Chapter 9

Automated Internet Trading Based on Optimized Physics Models of Markets

L. Ingber and R.P. Mondescu

We describe a real-time, internet-based S&P futures trading system, including a description of general aspects of internet-mediated interactions with electronic exchanges. Inner-shell stochastic nonlinear dynamic models are developed, and Canonical Momenta Indicators (CMI) are derived from a fitted Lagrangian used by outer-shell trading models dependent on these indicators. Recursive and adaptive optimization using Adaptive Simulated Annealing (ASA) is used for fitting parameters shared across these shells of dynamic and trading models.

1 Introduction

Launching and exploiting a successful automated trading system implies accomplishing two major tasks, of almost equal significance:

- designing and developing a robust trading model of markets of interest,
- connecting the system to markets, addressing two problems
  - the communications hardware infrastructure,
  - the software interface.

To develop a robust and consistent model of markets, we should remark that real-world problems are rarely solved in closed algebraic form, yet methods must be devised to deal with this complexity to extract practical informations in finite time. This is indeed true in the field of financial engineering, where time series of various financial instruments reflect non-equilibrium, highly non-linear, possibly even chaotic (Peters 1991) underlying processes. A further difficulty is the huge amount of data necessary to be processed. Under these circumstances, to develop models and schemes for automated, profitable trading is a non-trivial task.

Apparently, the connectivity task involves mostly a programming effort, where a host of technical tools may considerably simplify the task. In practice an equal amount of work must be devoted to a proper design of various software components and solving multiple hardware problems, given the following constraints:

- necessity of accessing multiple markets.
- lack of a standard API (Application Programming Interface) for accessing different exchanges.
- lack of an universal language of communication between financial institutions.
- stringent reliability requirements posed on the communication infrastructure.

Currently, there are sustained efforts toward an unified, non-proprietary financial "electronic" language (FIX – Financial Information Exchange – open protocol (FIX Protocol 2000)). FIX approach is to define and promote a common set of types of messages, their format and the session-level interaction, for communicating securities transactions between two parties, in a real-time electronic trading environment.

1.1 Approaches

Detailed discussions pertinent to the theoretical model underlying the trading system and computational aspects were published previously, see (Ingber and Mondescu 2001).

Regarding the financial modeling aspect, in the context of this chapter, it is important to stress that dealing with such complex systems invariably requires modeling of dynamics, modeling of actions on these dynamics, and algorithms to fit parameters in these models to real data. We have elected to use methods of mathematical physics for our models of the dynamics, artificial intelligence (AI) heuristics for our models of trading rules acting on indicators derived from our dynamics, and methods of sampling global optimization for fitting our parameters. Too often there is confusion about how these three elements are being used for a complete system. For example, in the literature often there is discussion of neural net trading systems or genetic algorithm trading systems. However, neural net models (used for either or both models discussed here) also require some method of fitting their parameters, and genetic algorithms must have some kind of cost function or process specified to sample a parameter space, and so on.

Some powerful methods have emerged during years, appearing from at least two directions: One direction is based on inferring rules from past and current behavior of market data leading to learning-based, inductive techniques, such as neural networks, or fuzzy logic. Another direction starts from the bottom-up, trying to build physical and mathematical models based on different economic prototypes. In many ways, these two directions are complementary and a proper understanding of their main strengths and weaknesses should lead to synergetic effects beneficial to their common goals.

Among approaches in the first direction, neural networks already have won a prominent role in the financial community. This is due to their ability to handle large quantities of data and to uncover and model nonlinear functional relationships between various combinations of fundamental indicators and price data (Azoff 1994, Gately 1996).

In the second direction we can include models based on nonequilibrium statistical mechanics (Ingber 2000) fractal geometry (Mandelbrot 1997), turbulence (Mantegna and Stanley 1996), spin glasses and random matrix theory (Laloux *et al.* 1999), renormalization group (Johansen *et al.* 1999), and gauge theory (Ilinsky and Kalinin 1997). Although the very complex nonlinear multivariate character of financial markets is recognized (Hull 2000), these approaches seem to have had a lesser impact on current quantitative finance practice, although it is increasing becoming clear that this direction can lead to practical trading strategies and models.

To bridge the gap between theory and practice, as well as to afford a comparison with neural networks techniques, we focus on presenting an effective trading system of S&P futures, anchored in the physical principles of non-equilibrium statistical mechanics applied to financial markets (Ingber 1984, 2000).

Starting with nonlinear, multivariate, nonlinear stochastic differential equation descriptions of the price evolution of cash and futures indices, we build an algebraic cost function in terms of a Lagrangian. Then, a maximum likelihood fit to the data is performed using a global optimization algorithm, Adaptive Simulated Annealing (ASA) (Ingber 1993a). As firmly rooted in field theoretical concepts, we derive market canonical momenta indicators, and we use these as technical signals in a recursive ASA optimization that tunes the outer-shell of trading rules. We do not employ metaphors for these physical indicators, but rather derive them directly from models fit to data.

The outline of the chapter is as follows: Just below we briefly discuss the optimization method and momenta indicators.

In Section 2 we discuss some general, technical elements related to building an internet-based interface between the provider of financial services (e.g., an exchange) and the client using an electronic trading system.

In the ensuing two sections we establish the theoretical framework supporting our model, and the statistical mechanics approach together with the optimization method, respectively. In Section 5 we detail the trading system, and in Section 6 we describe our results. Our conclusions are presented in Section 7.

1.2 Optimization

Large-scale, non-linear fits of stochastic nonlinear forms to financial data require methods robust enough across data sets. (Just one day, tick data for regular trading hours could reach 10,000-30,000 data points.) Simple regression techniques exhibit deficiencies with respect to obtaining reasonable fits. They too often get trapped in local minima typically found in nonlinear stochastic models of such data. ASA is a global optimization algorithm that has the advantage – with respect to other global optimization methods as genetic algorithms, combinatorial optimization, and so on – not only to be efficient in its importance-sampling search strategy, but to have the statistical guarantee of finding the best optima (Ingber 1989, Ingber and Rosen 1993). This gives some confidence that a global minimum can be found, of course provided care is taken as necessary to tune the algorithm (Ingber 1996a).

It should be noted that such powerful sampling algorithms also are often required by other models of complex systems than those we use here (Ingber 1993b). For example, neural network models have taken advantage of ASA (Cohen 1994, Cozzio-Buëler 1995, Indiveri *et al.* 1993), as have other financial and economic studies (Mayer *et al.* 1996, Sakata and White 1998).

1.3 Indicators

In general, neural network approaches attempt classification and identification of patterns, or try forecasting patterns and future evolution of financial time series. Statistical mechanical methods attempt to find dynamic indicators derived from physical models based on general principles of non-equilibrium stochastic processes that reflect certain market factors. These indicators are used subsequently to generate trading signals or to try forecasting upcoming data.

In this chapter, the main indicators are called Canonical Momenta Indicators (CMI), as they faithfully mathematically carry the significance of market momentum, where the "mass" is inversely proportional to the price volatility (the "masses" are just the elements of the metric tensor in this Lagrangian formalism) and the "velocity" is the rate of price changes.

The concept of momentum is at least intuitively appreciated by all traders. Many traders use some algorithm to calculate the momenta of markets they are trading, e.g., perhaps to use as supplemental indicators to confirm other indicators to act on trades.

Markets increasingly are becoming inter-dependent, effectively defining a larger collective multivariate market. Many traders account for such circumstances by at least following indicators of other markets in addition to those they are explicitly trading. Clearly, it would be beneficial to have accurate measures of such inter-dependencies, beyond statistical correlations, to have indicators that measure the importance of inter-dependencies of the dynamic evolution of the markets. However, it also would be useful if such information could be presented in an understandable intuitive manner, without altering any detailed content. Canonical momenta can satisfy this wish-list, and a detailed application to trading is described below.

2 Connection to Electronic Exchanges

The growth of internet as a communication infrastructure and the exponential increase in computer power drastically altered the mechanics of securities trading. Electronic matching of orders eliminates market makers and brokers as intermediaries, allowing a vast increase in the number of market participants and better terms for financial execution of trading orders.

Despite more or less visible obstructions by the traditional players, electronic exchanges appeared or traditional exchanges converted to electronic ones (DTB – Germany, Matif – France, LIFFE – UK, Eurex – Germany and Switzerland merged futures exchanges) and their volume exploded (Burghardt 2001).

Intra-day price feeds, real-time streaming quotes (even order books – commonly referred to as Level II quotes – (Archipelago 2001)) and integrated trade systems are available at almost no cost, and complicated models could be programmed and run by all market participants.

Sophisticated automated systems at large financial institutions could browse a wealth of data and filtered it, based on various theoretical models, in the search of the arbitrage opportunity.

All these developments have made more prominent the role and the functionality of the interface connecting the trading system to the provider of financial services (which include both data sources and exchanges). By *financial services* we refer throughout to services related to trading (submission of orders, trading support or clearing services) provided by an exchange or other financial institutions to an end user client.

As a software application, a trading system has mainly two components: the computational kernel and the connection API. We talk here about the connection API at the client organization level. The API is the software layer allowing a trading tool of the client, the trader, to communicate with the software of the exchange or other provider of financial services.

