM time series. The second part is the quantile estimation of the FX rates as a function of the probability of the event happening once every year, once every 5 years, and so on. Because we use here a sample of 9 years, the first two columns represent values that have been actually seen in the data set, whereas the other columns are extrapolations based on the empirically estimated tail behavior. Although the probabilities we use here seem very small, some of the extreme risks shown in Table 5.7 may be experienced by traders during their active life.

TABLE 5.7 Extreme risks in the FX market.

Extreme risks over 6 hr for model distributions produced by Monte-Carlo simulations of *synthetic* time series fitted to USD-DEM, compared to empirical FX data studied through a tail estimation.

	Probabilities (p)					
	1/1 year	1/5 year	1/10 year	1/15 year	1/20 year	1/25 year
Models:						
Normal	0.4%	0.5%	0.6%	0.6%	0.7%	0.7%
Student 3	0.5%	0.8%	1.0%	1.1%	1.2%	1.2%
GARCH(1,1)	1.5%	2.1%	2.4%	2.6%	2.7%	2.9%
HARCH	1.8%	2.9%	3.5%	4.0%	4.3%	4.6%
USD rates:						
USD-DEM	1.7%	2.5%	3.0%	3.3%	3.5%	3.7%
USD-JPY	1.7%	2.4%	2.9%	3.2%	3.4%	3.6%
GBP-USD	1.6%	2.3%	2.6%	2.9%	3.1%	3.2%
USD-CHF	1.8%	2.7%	3.1%	3.5%	3.7%	4.0%
USD-FRF	1.6%	2.3%	2.8%	3.0%	3.3%	3.4%
USD-ITL	1.8%	2.8%	3.4%	3.8%	4.1%	4.4%
USD-NLG	1.7%	2.5%	2.9%	3.2%	3.4%	3.6%
Cross rates:						
DEM-JPY	1.3%	1.9%	2.2%	2.5%	2.6%	2.8%
GBP-DEM	1.1%	1.7%	2.1%	2.3%	2.5%	2.6%
GBP-JPY	1.6%	2.3%	2.7%	3.0%	3.2%	3.4%
DEM-CHF	0.7%	1.0%	1.2%	1.3%	1.4%	1.5%
GBP-FRF	1.1%	1.8%	2.2%	2.5%	2.7%	2.9%

An interesting piece of information displayed in Table 5.7 is the comparison of empirical results and results obtained from theoretical models.[16] The model's parameters, including the variance of the normal and Student-t distributions, result from fitting USD-DEM 30-min returns. For the GARCH (1,1) model (Bollerslev, 1986), the standard maximum likelihood fitting procedure is used (Guillaume *et al.*, 1994) and the GARCH equation is used to generate synthetic time series. The same procedure is used for the HARCH model (Müller *et al.*, 1997a). The model's results are computed using the average $\hat{m}$ and $\hat{\gamma}$ obtained by estimating the tail index of 10 sets of synthetic data for each of the models for the aggregated time series over 6 hr. As expected, the normal distribution model fares poorly as far as the extreme risks are concerned. Surprisingly, this is also the case for

[16] The theoretical processes such as GARCH and HARCH are discussed in detail in Chapter 8. Here they simply serve as examples for extreme risk estimation.

the Student-t distribution with 3 degrees of freedom. The GARCH (1,1) model gives results closer to those of USD-DEM but still underestimates the risks by a significant amount. The HARCH model slightly overestimates the extreme risk. This is probably due to its long memory, which does not allow the process to modify sufficiently its tail behavior under aggregation. Obviously, further studies need to be pursued to assess how well models such as HARCH can predict extreme risks. In general, ARCH-type processes seem to capture the tail behavior of FX rates better than the simple unconditional distribution models. The advantage of having a model for the process equation is that it allows the use of a dynamic definition of the movements and it can hopefully provide an early warning in case of turbulent situations.

Paradoxically, in situations represented by the center of the distribution (nonextreme quantiles), the usual Gaussian-based models would overestimate the risk. Our study is valid for the tails of the distribution, but it is known that far from the tails the normal distribution produces higher quantiles than actually seen in the data.

5.5 SCALING LAWS

In this section, we examine the behavior of the absolute size of returns as a function of the frequency at which they are measured. As already mentioned, there is no privileged time interval at which the data and the generating process should be investigated. Thus it is important to study how the different measures relate to each other. One way of doing this is to analyze the dependence of mean volatility on the time interval on which the returns are measured. For usual stochastic processes such as the random walk, this dependence gives rise to very simple scaling laws (Section 5.5.2). Since Müller *et al.* (1990) have empirically documented the existence of scaling laws in FX rates, there has been a large volume of work confirming these empirical findings, including Schnidrig and Würtz (1995); Fisher *et al.* (1997); Andersen *et al.* (2000) and Mantegna and Stanley (2000). This evidence is confirmed for other financial instruments,[17] as reported by Mantegna and Stanley (1995) and Ballocchi *et al.* (1999b). The examination of the theoretical foundations of scaling laws are studied in Groenendijk *et al.* (1996); LeBaron (1999b), and Barndorff-Nielsen and Prause (1999).

Before discussing the literature, we present the empirical findings.

5.5.1 Empirical Evidence

The scaling law is empirically found for a wide range of financial data and time intervals in good approximation. It gives a direct relation between time intervals

[17] Brock (1999) has extensive discussions of scaling in economics and finance. He points out that most of these regularities are unconditional objects and may have little power to discriminate between a broad class of stochastic processes. Brock (1999) points out that the main object of interest in economics and finance is the conditional predictive distribution as in Gallant *et al.* (1993). Scaling laws may help in restricting the acceptable classes of conditional predictive distributions.

Δt and the average volatility measured as a certain power p of the absolute returns observed over these intervals,

$$\{E[|r|^p]\}^{1/p} = c(p)\, \Delta t^{D(p)} \quad (5.10)$$

where E is the expectation operator, and $c(p)$ and $D(p)$ are deterministic functions of p. We call D the drift exponent, which is similar to the characterization of Mandelbrot (1983, 1997). We choose this form for the left part of the equation in order to obtain, for a Gaussian random walk, a constant drift exponent of 0.5 whatever the choice of p. A typical choice is $p = 1$, which corresponds to absolute returns.

Taking the logarithm of Equation 5.10, the estimation of c and D can be carried out by an ordinary least squares regression. Linear regression is, in this case, an approximation. Strictly speaking, it should not be used because the $E[|r|]$ values for *different* intervals Δt are not totally independent. The longer intervals are aggregates of shorter intervals. Consequently, the regression is applied here to slightly dependent observations. This approximation is acceptable because the factor between two neighboring Δt is at least 2 (sometimes more to get even values in minutes, hours, days, weeks, and mean months), and the total span of analyzed intervals is large: from 10 min to 2 months. In addition, we shall see in Chapters 7 and 8 that volatility measured at different frequencies carries asymmetric information. Thus we choose the standard regression technique,[18] as Friedmann and Vandersteel (1982) and others do. The error terms used for $E[|r|]$ take into account the approximate basic errors of our prices[19] and the number of independent observations for each Δt.