Based on the data (prices, volume, time, various indicators) input and on the theoretical model used, the computational kernel generates the trading signals and sends them to the order execution module, a component of the connection API.

The connection API must address two classes of problems:

1. Access to real-time price quotes.
2. Execution of the trade order.

We remark that above and in what follows we choose to use – for clarity purposes – the term *connection API* as a rather broad grouping of functional units that may not necessarily reflect a more constrained software engineering point of view. For example, in most cases the data access component requires a separate, independent development effort from the order execution module.

A more complex, commercial version of a connection API should have certain features, among which we list

- enables universal access to multiple exchanges with unique API,
- allows proprietary trading tools or other systems to connect to the order execution system,
- provides compatibility with multiple financial instruments (stocks, bonds, futures, and so on),
- provides order routing service with real-time updates and various execution types and order qualifiers,
- provides back-office services (trade confirmations, profit/loss reports, execution reports, full order book update, settlement prices),
- provides market news services (market opening/closing announcements, market updates, instruments status/specifications changes),
- provides queries service: range of trades, range of prices, product specification changes.

Collecting and processing real-time price data could be done using 3rd party applications (two random examples: Reuters Triarch real-time services, ESignal data services (eSignal 2001)), or by directly writing into the API provided by the exchange, e.g., the Chicago Mercantile Exchange (CME) Market Data API – MDAPI 1.0 – or the Eurex Values/Gate 3.0 API (Eurex 2001).

Usually, most vendors provide integrated solutions, essentially trading applications that combine both the data and the execution systems. These applications are usually black-box systems that does not offer a lower level control of data, trading signals and trading orders, imperative requirements for building a proprietary trading tool.

We focus next on describing the technological and design aspects common to the connection API, with emphasis on the order routing component of the API. We choose to do so because it is more complex than the data access module and less details are available to a general audience.

2.1 Internet Connectivity: Overview

In general, connecting a trading system directly to one (or multiple) exchanges is a process requiring support and control from the dedicated technology and marketing departments of the exchange. It is reasonably understood that the trading system cannot be launched live without passing several quality control check-points, imposed both by in-house and exchange Quality Assurance (QA) departments.

The evolution of the trading application from concept to production tool could be subscribed to the following milestones:

- initial software development (concept, design, proto-type),
- advanced development,
- technical certification with sub-stages
  - functional testing,

  - failover/recovery testing,
  - stress testing,
- network certification,
- pre-production testing
  - connectivity testing,
  - clearing cycle (end-to-end) testing.

Associated with these development stages, various requirements (hardware and software) must be met within the automated trading environment. We describe these requirements below.

2.2 Internet Connectivity: Hardware Requirements

Reliable data feeds are critical components of a successful automated trading system. Internet access to exchanges through 3rd party applications/intermediaries and standard communication infrastructure (modems, cable modems, DSL, and so on) is possible, but due to reliability concerns and higher probability of connection breakdowns, it is limited for trading systems operating at longer time scales (daily, weekly trades) and lower trading volumes, or to personal trading.

When trading time scale decreases to minutes or seconds and large transactions, direct access to exchanges, with dedicated lines is required.

For both data access and order routing, the development, initial testing and certification phases require at least an ISDN line. The production stage necessitates frame relay (e.g., 256k AT&T) and ISDN connections as main communication backbone, and back-up lines, respectively.

Routers (e.g., Cisco 800, 2610) and possibly, a separate diagnostic line, are also required, as well as some 3rd party software applications (e.g., Reuters TIBCO).

All this equipment is usually installed by exchange personnel in collaboration hardware manufacturers technical support. Costs and timelines for hardware deployment should be factored in when evaluating capabilities of a trading model.

2.3 Internet Connectivity: Software Requirements

Besides design aspects, important considerations are the choice of language and development platform. At this moment, preponderantly for trading engines requiring fast execution, Java still does not offer the required speed and reliability. The languages of choice remain C++ and C.

Although at the client level, the computational kernel could be developed on any software platform, the need to interface with the API provided by exchanges limits considerably the platform choices: currently, Windows NT and Sun Solaris are the preferred operating systems, with some exchanges supporting also IBM AIX.

Moreover, commercial development environments (as Microsoft Visual Studio or Sun Workshop) and sometimes 3rd party libraries (e.g., Rogue Wave (RogueWave 2001)) are also necessary (at least when reaching certification and production levels), as only these are usually supported by exchanges.

2.4 API Order Execution Module: Components and Functionality

In terms of design, the connection API must insulate the computational kernel of various code changes operated by outside providers (e.g., exchanges) to which the system is connected. Function of specific interests, various design patterns (factory, template, bridge, façade, adapter (Gamma *et al.* 1994)) could be applied.

The basic order of events necessary to be handled by the order routing and execution component of the connection API is:

1. initialization (instantiate various object factories, register with the server to receive responses, and so on),
2. connect to exchange API server (open session),
3. authenticate connection (login),
4. subscribe to a particular instrument (or multiple instruments), or to a particular field of a instrument (e.g., bid prices for a certain stock),
5. create and submit orders,
6. terminate communication with the exchange server and disconnect.

After opening the trading session, the connection API should insure (when queried) that connection status and execution reports are available.

Various types of order (market order, stop order, limit order, stop limit order, market if touched = the opposite of a stop order) and types of time-in-force (we list here only those suitable for automated trading) must be handled by the order routing module. The particular order type and time-in-force type applied in actual trading are chosen function of the characteristics of the trading model:

- fill-or-kill, a limit order, which is canceled if not filled immediately and completely,
- fill-and-kill, a limit order that, if not filled completely, all remaining quantity is cancelled,
- good-till-cancel, an order to be held until filled or until is cancelled.

Note that not all of these above qualifiers are necessarily supported by the exchange of interest.

The main task of the order execution API is to create orders. An order will contain several fields, among which we list the most important:

- order identification number,

- exchange identification code,
- instrument identifier,
- order type (market, limit, stop,...),
- execution type (fill-and-kill, and so on),
- price (for stop, limit, stop-limit orders),
- quantity,
- time of entry.

The order execution API component sends and receives (generally FIX-compliant) messages. We quote several of them below:

- single order (new order for a single instrument),
- cancel request (request to cancel an order),
- cancel/replace request (a request to cancel a previous order and replace it with a new order),
- status request (a request for status of an order),
- heartbeat (a periodic signal send by exchange server to verify that connection is alive),
- reject (the order was rejected by the exchange server),
- cancel reject (the cancel request send by the client was rejected by the exchange server),
- execution report.

Logic for taking appropriate action function of the message (or combination of messages) received must be implemented at the API level, in connection with signals produced by the computational engine.

Finally, from a development point of view, correct processing of previous categories of messages is essential. In particular some points need attention:

- the cancel/replace logic, which may depend on the exchange (e.g., with the CME FIX API the client needs to send a status request to check the state of an order),
- the closing of a session (should be done gracefully, otherwise lost messages or damaged session accounting could occur),

- error handling (all possible errors/exceptions should be dealt properly),
- connection management (a crucial component of a connection API. The API should dynamically monitor and react to connectivity problems).

3 Models

3.1 Langevin Equations for Random Walks

The use of Brownian motion as a model for financial systems is generally attributed to Bachelier (Bachelier 1900), though he incorrectly intuited that the noise scaled linearly instead of as the square root relative to the random log-price variable. Einstein is generally credited with using the correct mathematical description in a larger physical context of statistical systems. However, several studies imply that changing prices of many markets do not follow a random walk, that they may have long-term dependences in price correlations, and that they may not be efficient in quickly arbitraging new information (Jensen 1978, Mandelbrot 1971, Taylor 1982). A random walk for returns, rate of change of prices over prices, is described by a Langevin equation with simple additive noise η, typically representing the continual random influx of information into the market.

$$\begin{aligned} \dot{M} &= -f + g\eta , \\ \dot{M} &= \frac{dM}{dt} , \\ < \eta(t) >_\eta = 0, &\quad < \eta(t), \eta(t') >_\eta = \delta(t - t') , \end{aligned} \tag{1}$$

where f and g are constants, and M is the logarithm of (scaled) price, $M(t) = \log(P(t)/P(t - dt))$. Price, although the most dramatic observable, may not be the only appropriate dependent variable or order parameter for the system of markets (Brown *et al.* 1983). This possibility has also been called the "semi-strong form of the efficient market hypothesis" (Jensen 1978).

The generalization of this approach to include multivariate nonlinear non-equilibrium markets led to a model of statistical mechanics of financial markets (SMFM) (Ingber 1984).

3.2 Adaptive Optimization of F^x Models

Our S&P model for the evolution of futures price F is

$$
\begin{aligned}
dF &= \mu dt + \sigma F^x dz, \\
< dz > &= 0, \\
< dz(t)\, dz(t') > &= dt \delta(t - t'),
\end{aligned}
\tag{2}
$$

where the exponent x of F is one of the dynamical parameters to be fit to futures data together with μ and σ.

We have used this model in several ways to fit the distribution's volatility defined in terms of a scale and an exponent of the independent variable (Ingber 2000).