The results presented here are computed for the cases $p = 1$ and $p = 2$. It is also interesting to compute instead of $\{E[|r|^p]\}^{1/p}$ the interquartile range as a function of Δt. Such a measure does not include the tails of the distribution. Recently, the development of the multifractal model (Fisher *et al.*, 1997) has brought researchers to study a whole spectrum of (also noninteger) values for the exponent p. In the same line of thought, multicascade models (Frisch, 1995) are also multifractals and involve different drift exponents for different measures of volatility. This feature is a typical signature of multifractality.

In Figure 5.8, we present empirical scaling laws for USD-JPY and GBP-USD for $p = 1$. Both the intervals Δt and the volatilities $E[|r|]$ are plotted on a logarithmic scale. The sample includes 9 years of tick-by-tick data from January 1, 1987, to December 31, 1995. The empirical scaling law is indeed a power law as indicated by the straight line. It is well respected in a very wide range of time intervals from 10 min to 2 months. The standard errors of the exponents D are less then 1%. The 1-day point in Figure 5.8 can be identified only by the corresponding label, not by any change or break between the intraday and interday behaviors.

[18] See Mosteller and Tukey (1977, Chapter 14).

[19] In Section 5.5.3, we give a full treatment of this problem.

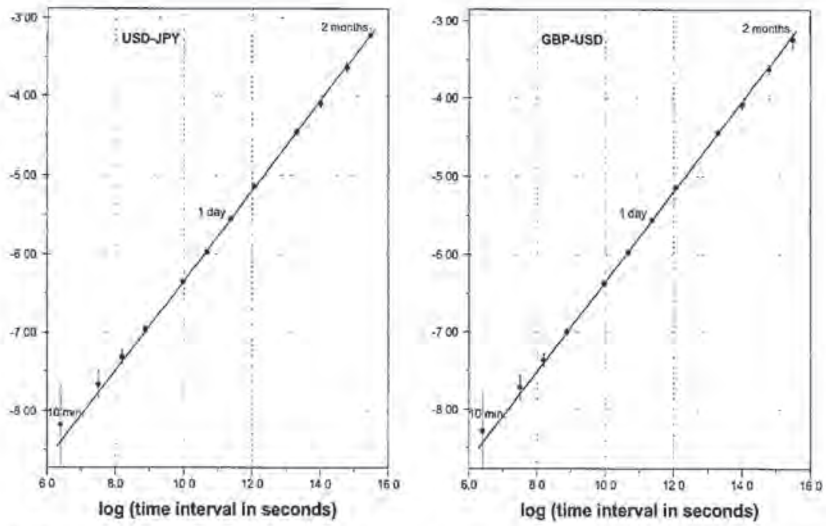


FIGURE 5.8 Scaling law for USD-JPY (right) and GBP-USD (left). On the y-axis, the natural logarithm of the mean absolute return ($p = 1$ in Equation 5.10) is reported. The error bars correspond to the mode described in Section 5.5.3. The sample period is January 1, 1987, to December 31, 1995.

Small deviations for the extreme interval sizes can be explained. The data errors grow on both sides, as discussed in Section 5.5.3. For long intervals, the number of observations in the sample becomes smaller and smaller, leading to a growing stochastic error. For very short intervals, the price uncertainty within the bid-ask spread becomes important. In fact, researchers such as Moody and Wu (1995) studied the scaling law at very high frequencies and obtained different exponents because they did not take into account the problem of price uncertainty. Recently, Fisher *et al.* (1997) also found a break of the scaling law around 2 hr. It is clear on both plots of Figure 5.8 that for time intervals shorter than 1 hour the points start to depart from a straight line. This deviation can be treated in two ways. In a first approach, we treat the price uncertainty as a part of the measurement error, leading to error bars that are as wide as to easily include the observed deviation at short time intervals. In a second approach, the deviation is identified as a bias that can be explained, modeled, and even eliminated by a correction. Both approaches are presented in Section 5.5.3.

In Table 5.8, we report the values of the drift exponent for four of the major FX rates against the USD and for gold for three different measures of volatility. Each of these measures treats extreme events differently. The interquartile range completely ignores them. The measure with $p = 2$ gives more emphasis to the tails than $p = 1$. The scaling law exponents D are around 0.57 for all rates and for $p = 1$, very close to 0.5 for $p = 2$, and around 0.73 for the interquartile range.

TABLE 5.8 Drift exponents for FX rates.

Drift exponents with standard errors found for the USD against DEM, JPY, CHF, and GBP and XAU against USD for two different powers and for the interquartile range. The sampling period extends from January 1, 1987, to December 31, 1995.

Currency	$E[\|r\|]$	$\{E[\|r\|^2]\}^{1/2}$	Interquartile
DEM	0.575 ± 0.006	0.501 ± 0.003	0.725 ± 0.017
JPY	0.570 ± 0.005	0.480 ± 0.035	0.691 ± 0.012
GBP	0.578 ± 0.004	0.514 ± 0.003	0.718 ± 0.011
CHF	0.574 ± 0.005	0.500 ± 0.002	0.737 ± 0.015
XAU	0.576 ± 0.005	0.491 ± 0.002	0.754 ± 0.012

These numbers are somewhat lower than those published by Müller *et al.* (1990), but this earlier study only covered 3 years of data. It seems that over the years the drift exponent for $p = 1$ has slightly decreased from typically 0.59 to 0.57.

The lower the weight of tails in the statistics, the more the empirically determined drift exponent deviates from the Gaussian random walk value of 0.5. This behavior is a consequence of the changing form of the distribution under aggregation and can also be seen as a sign of multifractality, as mentioned earlier. We repeated our analysis with only studying either negative returns or positive ones. The results show no significant asymmetry of positive and negative changes in accordance to the studies of Section 5.4.1. All results indicate a very general scaling law that applies to different currencies as well as to commodities such as gold and silver (which was additionally tested with a smaller sample). This *phenomenological* law becomes more important as the return distributions are unstable and the scaling law cannot be explained as a trivial consequence of a stable random process. This point will be discussed further in Section 5.5.2. Besides the evidence presented in Section 5.4, we find further evidence here for an unstable distribution because the drift exponent changes for the different measures of volatility. We find lower exponents $D \approx 0.5$ for $\{E[|r|^2]\}^{1/2}$ and higher exponents $D \approx 0.7$ for the interquartile ranges, and these can only be explained by varying distribution forms for different time intervals.

Similar scaling laws have been found by Ballocchi *et al.* (1999b) in a study of Eurofutures contracts[20] on the London International Financial Futures Exchange (LIFFE) and the Chicago Mercantile Exchane (CME) and in stock indices by Mantegna and Stanley (2000). We report here the results for the drift exponent for $p = 1$ for various contracts in Table 5.9. Here again the scaling law displays a drift exponent significantly larger than that expected for a Gaussian random walk and very close to the values obtained for foreign exchange rates. The table

[20] For a full description of these data and how they are treated for such a study, see Section 2.4.

TABLE 5.9 Drift exponents for Eurofutures.

Drift exponents with standard errors found for Eurofutures contracts. The drift exponent is for $E[|r|]$ ($p = 1$). All values are significantly larger than 0.5.