A major component of our trading system is the use of adaptive optimization, essentially constantly retuning the parameters of our dynamic model each time new data is encountered in our training, testing and real-time applications. The parameters $\{\mu, \sigma\}$ are constantly tuned using a quasi-local simplex code (Barabino *et al.* 1980, Nelder and Mead 1964) included with the ASA (Adaptive Simulated Annealing) code (Ingber 1993a).

We have tested several quasi-local codes for this kind of trading problem, versus using robust ASA adaptive optimizations, and the faster quasi-local codes seem to work quite well for adaptive updates after a zeroth order parameters set is found by ASA (Ingber 1996b,c).

4 Statistical Mechanics of Financial Markets (SMFM)

4.1 Statistical Mechanics of Large Systems

Aggregation problems in nonlinear nonequilibrium systems typically are "solved" (accommodated) by having new entities/languages developed at these disparate scales in order to efficiently pass information back and forth between scales. This is quite different from the nature of quasi-equilibrium quasi-linear systems, where thermodynamic or cybernetic approaches are possible. These thermodynamic approaches typically fail for nonequilibrium nonlinear systems.

Many systems are aptly modeled in terms of multivariate differential rate-equations, known as Langevin equations (Haken 1983),

$$\begin{aligned} \dot{M}^G &= f^G + \hat{g}^G_j \eta^j , (G = 1, \dots, \Lambda)(j = 1, \dots, N), \\ \dot{M}^G &= \frac{dM^G}{dt}, \\ < \eta^j(t) >_\eta = 0, &\quad < \eta^j(t), \eta^{j'}(t') >_\eta = \delta^{jj'} \delta(t - t'), \end{aligned} \tag{3}$$

where f^G and $\hat{g}^G_j$ are generally nonlinear functions of mesoscopic order parameters M^G, j is an index indicating the source of fluctuations, and $N \geq \Lambda$. The Einstein convention of summing over repeated indices is used. Vertical bars on an index, e.g., $|j|$, imply no sum is to be taken on repeated indices. The "microscopic" index j relates to the typical physical nature of fluctuations in such statistical mechanical systems, wherein the variables η are considered to be aggregated from finer scales relative to the "mesoscopic" variables M.

Via a somewhat lengthy, albeit instructive calculation, outlined in several other papers (Ingber 1984, 1991, Ingber *et al.*1991), involving an intermediate derivation of a corresponding Fokker-Planck or

Schrödinger-type equation for the conditional probability distribution $P[M(t)|M(t_0)]$, the Langevin rate Eq. (3) is developed into the more useful probability distribution for M^G at long-time macroscopic time event $t_{u+1} = (u+1)\theta + t_0$, in terms of a Stratonovich path-integral over mesoscopic Gaussian conditional probabilities (Cheng 1972, Dekker 1979, Graham 1978, Langouche *et al.*1979, 1980). Here, macroscopic variables are defined as the long-time limit of the evolving mesoscopic system.

The corresponding Schrödinger-type equation is (Graham 1978, Langouche *et al.* 1979)

$$\begin{aligned} \frac{\partial P}{\partial t} &= \frac{1}{2}(g^{GG'}P)_{,GG'} - (g^G P)_{,G} + V, \\ g^{GG'} &= \delta^{jk}\hat{g}^G_j \hat{g}^{G'}_k, \\ g^G &= f^G + \frac{1}{2}\delta^{jk}\hat{g}^{G'}_j \hat{g}^G_{k,G'}, \\ [\ldots]_{,G} &= \frac{\partial[\ldots]}{\partial M^G}. \end{aligned} \tag{4}$$

This is properly referred to as a Fokker-Planck equation when $V \equiv 0$. Note that although the partial differential Eq. (4) contains information regarding M^G as in the stochastic differential Eq. (3), all references to j have been properly averaged over. I.e., $\hat{g}^G_j$ in Eq. (3) is an entity with parameters in both microscopic and mesoscopic spaces, but M is a purely mesoscopic variable, and this is more clearly reflected in Eq. (4). In the following, we often drop superscripts on M for clarity, with the understanding that M represents the vector $\{M^G\}$.

The calculation of the long-time evolution of these distributions most often defies any algebraic solution, and special techniques must be utilized. This is required, for example, to calculate many kinds of financial instruments, e.g., bond prices, options, derivatives, and so on. People have developed numerical algorithms for each representation, i.e., for the Langevin, Fokker-Planck and the Lagrangian probability representations. Methods to treat the latter are developed around the path-integral formalism:

The path integral representation can be written in terms of the prepoint discretized Lagrangian L, further discussed below (Graham 1978, Langouche *et al.*1980, 1982),

$$\begin{aligned}
P[M,t|M,t_0]dM(t) &= \int \ldots \int \underline{D}M \exp(-S) \\
&\quad \times \delta[M(t_0)]\delta[M(t)], \\
&S = \min \int_{t_0}^{t} dt' L, \\
\underline{D}M &= \lim_{u\to\infty} \prod_{v=1}^{u+1} g^{1/2} \prod_{G} (2\pi\theta)^{-1/2} dM^G(t_v), \\
L(\dot{M}^G, M^G, t) &= \frac{1}{2}(\dot{M}^G - g^G) g_{GG'} (\dot{M}^{G'} - g^{G'}) \\
&\quad -V, \\
g_{GG'} &= (g^{GG'})^{-1}, \\
g &= \det(g_{GG'}).
\end{aligned} \tag{5}$$

Mesoscopic variables have been defined as M^G in the Langevin and Fokker-Planck representations, in terms of their development from the microscopic system labeled by j. The entity $g_{GG'}$, is a bona fide metric of this space (Graham 1978). Short-time "forecast" of data points is realized using the most probable path equation (Dekker 1980)

$$\frac{dM^G}{dt} = g^G - g^{1/2}(g^{-1/2} g^{GG'})_{,G'}. \tag{6}$$

In the literature on economics, there appears to be sentiment to define Eq. (3) by the Itô, rather than the Stratonovich prescription. It is true that Itô integrals have Martingale properties not possessed by Stratonovich integrals (Oksendal 1998) which leads to risk-neural theorems for markets (Harrison and Kreps 1979, Pliska 1997), but the nature of the proper mathematics – actually a simple transformation between these two discretizations – should eventually be determined by proper aggregation of relatively microscopic models of markets. It should be noted that virtually all investigations of other physical systems, which are also continuous time models of discrete processes, conclude that the Stratonovich interpretation coincides with reality, when multiplicative noise with zero correlation time, modeled in terms of white noise η^j, is properly considered as the limit of real noise with finite correlation time (Gardiner 1983). The path integral succinctly demonstrates the difference between the two: The Itô prescription corresponds to the prepoint discretization of L, wherein $\theta\dot{M}(t) \to M(t_{v+1}) - M(t_v)$ and $M(t) \to M(t_v)$. The Stratonovich prescription corresponds to the midpoint discretization of L, wherein $\theta\dot{M}(t) \to M(t_{v+1}) - M(t_v)$ and $M(t) \to \frac{1}{2}(M(t_{v+1}) + M(t_v))$. In terms of the functions appearing in the Fokker-Planck Eq. (4), the Itô prescription of the prepoint discretized Lagrangian L, Eq. (5), is relatively simple, albeit deceptively so because of its nonstandard calculus. In the absence of a nonphenomenological microscopic theory, the difference between a Itô prescription and a Stratonovich prescription is simply a transformed drift (Langouche *et al.* 1982).

There are several other advantages to Eq. (5) over Eq. (3). Extrema and most probable states of M^G, $\ll M^G \gg$, are simply derived by a variational principle, similar to conditions sought in previous studies (Merton 1973). In the Stratonovich prescription, necessary, albeit not sufficient, conditions are given by

$$\begin{aligned} &\delta_G L = L_{,G} - L_{,\dot{G}:t} = 0, \\ &L_{,\dot{G}:t} = L_{,\dot{G}G'}\dot{M}^{G'} + L_{,\dot{G}\dot{G}'}\ddot{M}^{G'}. \end{aligned} \tag{7}$$

For stationary states, $\dot{M}^G = 0$, and $\partial\bar{L}/\partial\bar{M}^G = 0$ defines $\ll \bar{M}^G \gg$, where the bars identify stationary variables; in this case, the macroscopic variables are equal to their mesoscopic counterparts.

Note that $\bar{L}$ is not the stationary solution of the system, e.g., to Eq. (4) with $\partial P/\partial t = 0$. However, in some cases (Ingber 1985), $\bar{L}$ is a definite aid to finding such stationary states. Many times only properties of stationary states are examined, but here a temporal dependence is included. E.g., the $\dot{M}^G$ terms in L permit steady states and their fluctuations to be investigated in a nonequilibrium context. Note that Eq. (7) must be derived from the path integral, Eq. (5), which is at least one reason to justify its development.

4.2 Algebraic Complexity Yields Simple Intuitive Results

It must be emphasized that the output of this formalism is not confined to complex algebraic forms or tables of numbers. Because L possesses a variational principle, sets of contour graphs, at different long-time epochs of the path-integral of P over its variables at all intermediate times, give a visually intuitive and accurate decision-aid to view the dynamic evolution of the scenario. For example, this Lagrangian approach permits a quantitative assessment of concepts usually only loosely defined.

$$\text{“Momentum''} = \Pi^G = \frac{\partial L}{\partial(\partial M^G/\partial t)}, \tag{8a}$$

$$\text{“Mass''} = g_{GG'} = \frac{\partial^2 L}{\partial(\partial M^G/\partial t)\partial(\partial M^{G'}/\partial t)}, \tag{8b}$$

$$\text{“Force''} = \frac{\partial L}{\partial M^G}, \tag{8c}$$

$$\text{“F = ma''}: \ \delta L = 0 = \frac{\partial L}{\partial M^G} - \frac{\partial}{\partial t}\frac{\partial L}{\partial(\partial M^G/\partial t)}, \tag{8d}$$

where M^G are the variables and L is the Lagrangian. These physical entities provide another form of intuitive, but quantitatively precise, presentation of these analyses. For example, daily newspapers use some of this terminology to discuss the movement of security prices.

In this chapter, the Π^G serve as canonical momenta indicators (CMI) for these systems.

4.2.1 Derived Canonical Momenta Indicators (CMI)

The extreme sensitivity of the CMI gives rapid feedback on changes in trends as well as the volatility of markets, and therefore are good indicators to use for trading rules (Ingber 1996b). A time-locked moving average provides manageable indicators for trading signals. This current project uses such CMI developed as a byproduct of the ASA fits described below.

4.2.2 Intuitive Value of CMI

In the context of other invariant measures, the CMI transform covariantly under Riemannian transformations, but are more sensitive measures of activity than other invariants such as the energy density, effectively the square of the CMI, or the information which also effectively is in terms of the square of the CMI (essentially integrals over quantities proportional to the energy times a factor of an exponential including the energy as an argument). Neither the energy or the information give details of the components as do the CMI. In oscillatory markets the relative signs of such activity can be quite important.

The CMI present single indicators for each member of a set of correlated markets, “orthogonal” in the defined metric space. Each indicator is a dynamic weighting of short-time differenced deviations from drifts (trends) divided by covariances (risks). Thus the CMI also give information complementary to just trends or standard deviations separately.

4.3 Correlations

In this chapter we report results of our one-variable trading model. However, it is straightforward to include multi-variable trading models in our approach, and we have done this, for example, with coupled cash and futures S&P markets.

Correlations between variables are modeled explicitly in the Lagrangian as a parameter usually designated ρ. This section uses a simple two-factor model to develop the correspondence between the correlation ρ in the Lagrangian and that among the commonly written Wiener distribution dz.

Consider coupled stochastic differential equations for futures F and cash C:

$$dF = f^F(F, C)dt + \hat{g}^F(F, C)\sigma_F dz_F, \tag{9a}$$
$$dC = f^C(F, C)dt + \hat{g}^C(F, C)\sigma_C dz_C, \tag{9b}$$
$$< dz_i > = 0,\ i = \{F, C\}, \tag{9c}$$
$$< dz_i(t)dz_j(t') > = dt\delta(t - t'), i = j, \tag{9d}$$
$$< dz_i(t)dz_j(t') > = \rho dt\delta(t - t'), i \neq j, \tag{9e}$$

where $< . >$ denotes expectations with respect to the multivariate distribution.

These can be rewritten as Langevin equations (in the Itô prepoint discretization)

$$\frac{dF}{dt} = f^F + \hat{g}^F \sigma_F(\gamma^+ \eta_1 + \mathrm{sgn}\rho\, \gamma^- \eta_2), \tag{10a}$$
$$\frac{dC}{dt} = g^C + \hat{g}^C \sigma_C(\mathrm{sgn}\rho\, \gamma^- \eta_1 + \gamma^+ \eta_2), \tag{10b}$$
$$\gamma^\pm = \frac{1}{\sqrt{2}}[1 \pm (1 - \rho^2)^{1/2}]^{1/2}, \tag{10c}$$
$$n_i = (dt)^{1/2} p_i, \tag{10d}$$

where p_1 and p_2 are independent [0,1] Gaussian distributions.

The equivalent short-time probability distribution, P, for the above set of equations is

$$
\begin{aligned}
P &= g^{1/2}(2\pi dt)^{-1/2}\exp(-Ldt),\\
L &= \frac{1}{2}M^{\dagger}\underline{g}M,\\
M &= \begin{pmatrix} \frac{dF}{dt} - f^F \\ \frac{dC}{dt} - f^C \end{pmatrix},\\
g &= \det(\underline{g}).
\end{aligned}
\tag{11}
$$

$\underline{g}$, the metric in $\{F, C\}$-space, is the inverse of the covariance matrix,

$$
\underline{g}^{-1} = \begin{pmatrix} (\hat{g}^F\sigma_F)^2 & \rho\hat{g}^F\hat{g}^C\sigma_F\sigma_C \\ \rho\hat{g}^F\hat{g}^C\sigma_F\sigma_C & (\hat{g}^C\sigma_C)^2 \end{pmatrix}. \tag{12}
$$

The CMI indicators are given by the formulas

$$
\Pi^F = \frac{(dF/dt - f^F)}{(\hat{g}^F\sigma_F)^2(1-\rho^2)} - \frac{\rho(dC/dt - f^C)}{\hat{g}^F\hat{g}^C\sigma_F\sigma_C(1-\rho^2)}, \tag{13a}
$$

$$
\Pi^C = \frac{(dC/dt - f^C)}{(\hat{g}^C\sigma_C)^2(1-\rho^2)} - \frac{\rho(dF/dt - f^F)}{\hat{g}^C\hat{g}^F\sigma_C\sigma_F(1-\rho^2)}. \tag{13b}
$$

4.4 ASA Outline

The algorithm Adaptive Simulated Annealing (ASA) fits short-time probability distributions to observed data, using a maximum likelihood technique on the Lagrangian. This algorithm has been developed to fit observed data to a theoretical cost function over a D-dimensional parameter space (Ingber 1989), adapting for varying sensitivities of parameters during the fit. The ASA code can be obtained at no charge, via WWW from http://www.ingber.com/ or via FTP from ftp.ingber.com (Ingber 1993a).

4.4.1 General Description

It helps to visualize the problems presented by such complex systems as a geographical terrain. For example, consider a mountain range, with two "parameters," e.g., along the NorthSouth and EastWest directions. We wish to find the lowest valley in this terrain. ASA approaches this problem similar to using a bouncing ball that can bounce over mountains from valley to valley. We start at a high "temperature," where the temperature is an ASA parameter that mimics the effect of a fast moving particle in a hot object like a hot molten metal, thereby permitting the ball to make very high bounces and being able to bounce over any mountain to access any valley, given enough bounces. As the temperature is made relatively colder, the ball cannot bounce so high, and it also can settle to become trapped in relatively smaller ranges of valleys.

We imagine that our mountain range is aptly described by a "cost function." We define probability distributions of the two directional parameters, called generating distributions since they generate possible valleys or states we are to explore. We define another distribution, called the acceptance distribution, which depends on the difference of cost functions of the present generated valley we are to explore and the last saved lowest valley. The acceptance distribution decides probabilistically whether to stay in a new lower valley or to bounce out of it. All the generating and acceptance distributions depend on "temperatures."

Simulated annealing (SA) was developed in 1983 to deal with highly nonlinear problems (Kirkpatrick *et al.* 1983), as an extension of a Monte-Carlo importance-sampling technique developed in 1953 for chemical physics problems. In 1984 (Geman and Geman 1984), it was established that SA possessed a proof that, by carefully controlling the rates of cooling of temperatures, it could statistically find the best minimum, e.g., the lowest valley of our example above. This was good news for people trying to solve hard problems which could not be solved by other algorithms. The bad news was that the guarantee was only good if they were willing to run SA forever. In 1987, a method of fast annealing (FA) was developed (Szu and Hartley 1987), which permitted lowering the temperature exponentially faster, thereby statistically guaranteeing that the minimum could be found in some finite time. However, that time still could be quite long. Shortly thereafter, Very Fast Simulated Reannealing (VFSR) was developed in 1987 (Ingber 1989), now called Adaptive Simulated Annealing (ASA), which is exponentially faster than FA.

ASA has been applied to many problems by many people in many disciplines (Ingber 1993b, 1996a, Wofsey 1993). The feedback of many users regularly scrutinizing the source code ensures its soundness as it becomes more flexible and powerful.

4.4.2 Multiple Local Minima

Our criteria for the global minimum of our cost function is minus the largest profit over a selected training data set (or in some cases, this value divided by the maximum drawdown). However, in many cases this may not give us the best set of parameters to find profitable trading in test sets or in real-time trading. Other considerations such as the total number of trades developed by the global minimum versus other close local minima may be relevant. For example, if the global minimum has just a few trades, while some nearby local minima (in terms of the value of the cost function) have many trades and was profitable in spite of our slippage factors, then the scenario with more trades might be more statistically dependable to deliver profits across testing and real-time data sets.

Therefore, for the outer-shell global optimization of training sets, we have used an ASA OPTION, MULTI_MIN, which saves a user-defined number of closest local minima within a user-defined resolution of the parameters. We then examine these results under several testing sets.

5 Trading System

5.1 Use of CMI

As the CMI formalism carries the relevant information regarding the prices dynamics, we have used it as a signal generator for an automated trading system for S&P futures.