Expiry	Eurodollar	Euromark	Sterling
March 1995	0.60 ± 0.02	0.60 ± 0.01	0.61 ± 0.02
June 1995	0.66 ± 0.02	0.65 ± 0.01	0.62 ± 0.02
September 1995	0.68 ± 0.02	0.66 ± 0.01	0.62 ± 0.02
December 1995	0.64 ± 0.02	0.66 ± 0.01	0.64 ± 0.02
March 1996	0.57 ± 0.03	0.66 ± 0.01	0.63 ± 0.02
June 1996	0.70 ± 0.01	0.62 ± 0.01	0.62 ± 0.02
September 1996	0.70 ± 0.01	0.65 ± 0.01	0.62 ± 0.01
December 1996	0.69 ± 0.01	0.63 ± 0.02	0.60 ± 0.02
March 1997	0.66 ± 0.02	0.62 ± 0.02	0.63 ± 0.02

presents quite a dispersion of the exponents because the sample for each contract is relatively short. As a second step, we have repeated the scaling law analysis on an average of contracts. We averaged the mean absolute values of the returns (associated with each time interval) on the number of contracts. When the analysis referred to single Eurofutures, the average was computed on 9 contracts; when it referred to all Eurofutures and all contracts together, the average was computed on 36 contracts. Then we performed a linear regression for the logarithm of the computed averages against the corresponding logarithm of time intervals, taking the following time intervals: 1 day, 2 days, 1 week, 2 weeks, 4 weeks, 8 weeks, and half a year. The resulting drift exponent is 0.599 ± 0.007, remarkably close to the FX results. Ballocchi (1996) performed such studies for interbank short-term cash rates and computed the drift exponent of the mean absolute return averaged over 72 rates (12 currencies and 6 maturities from 1 month to 1 year), and the result is again 0.569 ± 0.007, close to the numbers listed in Table 5.8. To summarize, we find drift exponents of the mean absolute return of around 0.57 for FX rates and for cash interest rates, and 0.6 for Eurofutures.

5.5.2 Distributions and Scaling Laws

In this section we discuss how the distributions relate to the scaling law. There are remarkably few theoretical results on the relation between the drift exponents and the distribution aside from the trivial result for Gaussian distributions where all the exponents are 0.5. We only know of two recent papers that deal with this problem, Groenendijk *et al.* (1996) and Barndorff-Nielsen and Prause (1999). The latter gives $E[|r|]$ as a function of the aggregation and the parameters of the Normal Inverse Gaussian (NIG) Lévy process. The data generated from this process do not exhibit a scaling behavior, but the relationship is very close to a straight line when expressed in a double-logarithmic scale. The authors show that, with a particular choice of parameters, they can reproduce what they call the "apparent scaling" behavior of the USD-DEM data. There are other processes that present "apparent scaling" behavior. An example of such a process is given by LeBaron (1999a). In the literature, most scaling law results are of an empirical nature and are stylized facts directly obtained from the actual data. These results do not assume any particular data-generating process. Therefore, formal statistical tests are needed to test whether the empirically observed scaling laws are consistent with a particular type of null distribution. Although the findings of these statistical tests would not change the presence of empirical scaling laws, they would provide guidance for modeling return and volatility dynamics with those distributions consistent with the observed data dynamics.

We shall use here the approach of Groenendijk *et al.* (1996) to present a theorem they prove, which gives a good understanding of the relation between the drift exponent and other distributional properties for i.i.d. distributions. If we assume a simple random walk model

$$x(t) = x(t - \Delta t) + \varepsilon(t) \tag{5.11}$$

where x is the usual logarithmic price and the innovations $\varepsilon(t)$ are i.i.d., then the n-period return $r[n\Delta t](t)$ has the form

$$r[n\Delta t](t) = x(t) - x(t - n\Delta t) = \sum_{i=0}^{n-1} \varepsilon(t - i\Delta t) \tag{5.12}$$

where we have used Equation 5.11. In particular, if the ε_i's follow a normal distribution with mean zero and variance σ^2, the variance of $r[n\Delta t](t)$ is equal to $n\sigma^2$.

Groenendijk *et al.* (1996) consider the following quantity:

$$\ln\left(E\left[|r(n\Delta t)|^p\right]\right) - \ln\left(E\left[|r(\Delta t)|^p\right]\right) \tag{5.13}$$

where Δt is the smallest time interval and $\{r(n\Delta t)\}$ is the sequence of returns aggregated n-times. This quantity is directly related to the left-hand term of the scaling law shown in Equation 5.10.

The theorem they prove is related to the notion of stable distributions. Let us briefly state what this notion means. This class of distributions has the following attractive property. Let $\{a_n\}_{n=1}^{\infty}$ denote a sequence of increasing numbers such that $a_n^{-1} \sum_{i=1}^{n} \varepsilon_i$ has a nondegenerate limiting distribution, then this limiting distribution must belong to the class of stable distributions (Ibragimov and Linnik, 1971, p. 37). Ibragimov and Linnik (1971) show that the numbers a_n, which satisfy this requirement, are of the form $a_n = n^{1/\alpha} s(n)$, where $s(n)$ is a slowly varying function, that is,

$$\lim_{n\to\infty} \frac{s(tn)}{s(n)} = 1$$

with $t > 0$. Therefore, as $r(n\Delta t)$ is of the form $\sum_{i=1}^{n} \varepsilon_i$, we can expect $r_t(n\Delta t)$ to be of the order a_n (i.e., of the order $n^{1/\alpha}$), for large values of n. As a result, the dominating term in Equation 5.13 will be $p\ln(n)/\alpha$, such that the relationship between Equation 5.13 and $\ln(n)$ will be linear with slope p/α for all distributions that satisfy a generalized central limit law.

Theorem: *Let* $\{\varepsilon_t\}_{t=1}^{\infty}$ *denote a sequence of i.i.d. random variables with common distribution* $F(.)$. *Let* $F(.)$ *belong to the domain of attraction of a stable law with index* α. *Let* p *be such that* $0 \le p < \alpha$ *for* $\alpha < 2$ *and* $0 \le p \le 2$ *for* $\alpha = 2$. *Then*

$$\lim_{n\to\infty} \frac{\ln(E[|r(n\Delta t)|^p]) - \ln(E[|r(\Delta t)|^p])}{\ln(n)} = \frac{p}{\alpha} \qquad (5.14)$$

Let us now consider the case $p = 2$. Following the theorem, the class of distributions of $\varepsilon(t)$ must now be restricted to the distribution with finite variance. Using the independence assumption, we can easily obtain the following result:

$$E[|r(n\Delta t)|^2] = E\left[\left|\sum_{i=0}^{n-1} \varepsilon(t - i\Delta t)\right|^2\right] = \sum_{i=0}^{n-1} E[\varepsilon^2(t - i\Delta t)] = n\sigma^2 \qquad (5.15)$$

Consequently, the numerator in Equation 5.14 reduces to $\ln(n)$ and the fraction to 1. This is in accordance with the theorem, as it follows from the standard limit theory that distributions with finite variance lie in the domain of attraction of the normal distribution, which has an $\alpha = 2$, which, in this particular case, should not be confused with the tail index α of Section 5.4.2. In our study of tail indices, the conclusion was that the second moment of the distribution was finite (our tail indices were all largely above 2). Also, the results lead to a value of 0.5 (Table 5.8), which is the value one should theoretically obtain for $p = 2$ (as we were studying the square root of the second moment). There is, though, a difference between our empirical results and this theoretical result for $p = 1$ and for the interquartile range. There are at least three explanations for this. The first is that the distribution $F(.)$ of the random variable is not really common, which can be caused by heteroskedasticity. The second explanation is that the theorem is an asymptotic result and we might not have converged with our data to the limit. The third explanation is related to the i.i.d. assumption for the innovations under which the theoretical result is obtained. A drift exponent for $E[|r|]$, which is different from 0.5, could be an indication that there is a certain dependence between consecutive prices. We have already seen some of these dependencies for the very short-term (negative autocorrelation of returns) and we shall see some more in this and in the next chapter.