While currently we are integrating fast-response CMI signals into the trading model, next we discuss averaged CMI signals characterizing longer time scales.

Based on a previous work (Ingber 1996c) applied to daily closing data, the overall structure of the trading system consists in 2 layers, as follows: We first construct the "short-time" Lagrangian function in the Itô representation (with the notation introduced in Section 3.3)

$$L(i|i-1) = \frac{1}{2\sigma^2 F_{i-1}^{2x}} \left(\frac{dF_i}{dt} - f^F \right)^2 \tag{14}$$

with i the post-point index, corresponding to the one factor price model

$$dF = f^F dt + \sigma F^x dz(t), \tag{15}$$

where f^F and $\sigma > 0$ are taken to be constants, $F(t)$ is the S&P future price, and dz is the standard Gaussian noise with zero mean and unit standard deviation. We perform a global, maximum likelihood fit to the whole set of price data using ASA. This procedure produces the optimization parameters $\{x, f^F\}$ that are used to generate the CMI. One computational approach was to fix the diffusion multiplier σ to 1 during training for convenience, but used as free parameters in the adaptive testing and real-time fits. Another approach was to fix the scale of the volatility, using an improved model,

$$dF = f^F dt + \sigma \left(\frac{F}{< F >} \right)^x dz(t), \tag{16}$$

where σ now is calculated as the standard deviation of the price increments $\Delta F/dt^{1/2}$, and $< F >$ is just the average of the prices.

As already remarked, to enhance the CMI sensitivity and response time to local variations (across a certain window size) in the distribution of price increments, the momenta are generated applying an adaptive procedure, i.e., after each new data reading another set of $\{f^F, \sigma\}$ parameters are calculated for the last window of data, with the exponent x – a contextual indicator of the noise statistics – fixed to the value obtained from the global fit.

The CMI computed in this manner are fed into the outer shell of the trading system, where an AI-type optimization of the trading rules is executed, using ASA once again.

The trading rules are a collection of logical conditions among the CMI, prices and optimization parameters that could be window sizes, time resolutions, or trigger thresholds. Based on the relationships between CMI and optimization parameters, a trading decision is made. The cost function in the outer shell is either the overall equity or the risk-adjusted profit (essentially the return). The inner and outer shell optimizations are coupled through some of the optimization parameters (e.g., time resolution of the data, window sizes), which justifies the recursive nature of the optimization.

Next, we describe in more details the concrete implementation of this system.

5.2 Data Processing

The CMI formalism is general and by construction permits us to treat multivariate coupled markets. In certain conditions (e.g., shorter time scales of data), and also due to superior scalability across different markets, it is desirable to have a trading system for a single instrument, in our case the S&P futures contracts that are traded electronically on Chicago Mercantile Exchange (CME). The focus of our system was intra-day trading, at time scales of data used in generating the buy/sell signals from 10 to 60 secs. In particular, we here give some results obtained when using data having a time resolution Δt of 55 secs (the time between consecutive data elements is 55 secs). This particular choice of time resolution reflects the set of optimization parameters that have been applied in actual trading.

It is important to remark that a data point in our model does not necessarily mean an actual tick datum. For some trading time scales and for noise reduction purposes, data is pre-processed into sampling bins of length Δt using either a standard averaging procedure or spectral filtering (e.g., wavelets, Fourier) of the tick data. Alternatively, the data can be defined in block bins that contain disjoint sets of averaged tick data, or in overlapping bins of widths Δt that update at every $\Delta t' < \Delta t$, such that an effective resolution $\Delta t'$ shorter than the width of the sampling bin is obtained. We present here work in which we have used disjoint block bins and a standard average of the tick data with time stamps falling within the bin width.

In Figures 1 and 2 we present examples of S&P futures data sampled with 55 secs resolution. We remark that there are several time scales – from minutes to one hour – at which an automated trading system might extract profits.

Figure 1 illustrates that the profitable regions are prominent even for data representing a relatively flat market period. I.e., June 20 shows an uptrend region of about 1 hour 20 minutes and several short and long trading domains between 10 minutes and 20 minutes.

Figure 2 illustrates the sustained short trading region of 1.5 hours and several shorter long and short trading regions of about 10–20 minutes.

In both situations, there are a larger number of opportunities at time

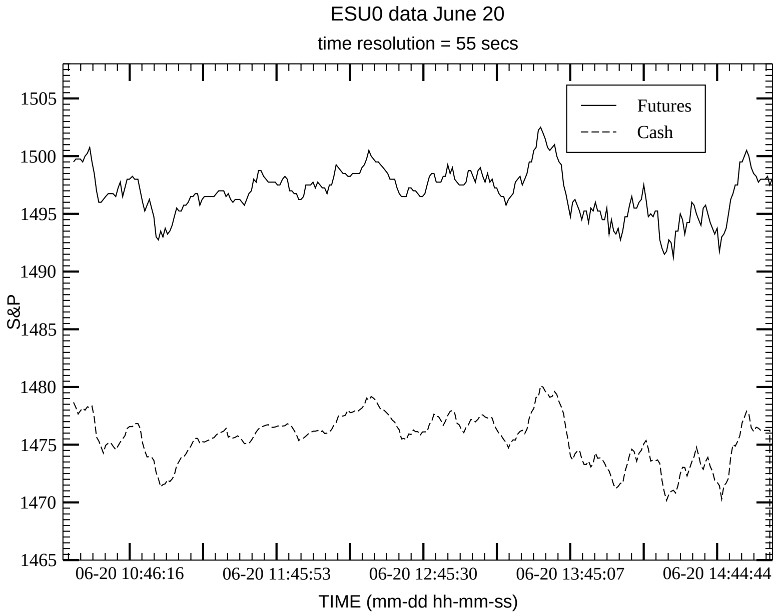

Figure 1. Futures and cash data, contract ESU0 June 20: (*solid line*) – futures; (*dashed line*) – cash.

resolutions smaller than 5 minutes.

The time scale at which we sample the data for trading is itself a parameter that is extracted from the optimization of the trading rules and of the Lagrangian cost function Eq. (14). This is one of the coupling parameters between the inner- and the outer-shell optimizations.

5.3 Inner-Shell Optimization

A cycle of optimization runs has three parts, training and testing, and finally real-time use – a variant of testing. Training consists in choosing a data set and performing the recursive optimization, which produces optimization parameters for trading. In our case there are six parameters: the time resolution Δt of price data, the length of

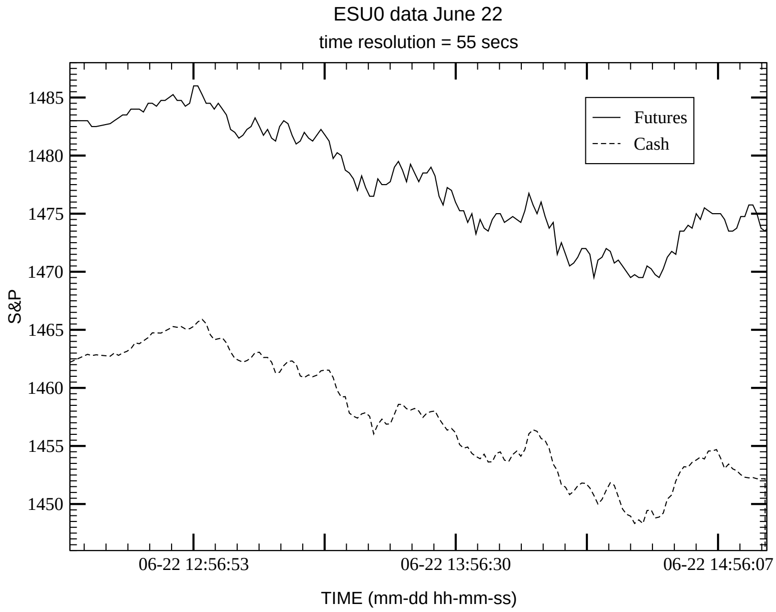

Figure 2. Futures and cash data, contract ESU0 June 22: (*solid line*) – futures; (*dashed line*) – cash.

window W used in the local fitting procedures and in computation of moving averages of trading signals, the drift f^F, volatility coefficient σ and exponent x from Eq. (15), and a multiplicative factor M necessary for the trading rules module, as discussed below.

The optimization parameters computed from the training set are applied then to various test sets and final profit/loss analyses are produced. Based on these, the best set of optimization parameters are chosen to be applied in real-time trading runs. We remark once again that a single training data set could support more than one profitable sets of parameters and can be a function of the trader's interest and the specific market dynamics targeted (e.g., short/long time scales). The optimization parameters corresponding to the global minimum in the training session may not necessarily represent the parameters that led to robust profits across real-time data.