5.5.3 A Simple Model of the Market Maker Bias

In Section 5.5.1, we reported the findings published in Müller *et al.* (1990) about the scaling law of volatility measures. The parameters of this law seem very stable (see Guillaume *et al.*, 1997) but depend on the way the statistical quantities are computed and on the errors that enter the evaluation through the least square fit of the scaling law parameters. In Müller *et al.* (1990), we briefly mention the problem but in order to help people reproduce our results, we give here the full derivation of the error, which consists of a stochastic component and a market maker bias.

When making statistical studies of returns, researchers only consider the usual statistical error due to the limited number of observations. This error is clearly dominant when the return is measured over time intervals of a day or more. When the time interval is reduced to a few minutes, however, the *uncertainty* of the price definition due to the spread must be also considered. The market makers are biased toward one of the two prices, either the bid or the ask, thus introducing a bouncing effect that reflects in a negative autocorrelation of the returns in the very short-term (see Section 5.2.1, Goodhart and Figliuoli, 1991; Guillaume *et al.*, 1997). The true market price is between the bid and the ask quotes but not necessarily in their exact midpoint.[21] This uncertainty can be assessed to a considerable fraction of the nominal bid-ask spread. For short horizons, the amplitudes of price movements become comparable to the size of the spread. If spreads are large (especially minor FX rates), the uncertainty implies an important measurement error. For bid-ask data from electronic exchanges and transaction price data, the measurement error due to uncertainty is smaller or even negligible.

The purpose of this section is to derive the error on the statistical quantity entering the scaling law computation when the measurement error is also taken into account. We derived this model independently (Müller *et al.*, 1995) but it turns out that it is very similar to the model developed by Roll (1984) for equity prices.

We rely here on the definitions provided in Chapter 3 for the quantities we are going to use. The scaling law (see Equation 5.10) is empirically computed by fitting its logarithmic form,

$$\log(\overline{|\Delta x|}) = D \log \Delta t + \log c \tag{5.16}$$

The law becomes linear in this form. For the linear regression, we need to know the errors of $\log \overline{|\Delta x|}$. We saw that there is a similar scaling law for $(\langle |r|^2 \rangle)^{1/2}$:

$$\frac{1}{2} \log \left(\langle |r|^2 \rangle \right) = D' \log \Delta t + \log c' \tag{5.17}$$

The problem is to find the error on $\overline{|\Delta x|}$ knowing that we have an uncertainty related to the price definition and to the spread. Expressions with absolute values

[21] For a discussion of this point, see Müller *et al.* (1990); Bollerslev and Domowitz (1993); Goodhart and Payne (1996).

such as $\overline{|\Delta x|}$ are known for their poor analytical tractability. Therefore, the whole error computation is done for the analogous case of $(\langle |r|^2 \rangle)^{1/2}$. Following the arguments given in the introduction, let us assume that x_i^* is the series of true logarithmic market prices whereas the observed middle values x_i as defined by Equation 3.6 are subject to an additional market maker bias ε_i:

$$x_i \;=\; x_i^* + \varepsilon_i \tag{5.18}$$

The true return is defined analogous to Equation 3.7:

$$r_i^* \;\equiv\; r^*(t_i) \;\equiv\; x_i^* - x_{i-1}^* \tag{5.19}$$

Its relation to the observed return follows from Equations 3.7 and 5.18,

$$r_i \;=\; r_i^* + \varepsilon_i - \varepsilon_{i-1} \tag{5.20}$$

To compute the error, a minimum knowledge on the distribution of the stochastic quantities is required. We know that the returns r_i and r_i^* follow a Gaussian distribution only as a crude approximation (see Section 5.4), and the market maker bias ε_i might also be nonnormally distributed. Nevertheless, we shall assume Gaussian distributions (the same assumption is used in Roll, 1984) as approximations to make the problem analytically tractable:

$$r_i^* \in \mathcal{N}\left(0, \varrho^{*2}\right) \tag{5.21}$$

and

$$\varepsilon_i \in \mathcal{N}\left(0, \frac{\eta^2}{2}\right) \tag{5.22}$$

where $\mathcal{N}(\mu, \sigma^2)$ is the Gaussian probability distribution with mean μ and variance σ^2. The maximum deviation of the market maker bias is the distance between the middle price and the bid price (or ask price), that is half the spread. As a coarse approximation, we assume that the standard deviation is half that value, that is one-fourth of the typical value of the relative spread (Equation 3.12). This means that η^2 is assumed to be one-eighth of the squared relative spread.[22] Studies with transaction prices have shown that the "true" spread is very different from the quoted spread (see Section 5.2.2). This quantity is a kind of convention; the market maker is really interested in one of the bid or ask prices and adds or subtracts a canonical value to the price she or he wants to use. In normal market conditions, the price is settled with an equal distribution of buyers and sellers (same assumption as in Roll, 1984). Thus, in a first approximation, the two random variables, the true returns and the market maker bias as it appears in quoted prices can well be assumed to be independent. The choice of the market maker bias is somewhat

[22] We neglect spread changes.

arbitrary and depends on the model used. In the context of an efficient market with no arrival of information, Roll (1984) has assumed a similar bias.

Now, we are ready to compute the expectation of r_i^2 from Equation 5.20, using Equations 5.21 and 5.22 and the independence of r_i^* and ε_i,

$$\varrho^2 \equiv \mathrm{E}\left(r_i^2\right) = \mathrm{E}\left(\langle r^2 \rangle\right) = \varrho^{*2} + \eta^2 \qquad (5.23)$$

The squared observed returns are thus biased by the positive amount of η^2.

Empirical measures of $\langle r^2 \rangle$ are not only biased but also contain a stochastic error, which is defined as the deviation of $\langle r^2 \rangle$ from its expectation ϱ^2. The variance of this stochastic error can be formulated

$$\sigma^2 \equiv \mathrm{E}\left[\left(\langle r^2 \rangle - \varrho^2\right)^2\right] = \mathrm{E}\left(\langle r^2 \rangle^2 - 2\,\langle r^2 \rangle\,\varrho^2 + \varrho^4\right) \qquad (5.24)$$

The last form of this equation has the expanded terms of the square. The first term, $\langle r^2 \rangle^2$, can be explicitly written by inserting Equations 5.12 and 5.20; the other two terms can be simplified by inserting Equation 5.23. We obtain

$$\sigma^2 = \mathrm{E}\left\{\left[\frac{1}{n}\sum_{i=1}^{n}\left(r_i^* + \varepsilon_i - \varepsilon_{i-1}\right)^2\right]^2\right\} - \varrho^4 \qquad (5.25)$$

The first term is somewhat tedious to compute because of the two squares and the sum. We expand the squares to get many terms for which we have to compute the expectation values. All of those terms that contain r^* or ε to an odd power have a zero expectation due to the symmetry of the normal distribution and the independence of r^* and ε. The expectations of r^{*2} and ε^2 can be taken from Equations 5.21 and 5.22. The expectations of the fourth moments of normal distribution are

$$\mathrm{E}\left(r_i^{*4}\right) = 3\left[\mathrm{E}\left(r_i^{*2}\right)\right]^2 = 3\,\varrho^{*4} \qquad (5.26)$$

$$\mathrm{E}\left(\varepsilon_i^4\right) = \mathrm{E}\left(\varepsilon_{i-1}^4\right) = 3\left[\mathrm{E}\left(\varepsilon_i^2\right)\right]^2 = \frac{3}{4}\,\eta^4 \qquad (5.27)$$

as found in (Kendall *et al.*, 1987, pp. 321 and 338), for example. By inserting this and carefully evaluating all the terms, we obtain

$$\sigma^2 = \frac{n+2}{n}\,\varrho^{*4} + \frac{2\,(n+2)}{n}\,\varrho^{*2}\,\eta^2 + \frac{n^2+3\,n+1}{n^2}\,\eta^4 - \varrho^4 \qquad (5.28)$$

By inserting Equation 5.23, we can express the resulting stochastic error variance either in terms of ϱ^*,

$$\sigma^2 = \frac{2}{n}\,\varrho^{*4} + \frac{4}{n}\,\varrho^{*2}\,\eta^2 + \left(\frac{3}{n} - \frac{1}{n^2}\right)\eta^4 \qquad (5.29)$$

or in terms of ϱ,

$$\sigma^2 = \frac{2}{n}\varrho^4 + \left(\frac{1}{n} - \frac{1}{n^2}\right)\eta^4 \qquad (5.30)$$

Now, we know both the bias η^2 of an empirically measured $\langle r^2\rangle$ and the variance of its stochastic error. For reporting the results and using them in the scaling law computation, two alternative approaches are possible:

1. We can *subtract* the bias η^2 from the observed $\langle r^2\rangle$ and take the result with a stochastic error of a variance following Equation 5.30, approximating ϱ^2 by $\langle r^2\rangle$. We do not recommend this here because η^2 is only approximately known and thus contains an unknown error. However, the idea of bias modeling and bias elimination is further developed in Chapter 7.
2. We can take the originally obtained value of $\langle r^2\rangle$ and regard the bias η^2 as a *separate error component* in addition to the stochastic error. This is an appropriate way to go, given the uncertainty of η^2.

Following the second approach, we formulate a total error with variance σ^2_{total}, containing the bias and the stochastic error. The stochastic error is independent of the bias by definition, so the total error variance is the sum of the stochastic variance and the squared bias

$$\sigma^2_{\text{total}} = \sigma^2 + \eta^4 = \frac{2}{n}\varrho^4 + \left(1 + \frac{1}{n} - \frac{1}{n^2}\right)\eta^4 \qquad (5.31)$$

This is the final, resulting variance of the total error of $\langle r^2\rangle$.

For the application in the scaling law, we can use a good approximation for large values of $n \gg 1$, which is reasonable even for small values of n. By dropping higher order terms from Equation 5.31, we obtain

$$\sigma^2_{\text{total}} \approx \frac{2}{n}\varrho^4 + \eta^4 \approx \frac{2}{n}\langle r^2\rangle^2 + \eta^4 \qquad (5.32)$$

In the last form, the theoretical constant ϱ^2 has been replaced by its estimator $\langle r^2\rangle$, see Equation 5.23.

The mean squared return with error can be formulated as follows:

$$\langle r^2\rangle_{\text{with error}} = \langle r^2\rangle \pm \sqrt{\eta^4 + \frac{2}{n}\langle r^2\rangle^2} \qquad (5.33)$$

where the second term is the standard deviation of the error according to Equation 5.32.

The scaling law is usually formulated for $(\langle r^2\rangle)^{1/2}$ rather than $\langle r^2\rangle$, as in Equation 5.10. Applying the law of error propagation, we obtain

$$\begin{aligned}\langle r^2\rangle^{1/2}_{\text{with error}} &= \langle r^2\rangle^{1/2} \pm \frac{\mathrm{d}\langle r^2\rangle^{1/2}}{\mathrm{d}\langle r^2\rangle}\sqrt{\eta^4 + \tfrac{2}{n}\langle r^2\rangle^2} \\ &= \langle r^2\rangle^{1/2} \pm \sqrt{\frac{\eta^4}{4\langle r^2\rangle} + \frac{\langle r^2\rangle}{2n}}\end{aligned} \qquad (5.34)$$

The scaling law fitting is done in the linear form obtained for $\log(\langle r^2 \rangle)^{1/2}$ (see Equation 5.17). Again applying the law of error propagation, we obtain

$$
\begin{aligned}
\log\langle r^2 \rangle^{1/2}_{\text{with error}} &= \log\langle r^2 \rangle^{1/2} \pm \frac{\mathrm{d}\log\langle r^2 \rangle^{1/2}}{\mathrm{d}\langle r^2 \rangle^{1/2}} \sqrt{\frac{\eta^4}{4\,\langle r^2 \rangle} + \frac{\langle r^2 \rangle}{2\,n}} \\
&= \log\langle r^2 \rangle^{1/2} \pm \sqrt{\frac{\eta^4}{4\,\langle r^2 \rangle^2} + \frac{1}{2\,n}}
\end{aligned}
\tag{5.35}
$$

which gives rise to the following expression for the error variance of this quantity:

$$
\mathrm{Var}\left(\log\langle r^2 \rangle^{1/2}\right) = \frac{\eta^4}{4\,\langle r^2 \rangle^2} + \frac{1}{2\,n} \tag{5.36}
$$

The assumption is now that the variance of the error for $\log \overline{|\Delta x|}$ is approximately the same as that of $\log(\langle r^2 \rangle)^{1/2}$ in Equation 5.36 and that we only need to replace there the empirically obtained $(\langle r^2 \rangle)^{1/2}$ by the empirically obtained $\overline{|\Delta x|}$. This approximation is justified by the similar sizes and behaviors of both quantities. We obtain

$$
\mathrm{Var}\left(\log \overline{|\Delta x|}\right) \approx \frac{\eta^4}{4\,\overline{|\Delta x|}^4} + \frac{1}{2\,n} \tag{5.37}
$$

This expression has interesting properties. In the case of long time intervals, $\overline{|\Delta x|} \gg \eta$, and the term $1/(2n)$ becomes the essential cause for errors. In the case of short time intervals, n is very big but $\overline{|\Delta x|}$ is of the same order as η, and the first term of the right-hand side of the equation plays the central role. This explains the peculiar form of the errors in Figure 5.8, very large for high-frequency points, then diminishing (almost undistinguishable because of the high number of observations) and eventually increasing again when the number of observations becomes small.

5.5.4 Limitations of the Scaling Laws

We have already mentioned that the empirical results indicate a scaling behavior from time intervals of a few hours to a few months. Outside this range, the behavior departs from Equation 5.10 on both sides of the spectrum. Many authors noticed this effect, in particular, Moody and Wu (1995) and Fisher *et al.* (1997) for the short time intervals. It is important to understand the limitations of the scaling laws because realized volatility is playing more of an essential role in measuring volatility and thus market risk. It also serves as the quantity to be predicted in volatility forecasting and quality measurements of such forecasts, as we shall see in Chapter 8.