The training optimization occurs in two inter-related stages. An inner-shell maximum likelihood optimization over all training data is performed. The cost function that is fitted to data is the effective action constructed from the Lagrangian Eq. (14) including the pre-factors coming from the measure element in the expression of the short-time probability distribution Eq. (11). This is based on the fact (Langouche *et al.* 1982) that in the context of Gaussian multiplicative stochastic noise, the macroscopic transition probability $P(F,t|F',t')$ to start with the price $F'(=F_{i-1})$ at $t'(=t_{i-1})$ and reach the price $F(=F_i)$ at $t(=t_i)y$ is determined by the short-time Lagrangian Eq. (14),

$$P(F,t|F',t') = \frac{1}{(2\pi\sigma^2 F_{i-1}^{2x} dt_i)^{1/2}} \times \exp\left(-\sum_{i=1}^{N} L(i|i-1)dt_i\right), \tag{17}$$

with $dt_i = t_i - t_{i-1}$. Recall that the main assumption of our model is that price increments (or the logarithm of price ratios, depending on which variables are considered independent) could be described by a system of coupled stochastic, non-linear equations as in Eq. (9a). These equations are deceptively simple in structure, yet depending on the functional form of the drift coefficients and the multiplicative noise, they could describe a variety of interactions between financial instruments in various market conditions (e.g., constant elasticity of variance model (Cox and Ross 1976), stochastic volatility models, and so on). In particular, this type of models include the case of Black-Scholes price dynamics ($x = 1$).

In the system presented here, we have applied the model from Eq. (15). The fitted parameters were the drift coefficient f^F and the exponent x. In the case of a coupled futures and cash system, besides the corresponding values of f^F and x for the cash index, another parameter, the correlation coefficient ρ as introduced in Eq. (9a), must be considered.

5.4 Trading Rules (Outer-Shell) Recursive Optimization

In the second part of the training optimization, we calculate the CMI and execute trades as required by a selected set of trading rules based on CMI values, price data or combinations of both indicators.

Recall that three external shell optimization parameters are defined: the time resolution Δt of the data expressed as the time interval between consecutive data points, the window length W (in number of time epochs or data points) used in the adaptive calculation of CMI, and a numerical coefficient M that scales the momentum uncertainty discussed below.

At each moment a local refit of f^F and σ over data in the local window W is executed, moving the window M across the training data set and using the zeroth order optimization parameters f^F and x resulting from the inner-shell optimization as a first guess. It was found that a faster quasi-local code is sufficient for computational purposes for these adaptive updates. In more complicated models, ASA can be successfully applied recursively, although in real-time trading the response time of the system is a major factor that requires attention.

All expressions that follow can be generalized to coupled systems in the manner described in Section 3. Here we use the one factor nonlinear model given by Eq. (15). At each time epoch we calculate the following momentum related quantities:

$$\Pi^F = \frac{1}{\sigma^2 F^{2x}} \left(\frac{dF}{dt} - f^F \right) ,$$

$$\Pi_0^F = -\frac{f^F}{\sigma^2 F^{2x}},$$

$$\Delta\Pi^F = < (\Pi^F - < \Pi^F >)^2 >^{1/2} = \frac{1}{\sigma F^x \sqrt{dt}}, \qquad (18)$$

where we have used $< \Pi^F >= 0$ as implied by Eqs. (15) and (14).

In the previous expressions, Π^F is the CMI, Π_0^F is the neutral line or the momentum of a zero change in prices, and $\Delta\Pi^F$ is the uncertainty of momentum. The last quantity reflects the Heisenberg principle, as derived from Eq. (15) by calculating

$$\Delta F \equiv < (dF - < dF >)^2 >^{1/2} = \sigma F^x \sqrt{dt},$$
$$\Delta\Pi^F \, \Delta F \geq 1, \tag{19}$$

where all expectations are in terms of the exact noise distribution, and the calculation implies the Itô approximation (equivalent to considering non-anticipative functions). Various moving averages of these momentum signals are also constructed. Other dynamical quantities, as the Hamiltonian, could be used as well. (By analogy to the energy concept, we found that the Hamiltonian carries information regarding the overall trend of the market, giving another useful measure of price volatility.)

Regarding the practical implementation of the previous relations for trading, some comments are necessary. In terms of discretization, if the CMI are calculated at epoch i, then $dF_i = F_i - F_{i-1}$, $dt_i = t_i - t_{i-1} = \Delta t$, and all prefactors are computed at moment $i-1$ by the Itô prescription (e.g., $\sigma F^x = \sigma F^x_{i-1}$). The momentum uncertainty band $\Delta\Pi^F$ can be calculated from the discretized theoretical value Eq. (18), or by computing the estimator of the standard deviation from the actual time series of Π^F.

There are also two ways of calculating averages over CMI values: One way is to use the set of local optimization parameters $\{f^F, \sigma\}$ obtained from the local fit procedure in the current window W for all CMI data within that window (local-model average). The second way is to calculate each CMI in the current local window W with another set $\{f^F, \sigma\}$ obtained from a previous local fit window measured from the CMI data backwards W points (multiple-models averaged, as each CMI corresponds to a different model in terms of the fitting parameters $\{f^F, \sigma\}$).

The last observation is that the neutral line divides all CMI in two classes: long signals, when $\Pi^F > \Pi_0^F$, as any CMI satisfying this condition indicates a positive price change, and short signals when $\Pi^F < \Pi_0^F$, which reflects a negative price change.

After the CMI are calculated, based on their meaning as statistical momentum indicators, trades are executed following a relatively simple model: Entry in and exit from a long (short) trade points are defined as points where the value of CMIs is greater (smaller) than a certain fraction of the uncertainty band $M\,\Delta\Pi^F$ ($-M\,\Delta\Pi^F$), where M is the multiplicative factor mentioned in the beginning of this subsection. This is a choice of a symmetric trading rule, as M is the same for long and short trading signals, which is suitable for volatile markets without a sustained trend, yet without diminishing too severely profits in a strictly bull or bear region.

Inside the momentum uncertainty band, one could define rules to stay in a previously open trade, or exit immediately, because by its nature the momentum uncertainty band implies that the probabilities of price movements in either direction (up or down) are balanced. From another perspective, this type of trading rule exploits the relaxation time of a strong market advance or decline, until a trend reversal occurs or it becomes more probable.

Other sets of trading rules are certainly possible, by utilizing not only the current values of the momenta indicators, but also their local-model or multiple-models averages. A trading rule based on the maximum distance between the current CMI data Π_i^F and the neutral line Π_0^F shows faster response to markets evolution and may be more suitable to automatic trading in certain conditions.

Stepping through the trading decisions each trading day of the training set determined the profit/loss of the training set as a single value of the outer-sell cost function. As ASA importance-sampled the outer-shell parameter space $\{\Delta t, W, M\}$, these parameters are fed into the inner shell, and a new inner-shell recursive optimization cycle begins. The final values for the optimization parameters in the training set are fixed when the largest net profit (calculated from the total equity by subtracting the transactions costs defined by the slippage factor) is realized. In practice, we have collected optimization parameters from multiple local minima that are near the global minimum (the outer-shell cost function is defined with the sign reversed) of the training set.

The values of the optimization parameters $\{\Delta t, W, M, f^F, \sigma, x\}$ resulting from a training cycle are then applied to out-of-sample test sets. During the test run, the drift coefficient f^F and the volatility coefficient σ are refitted adaptively as described previously. All other parameters are fixed. We have mentioned that the optimization parameters corresponding to the highest profit in the training set may not be the sufficiently robust across test sets. Then, for all test sets, we have tested optimization parameters related to the multiple minima (i.e., the global maximum profit, the second best profit, and so on) resulting from the training set.

We performed a bootstrap-type reversal of the training-test sets (repeating the training runs procedures using one of the test sets, including the previous training set in the new batch of test sets), followed by a selection of the best parameters across all data sets. This is necessary to increase the chances of successful trading sessions in real-time.

6 Results

6.1 Alternative Algorithms

In the previous sections we noted that there are different combinations of methods of processing data, methods of computing the CMI and various sets of trading rules that need to be tested – at least in a sampling manner – before launching trading runs in real-time:

1. Data can be preprocessed in block or overlapping bins, or forecasted data derived from the most probable transition path (Dekker 1980) could be used as in one of our most recent models.
2. Exponential smoothing, wavelets or Fourier decomposition can be applied for statistical processing. We presently favor exponential moving averages.
3. The CMI can be calculated using averaged data or directly with tick data, although the optimization parameters were fitted from preprocessed (averaged) price data.
4. The trading rules can be based on current signals (no average is performed over the signal themselves), on various averages of the CMI trading signals, on various combination of CMI data (momenta, neutral line, uncertainty band), on symmetric or asymmetric trading rules, or on mixed price-CMI trading signals.
5. Different models (one and two-factors coupled) can be applied to the same market instrument, e.g., to define complementary indicators.

The selection process evidently must consider many specific economic factors (e.g., liquidity of a given market), besides all other physical, mathematical and technical considerations. In the work presented here, as we tested our system and using previous experience, we focused toward S&P500 futures electronic trading, using block processed data, and symmetric, local-model and multiple-models trading rules based on CMI neutral line and stay-in conditions.

6.2 Trading System Design

The design of a successful electronic trading system is complex as it must incorporate several aspects of a trader's actions that sometimes are difficult to translate into computer code. Three important features that must be implemented are factoring in the transactions costs, devising money management techniques, and coping with execution deficiencies.