In the previous section, we saw that the bid and ask bounce generates an uncertainty on the middle price, and we have estimated its contribution to the error of the volatility estimation. For most risk assessment of portfolios, a good estimation of the *daily volatility* is required. Unfortunately, in practice, departures from the i.i.d. diffusion process make the realized volatility computed with returns measured at very short intervals no longer an unbiased and consistent estimator of the daily volatility. It is thus interesting to go one step further than in the previous section and to model the bias in order to obtain an easy correspondence between volatilities estimated at different frequencies. Recently, Corsi *et al.* (2001) have investigated and modeled the bias and its effect on measurements of realized volatility.

There are two limitations to the precision of the estimation of realized volatility. The number of return observations in the measurement period is limited and leads to a stochastic error (noise). One can easily see that for long time intervals (a year and more) it becomes difficult to assess the statistical significance of the volatility estimation because there are not more than a handful of independent observations. This number grows and the noise shrinks when the return measurement intervals shrink, but then the bias starts to grow. Until now, the only choice was a clever trade-off between the noise and the bias, which led to typical return intervals of about an hour. Tick frequency and data gaps play a major role. The goal is to define a superior realized volatility, which combines the low noise of short return interval sizes with the low bias of large return intervals. We shall not enter in the details here since it is still a research in progress. It suffices to mention that this is a crucial issue for a widespread use of high-frequency data in volatility estimation.

Gençay *et al.* (2001d) also provide evidence that the scaling behavior breaks for returns measured at higher intervals than 1 day. Figure 5.9 reports their decomposition of the variance on a scale-by-scale basis through the application of a nondecimated discrete wavelet transform.[23] This methodology does not assume any distributional form to the returns. The wavelet variance for each absolute return series is plotted on a log-log scale in Figure 5.9. For example the first scale is associated with 20-min changes, the second scale is associated with 2*20=40-min changes, and so on. Each increasing scale represents lower frequencies. The first six scales capture the frequencies $1/128 \leq f \leq 1/2$; that is, oscillations with a period length of 2 (40 min) to 128 (2560 min). Because there are 72*20 = 1440 min in a day, we conclude that the first six scales are related with intraday dynamics of the sample.

In the seventh scale, an apparent break is observed in the variance for both series which is associated with 64*20 = 1280-min changes. Because there are 1440 min in a day, the seventh scale corresponds to 0.89 day. Therefore, the seventh and higher scales are taken to be related with 1 day and higher dynamics.

For a power law process, $\nu^2(\tau_j) \propto \tau_j^{-\alpha-1}$ so that an estimate of α is obtained by regressing $\log \nu^2(\tau_j)$ on $\log \tau_j^{-\alpha-1}$. Figure 5.9 plots the ordinary least squares (OLS) fits of the sample points for two different regions. Estimated slopes for the smallest six scales are −0.48 and −0.59 for USD-DEM and USD-JPY series,

[23] A extensive study of wavelet methods within the context of time series analysis and filtering is presented in Gençay *et al.* (2001b).

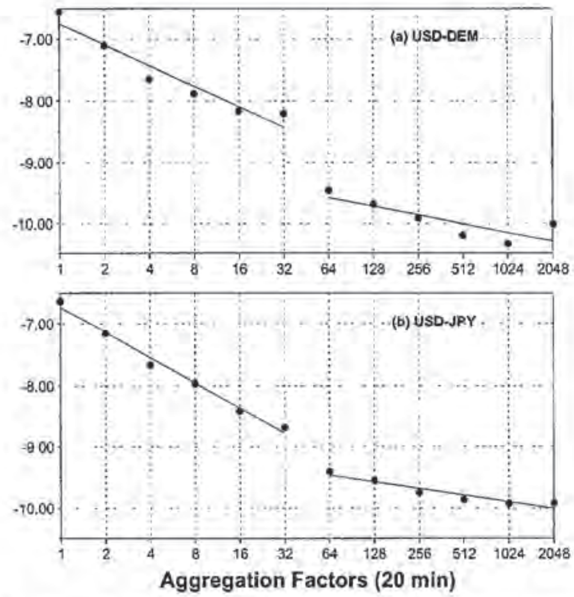


FIGURE 5.9 Wavelet variance for 20-min absolute returns of (a) USD-DEM and (b) USD-JPY from December 1, 1986, through December 1, 1996, on a log-log scale. The circles are the estimated variances for each scale. The straight lines are ordinary least squares (OLS) fits. Each scale is associated with a particular time period. For example the first scale reflects 20-min changes, the second scale reflects $2 * 20 = 40$-min changes, the third scale reflects $4 * 20 = 160$-min changes, and so on. The seventh scale is $64 * 20 = 1280$-min changes. Because there are 1440-min per day, the seventh scale corresponds to approximately one day. The last scale shows approximately 28 days.

respectively. This result implies that $\alpha = -0.52$ for the USD-DEM series and $\alpha = -0.40$ for the USD-JPY absolute return series for the first six scales (intraday).

5.6 AUTOCORRELATION AND SEASONALITY

Before closing this chapter, we investigate the autocorrelation and the seasonality of high-frequency data. Are returns and the volatility serially correlated, beyond the negative short-term autocorrelation in Section 5.2.1? Are there periodic patterns, seasonality, in the data? Clearly, we expect to find very little data during weekends and holidays, but what else can be said about different weekdays and daytimes? We answer these questions by using two types of statistical analysis. The autocorrelation function of a stochastic quantity reveals at the same time serially dependence and periodicity. The autocorrelation function signals a periodic pattern by peaking at lags that are integer multiples of the particular period. We call the other type of analysis an intraday-intraweek analysis. It relates quantities to the time of the day (or the week) when they are observed. We thus obtain average quantities evaluated for every hour of a day or of a week.

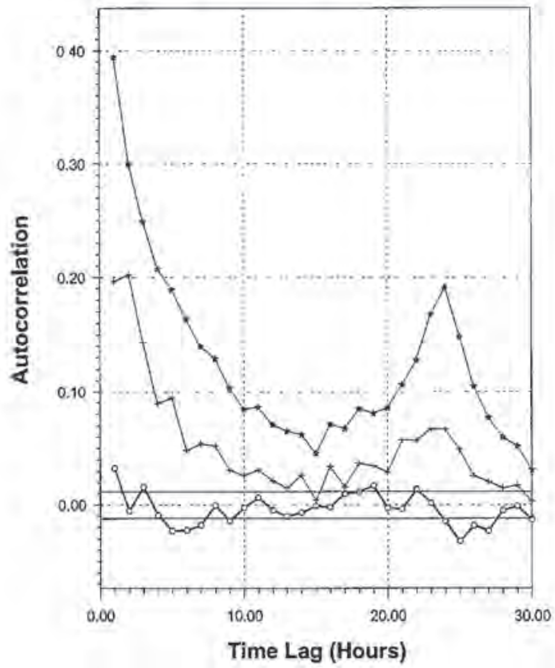


FIGURE 5.10 Autocorrelations of hourly returns (○), their absolute values (⋆), and their squares (+) as functions of the time lag, for XAU (gold) against USD. The band about the zero autocorrelation line represents 95% significance of the hypothesis of independent Gaussian observations.

5.6.1 Autocorrelations of Returns and Volatility

A convenient way to discover stylized properties of returns is to conduct an autocorrelation study. The autocorrelation function examines whether there is a linear dependency between the current and past values of a variable:

$$\rho_x(\tau) = \frac{\sum[x(t-\tau) - \langle x \rangle][x(t) - \langle x \rangle]}{\sqrt{\sum[x(t-\tau) - \langle x \rangle]^2 \sum[x(t) - \langle x \rangle]^2}} \tag{5.38}$$

where $x(t)$ can be any time series of a stochastic variable and τ is the time lag. The autocorrelation function peaks at lags corresponding to the periods of seasonal patterns.

Here we present an analysis of the autocorrelation function ρ of hourly returns, their absolute values, and their squares over a sample of 3 years from March 1, 1986, to March 1, 1989. We see in Figure 5.10 that the last two variables have a significant, strong autocorrelation for small time lags (few hours), which indicates the existence of volatility clusters or patterns. More interesting is the significant peak for time lags of and around 24 hr. This is a strong indication of *seasonality* with a period of 1 day. The autocorrelation of the returns

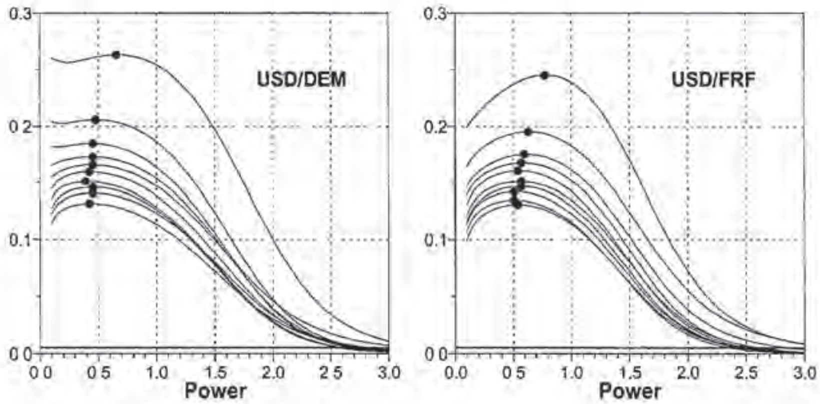


FIGURE 5.11 The first 10 lags of the autocorrelation function of $|r|^p$ as a function of the power p for USD-DEM and USD-FRF (first lag on top, 10th at the bottom). The maxima are shown by bullet signs (•). The returns are measured over 30 min (in ϑ-time, see Chapter 6). Right above the bottom, at a very low autocorrelation value, there is a horizontal line in both graphs. This is the upper limit of the 95% confidence band of the hypothesis of independent Gaussian observations.

themselves is insignificant as it stays mostly inside the confidence band of Figure 5.10.

Because the autocorrelation function varies when the absolute returns are raised to a different power, as can be seen in Figure 5.10, we systematically studied the influence of the power p on the autocorrelation function. Some studies on the influence of the power of absolute returns on the autocorrelation have been published (Granger and Ding, 1995; Müller *et al.*, 1998; Bouchaud and Potters, 2000). Granger and Ding (1995) conclude that the exponent $p = 1$ leads to the highest autocorrelation. Here, as in Müller *et al.* (1998), we report a full analysis of the autocorrelation coefficients as a function of the power p of absolute returns. Figure 5.11 shows how the tails of the distribution influence autocorrelation. Increasing the power of the absolute returns boils down to increasing the relative importance of extreme events in the statistics. In Figure 5.11, we see that the autocorrelation, for the 10 lags considered, decreases when the influence of extreme returns is increased. In other words, extreme events are less correlated with each other than average or small absolute returns. From this study, it seems that the heteroskedasticity is mainly due to the average behavior, not the extreme events. This is represented by a low exponent p smaller than 1; the maximum autocorrelation is for values of p close to one-half.

The results presented in Table 5.3 can well explain a feature shown in Figures 5.10 and 5.11 where the positive autocorrelation of absolute returns is stronger than that of squared returns. The tail index α being almost always below 4, the

TABLE 5.10 Time conversion table.

Time conversion table between Greenwich Mean Time (GMT), Europe (MET), USA (EST), and Japan (JPT). The letter D indicates a particular day. Note that GMT and JPT are not subject to daylight saving changes whereas the regions under MET and EST are.

GMT		MET		EST		JPT	
D	0:00	D	1:00	D-1	19:00	D	9:00
D	3:00	D	4:00	D-1	22:00	D	12:00
D	6:00	D	7:00	D	1:00	D	15:00
D	9:00	D	10:00	D	4:00	D	18:00
D	12:00	D	13:00	D	7:00	D	21:00
D	15:00	D	16:00	D	10:00	D+1	0:00
D	18:00	D	19:00	D	13:00	D+1	3:00
D	21:00	D	22:00	D	16:00	D+1	6:00

fourth moment of the distribution is unlikely to converge.[24] The denominator of the autocorrelation coefficient of squared returns is only finite if the fourth moment converges, whereas the convergence of the second moment[25] suffices to make the denominator of the autocorrelation of absolute returns finite. The second moment is finite if α is larger than 2. Indeed, besides the empirical evidence shown in Figure 5.11, we find that the difference between the autocorrelations of absolute returns and squared returns grows with increasing sample size. This difference computed on 20 years of daily data is much larger than that computed on only 8 years. For a lag of 9 days, we obtain autocorrelations of 0.11 and 0.125 for the absolute returns over 8 and 20 years, respectively, while we obtain 0.072 and 0.038 for the squared returns, showing a strong decrease when going to a larger sample. The same effect as for the daily returns is found for the autocorrelation of 20-min returns, where we compared a 9-year sample to half-yearly samples.

5.6.2 Seasonal Volatility: Across Markets for OTC Instruments

The most direct way to analyze *seasonal heteroskedasticity* in the form of daily volatility patterns is through our *intraday statistics*. We construct a uniform time grid with 24 hourly intervals for the statistical analysis of the volatility, the number of ticks, and the bid-ask spreads.

The low trading activity on weekends implies a weekly periodicity of trading activity and is a reason for adding *intraweek* statistics to the intraday statistics. Both statistics are technically the same, but the intraweek analysis uses a grid of

[24] Loretan and Phillips (1994) come to a similar conclusion when examining the tail behavior of daily closing prices for FX rates.

[25] For a more formal proof of the existence of the autocorrelation function of stochastic variables obeying fat-tailed probability distribution, see Davis *et al.* (1999).

TABLE 5.11 Average number of ticks.

Average number of ticks for each day of the week (including weekends) for the USD against DEM, JPY, CHF, and GBP and XAU (gold) against USD. The sampling period is from January 1, 1987, to December 31, 1993. The tick activity has increased over the years except for XAU (gold).

	DEM	JPY	GBP	CHF	XAU
Monday	4888	2111	1773	1764	607
Tuesday	5344	2438	2043	2031	698
Wednesday	5328	2460	2033	2022	717
Thursday	5115	2387	1948	1914	702
Friday	4495	1955	1633	1670	675
Saturday	17	16	15	14	27
Sunday	168	181	46	34	6

seven intervals from Monday 0:00-24:00 to Sunday 0:00-24:00 (GMT). With this choice, most of the active periods of the main markets (America, Europe, East Asia) on the same day are included in the same interval. The correspondence of the hours between main markets is shown in Table 5.10. The analysis grids have the advantage of a very simple and clear definition, but they treat *business holidays* outside the weekends (about 3% of all days) as working days and thus bias the results. The only remedy against that would be a worldwide analysis of holidays, with the open question of how to treat holidays observed in only one part of the world. Another problem comes from the fact that daylight saving time is not observed in Japan while it is in Europe and in the United States. This changes the significance of certain hours of the day in winter and in summer when they are expressed in GMT. An alternative here would be to separate the analysis according to the winter and the summer seasons.