Generally, most trading costs can be included under the “slippage factor,” although this could easily lead to poor estimates. Given that the margin of profits from exploiting market inefficiencies are thin, a high slippage factor can easily result in a non-profitable trading system. In our situation, for testing purposes we used a $35 slippage factor per buy & sell order, a value we believe is rather high for an electronic trading environment, although it represents less than three ticks of a mini-S&P futures contract. (The mini-S&P is the S&P futures contract that is traded electronically on CME.) This higher value was chosen to protect ourselves against the bid-ask spread, as our trigger price (at what price the CMI was generated) and execution price (at what price a trade signaled by a CMI was executed) were taken to be equal to the trading price. (We have changed this aspect of our algorithm in later models.) The slippage is also strongly influenced by the time resolution of the data. Although the slippage is linked to bid-ask spreads and markets volatility in various formulas (Kaufman 1998), the best estimate is obtained from experience and actual trading.

Money management was introduced in terms of a trailing stop condition that is a function of the price volatility, and a stop-loss threshold that we fixed by experiment to a multiple of the mini-S&P contract value ($200). It is tempting to tighten the trailing stop or to work with a small stop-loss value, yet we found – as otherwise expected – that higher losses occurred as the signals generated by our stochastic model were bypassed.

Regarding the execution process, we have to account for the response of the system to various execution conditions in the interaction with the electronic exchange: partial fills, rejections, uptick rule (for equity trading), and so on. Except for some special conditions, all these steps must be automated.

6.3 Some Explicit Results

Typical CMI data in Figures 3 and 4 (obtained from real-time trading after a full cycle of training-testing was performed) are related to the price data in Figures 1 and 2. We have plotted the fastest (55 secs apart) CMI values Π^F, the neutral line Π^F_0 and the uncertainty band $\Delta\Pi^F$. All CMI data were produced using the optimization parameters set $\{55\mathrm{secs}, 88\mathrm{epochs}, 0.15\}$ of the second-best net profit obtained with a training set based on the March data of the ESM0 contract (mini-S&P June 2000 contract). We recall the meaning of the optimization parameters from 5.4: the first factor is the frequency of CMI signals (or time-step between consecutive CMIs), the second parameter is the width in time-step units of the time-window used for local statistics, and the third parameter is the scaling factor of the momentum uncertainty.

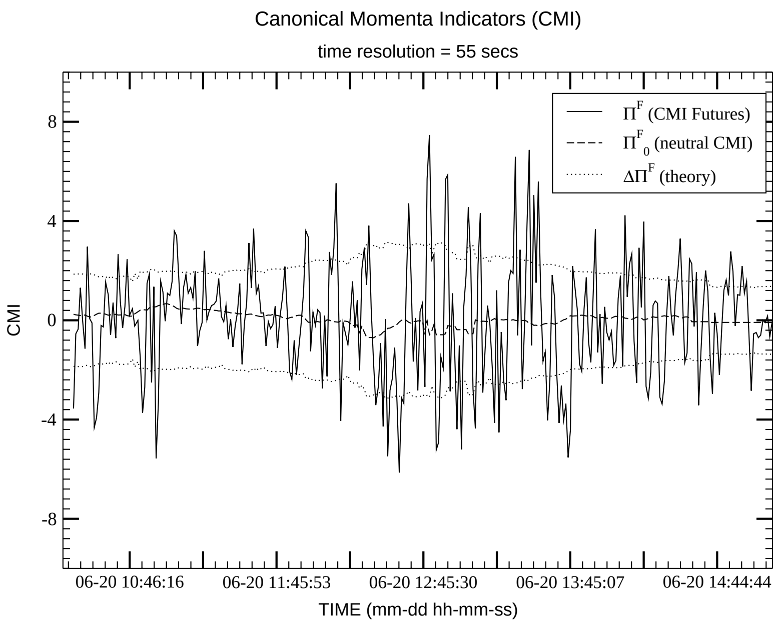

Figure 3. CMI data, real-time trading June 20: (*solid line*) – CMI; (*dashed line*) – neutral line; (*dotted line*) – uncertainty band.

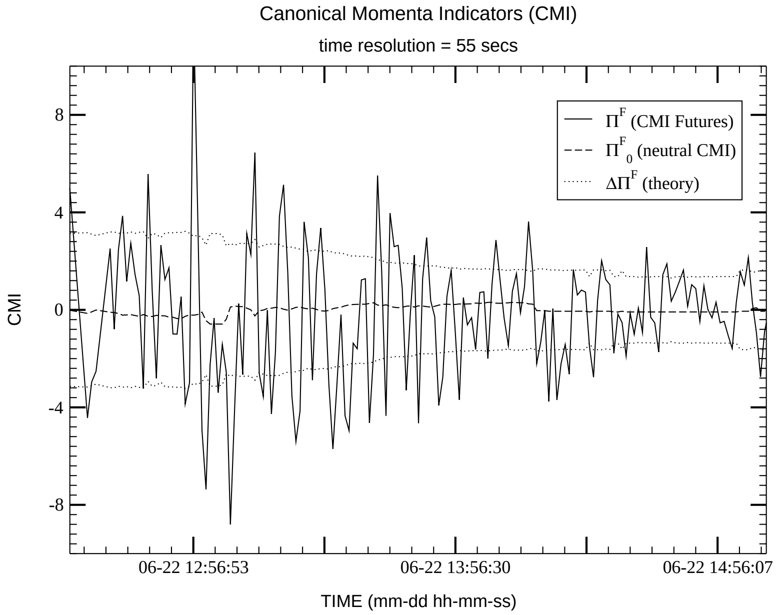

Figure 4. CMI data, real-time trading, June 22: (*solid line*) – CMI; (*dashed line*) – neutral line; (*dotted line*) – uncertainty band.

Although the CMIs exhibit an inherently ragged nature and oscillate around a zero mean value within the uncertainty band – the width of which is decreasing with increasing price volatility, as the uncertainty principle would also indicate – time scales at which the CMI average or some persistence time are not balanced about the neutral line.

These characteristics, which we try to exploit in our system, are better depicted in Figures 5 and 6.

One set of trading signals, the local-model average of the neutral line $< \Pi_0^F >$ and the uncertainty band multiplied by the optimization factor $M = 0.15$, and centered around the theoretical zero mean of the CMI, is represented versus time. Note entry points in a short trading position ($< \Pi_0^F > > M\, \Delta\Pi^F$) at around 10:41 (Figure 5

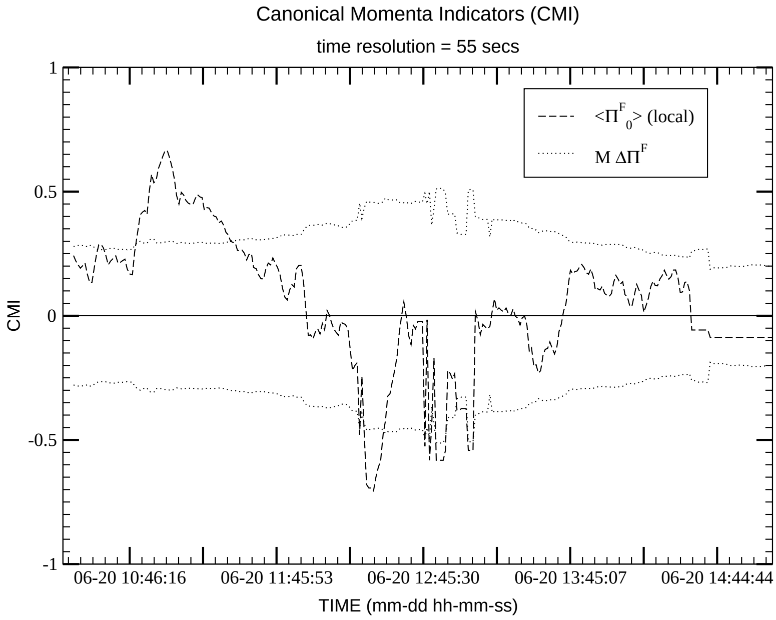

Figure 5. CMI trading signals, real-time trading June 20: (*dashed line*) – local-model average of the neutral line; (*dotted line*) – uncertainty band multiplied by the optimization parameter $M = 0.15$.

in conjunction with S&P data in Figure 1) with a possible exit at 11:21 (or later), and a first long entry ($< \Pi_0^F > < -M\ \Delta\Pi^F$) at 12:15. After 14:35, a stay long region appears ($< \Pi_0^F > < 0$), which indicates correctly the price movement in Figure 1.

In Figure 6 corresponding to June 22 price data from Figure 2, a first long signal is generated at around 12:56 and a first short signal is generated at 14:16 that reflects the long downtrend region in Figure 2. Due to the averaging process, a time lag is introduced, reflected by the long signal at 12:56 in Figure 4, related to a past upward trend seen in Figure 2; yet the neutral line relaxes rather rapidly (given the 55-second time resolution and the window of 88 $\approx$ 1.5 hour) toward the uncertainty band. A judicious choice of trading rules, or avoiding standard averaging methods, helps in controlling this lag problem.