An analysis of trading volumes in the daily and weekly grids is impossible as there is no raw data available. The average numbers of ticks, however, give an idea about the worldwide market activity as a function of daytime and weekdays. They are counts of original quotes by representative market makers, though biased by our data supplier. The two bottom graphs of Figure 5.12 improve our knowledge of the intraday and intraweek studies. They show, for example, that even the least active hour, 3:00 to 4:00 GMT (noon break in East Asia), contains about 20 ticks for DEM, a sufficient quantity for a meaningful analysis. The intraweekly results are shown in Table 5.11. The ranking of the FX rates according to the amount of published quotes corresponds fairly well to the ranking obtained by the Bank of International Settlements (BIS) with its survey of the volume of transactions on the FX market (Bank for International Settlements, 1995, 1999).

Intraday volatility in terms of mean absolute returns is plotted in the two top histograms of Figure 5.12 for USD-DEM. Both histograms indicate distinctly uneven intraday-intraweek volatility patterns. The daily maximum of average

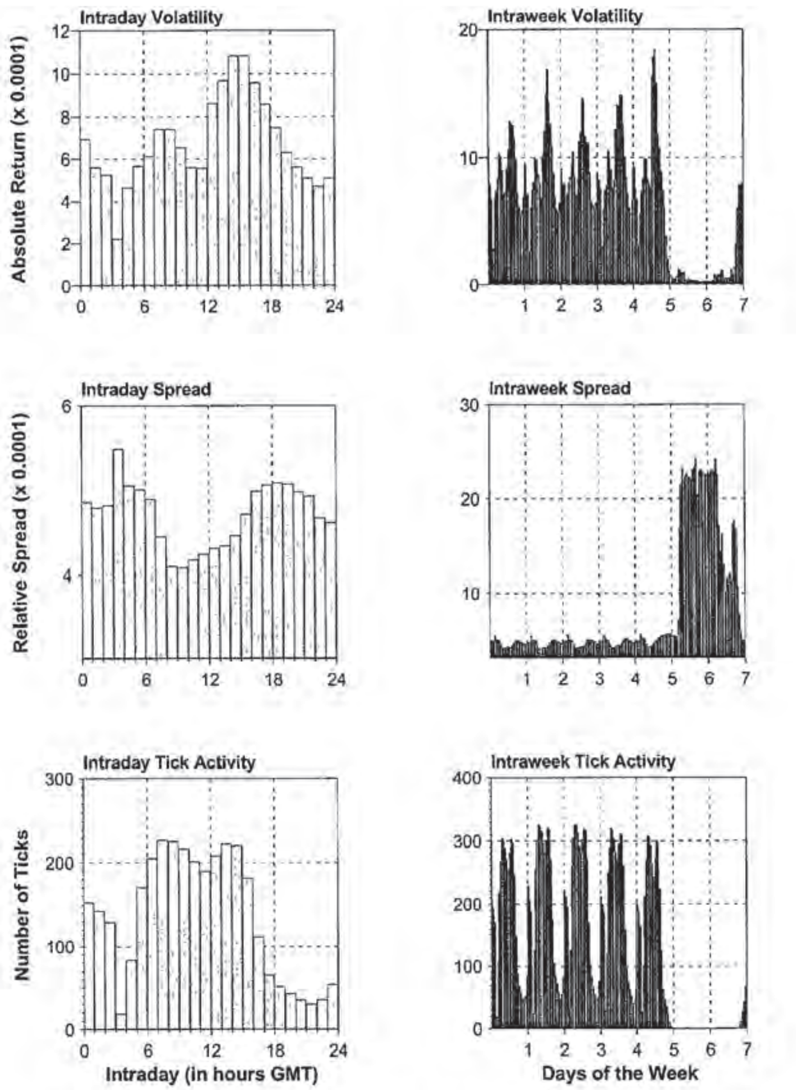


FIGURE 5.12 Hourly intraday and intraweek distribution of the absolute return, the spread and the tick frequency: a sampling interval of $\Delta t = 1$ hour is chosen. The day is subdivided into 24 hours from 0:00 – 1:00 to 23:00 – 24:00 (GMT) and the week is subdivided into 168 hours from Monday 0:00 – 1:00 to Sunday 23:00 – 24:00 (GMT) with index i. Each observation of the analyzed variable is made in one of these hourly intervals and is assigned to the corresponding subsample with the correct index i. The sample pattern does not account for bank holidays and daylight saving time. The FX rate is USD-DEM and the sampling period covers the 6 years from 1987 to 1992.

TABLE 5.12 Average volatility.

Average volatility for each day of the week (including weekends) for the USD against DEM, JPY, CHF, and GBP and XAU (gold) against USD; for the period from January 1, 1987, to December 31, 1993. The volatility figures have to be multiplied by 10^{-3}. They refer to one day. Corresponding *annualized* volatility figures are obtained through another multiplication by the factor $\sqrt{365.25} \approx 19.11$.

	DEM	JPY	GBP	CHF	XAU
Monday	6.12	4.66	5.44	6.04	5.75
Tuesday	5.28	5.17	5.49	5.88	5.48
Wednesday	4.93	5.02	5.04	5.52	5.47
Thursday	5.83	5.15	5.04	5.91	5.39
Friday	6.62	5.00	5.86	6.53	5.87
Saturday	0.58	0.74	0.76	0.88	1.19
Sunday	2.25	2.04	1.77	1.70	1.25

volatility is roughly four times higher than the minimum. The patterns can be explained by considering the structure of the world market, which consists of three main parts with different time zones: America, Europe, and East Asia. Even the lunch-break familiar to the European and East Asian markets, but not to the American one, can be detected in the form of the two minima of the histogram for USD-DEM. The main daily maximum occurs when both the American and the European markets are active. Other markets have similar patterns with characteristic differences in the weights of the markets, such as a higher volatility for the USD-JPY when the East Asian markets are active (as to be shown in Chapter 6). The patterns for USD-CHF and USD-DEM are similar, as expected. The pattern for XAU-USD reflects the well-known fact that the East Asian gold market is less active than the European and American ones. Different volatilities across the American, European, and Japanese markets were also detected by Ito and Roley (1987),[26] in their intraday study of the Japanese Yen.

Table 5.12 shows quite similar volatilities for the working days of the week. It does not confirm the weekend effect found by McFarland *et al.* (1982) with systematically lower volatilities on Fridays.[27] Their analysis was however different by taking daily changes at 18:00 GMT and putting together Saturdays, Sundays, and Mondays. The volatility is low on weekends, but, for FX rates, higher on Sundays than on Saturdays. This is due to the early Monday mornings in East Asia and in Australia, which coincide with Sunday nights in GMT.

The intraweek volatilities of Table 5.12 are correlated with the activities measured in terms of the number of ticks (Table 5.11). The analogous correlation

[26] For these authors volatility is measured by both the standard deviation and the mean absolute returns.

[27] The effect seems to be the reverse on the stock market (high returns on Fridays).