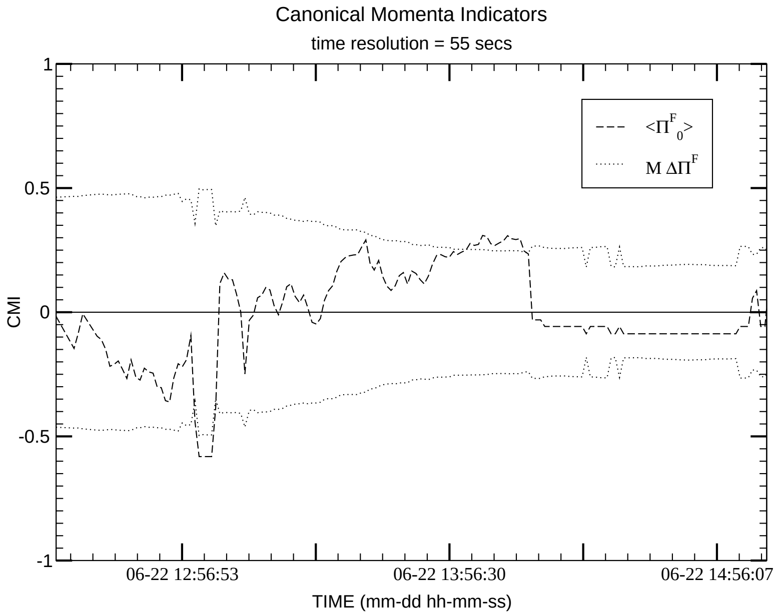

Figure 6. CMI trading signals, real-time trading June 22: (*dashed line*) – local-model average of the neutral line; (*dotted line*) – uncertainty band multiplied by the optimization parameter $M = 0.15$.

Recall that the trading rules presented are symmetric (the long and short entry/exit signals are controlled by the same M factor), and we apply a stay-long condition if the neutral-line is below the average momentum $< \Pi^F >= 0$ and stay-short if $< \Pi^F_0 >> 0$. The drift f^F and volatility coefficient σ are refitted adaptively and the exponent x is fixed to the value obtained in the training set. Typical values are $f^F \in \pm[0.003 : 0.05]$, $x \in \pm[0.01 : 0.03]$. During the local fit, due to the shorter time scale involved, the drift may increase by a factor of ten, and $\sigma \in [0.01 : 1.2]$.

We note that the most robust optimization factors – in terms of maximum cumulative profit resulted for all test sets – do not correspond to the maximum profit in the training sets: For the local-model rules, the optimum parameters are $\{55, 88, 0.15\}$, and for the multiple models rules the optimum set is $\{45, 72, 0.2\}$, both realized by a four-days training set from the March 2000 mini-S&P contract (Ingber and Mondescu 2001).

Other observations are that, for the data presented here, the multiple-models averages trading rules consistently performed better and are more robust than the local-model averages trading rules. The number of trades is similar, varying between 15 and 35 (eliminating cumulative values smaller than 10 trades), and the time scale of the local fit is rather long in the 30 minutes to 1.5 hour range. In the current set-up, this extended time scale implies that is advisable to deploy this system as a trader-assisted tool.

An important factor is the average length of the trades. For the type of rules presented in this work, this length is of several minutes, up to one hour, as the time scale of the local fit window mentioned above suggested.

Related to the length of a trade is the length of a winning long/short trade in comparison to a losing long/short trade. Our experience indicates that a ratio of 2:1 between the length of a winning trade and the length of a losing trade is desirable for a reliable trading system. Here, using the local-model trading rules seems to offer an advantage, although this is not as clear as one would expect. More details regarding the data and results obtained with the trading system are given in our earlier work (Ingber and Mondescu 2001).

7 Conclusions

7.1 Main Features

The main stages of building and testing this system are summarized in the followiong lines:

1. We developed a multivariate, nonlinear statistical mechanics

model of S&P futures and cash markets, based on a system of coupled stochastic differential equations.

2. We constructed a two-stage, recursive optimization procedure using methods of ASA global optimization: An inner-shell extracts the characteristics of the stochastic price distribution and an outer-shell generates the technical indicators and optimize the trading rules.
3. We trained the system on different sets of data and retained the multiple minima generated (corresponding to the global maximum net profit realized and the neighboring profit maxima).
4. We tested the system on out-of-sample data sets, searching for most robust optimization parameters to be used in real-time trading. Robustness was estimated by the cumulative profit/loss across diverse test sets, and by testing the system against a bootstrap-type reversal of training-testing sets in the optimization cycle.

   Modeling the market as a dynamical physical system makes possible a direct representation of empirical notions as market momentum in terms of CMI derived naturally from our theoretical model. We have shown that other physical concepts as the uncertainty principle may lead to quantitative signals (the momentum uncertainty band $\Delta\Pi^F$) that captures other aspects of market dynamics and which can be used in real-time trading.
5. We presented and discussed the main aspects of developing an internet-based interface (API) for connecting a proprietary trading system to an exchange.

7.2 Summary

We have presented an internet-enabled trading system with its two components: the connection API and the computational trading engine.

The trading engine is composed of an outer-shell trading-rule model and an inner-shell nonlinear stochastic dynamic model of the market of interest, S&P500. The inner-shell is developed adhering to the mathematical physics of multivariate nonlinear statistical mechanics, from which we develop indicators for the trading-rule model, i.e., canonical momenta indicators (CMI). We have found that keeping our model faithful to the underlying mathematical physics is not a limiting constraint on profitability of our system; quite the contrary.

An important result of our work is that the ideas for our algorithms, and the proper use of the mathematical physics faithful to these algorithms, must be supplemented by many practical considerations en route to developing a profitable trading system. For example, since there is a subset of parameters, e.g., time resolution parameters, shared by the inner- and outer-shell models, recursive optimization is used to get the best fits to data, as well as developing multiple minima with approximate similar profitability. The multiple minima often have additional features requiring consideration for real-time trading, e.g., more trades per day increasing robustness of the system, and so on. The nonlinear stochastic nature of our data required a robust global optimization algorithm. The output of these parameters from these training sets were then applied to testing sets on out-of-sample data. The best models and parameters were then used in real-time by traders, further testing the models as a precursor to eventual deployment in automated electronic trading.

We have used methods of statistical mechanics to develop our inner-shell model of market dynamics and a heuristic AI type model for our outer-shell trading-rule model, but there are many other candidate (quasi-)global algorithms for developing a cost function that can be used to fit parameters to data, e.g., neural nets, fractal scaling models, and so on. To perform our fits to data, we selected an algorithm, Adaptive Simulated Annealing (ASA), that we were familiar with, but there are several other candidate algorithms that likely would suffice, e.g., genetic algorithms, tabu search, and so on.

We have shown that a minimal set of trading signals (the CMI, the neutral line representing the momentum of the trend of a given time window of data, and the momentum uncertainty band) can generate a rich and robust set of trading rules that identify profitable domains of trading at various time scales. This is a confirmation of the hypothesis that markets are not efficient, as noted in other studies (Brock *et al.*1992, Ingber 1984, 1996c).

7.3 Future Directions

Although this chapter focused on trading of a single instrument, the futures S&P 500, the code we have developed can accommodate trading on multiple markets. For example, in the case of tick-resolution coupled cash and futures markets, which was previously prototyped for inter-day trading (Ingber 1996b,c),the utility of CMI stems from three directions:

1. The inner-shell fitting process requires a global optimization of all parameters in both futures and cash markets.
2. The CMI for futures contain, by our Lagrangian construction, the coupling with the cash market through the off-diagonal correlation terms of the metric tensor. The correlation between the futures and cash markets is explicitly present in all futures variables.
3. The CMI of both markets can be used as complimentary technical indicators for trading in futures market.

Several near term future directions are of interest:

- finalizing the production-level order execution API,
- orienting the system toward shorter trading time scales (10-30 secs) more suitable for electronic trading,
- introducing fast response “averaging” methods and time scale identifiers (exponential smoothing, wavelets decomposition),
- identifying mini-crashes points using renormalization group techniques,
- investigating the use of CMI in pattern-recognition based trading rules,
- exploring the use of forecasted data evaluated from most probable transition path formalism.

7.4 Standard Disclaimer

We must emphasize that there are no claims that all results are positive or that the present system is a safe source of riskless profits. There as many negative results as positive, and a lot of work is necessary to extract meaningful information.

Acknowledgments

We thank Donald Wilson for his financial support. We thank K.S. Balasubramaniam and Colleen Chen for their programming support and participation in formulating parts of our trading system. Data was extracted from the DRW Reuters feed.

References

Archipelago (2001), http://www.tradearca.com .

Azoff, E. (1994), *Neural Network Time Series Forecasting of Financial Markets*, Wiley & Sons, New York, NY.

Bachelier, L. (1900), "Théorie de la spéculation," *Annales de l'Ecole Normale Supérieure*, vol. 17, pp. 21-86.

Barabino, G.P., Barabino, G.S., Bianco, B., and Marchesi, M. (1980), "A study on the performances of simplex methods for function minimization," *Proc. IEEE Int. Conf. Circuits and Computers*, pp. 1150-1153.

Brock, W., Lakonishok, J., and LeBaron, B. (1992), " Simple technical trading rules and the stochastic properties of stock returns," *J. Finance*, vol. 47, no. 5, pp. 1731-1763.

Brown, P., Kleidon, A.W., and Marsh, T.A. (1983), “New evidence on the nature of size-related anomalies in stock prices,” *J. Fin. Econ.*, vol. 12, pp. 33-56.

Burghardt, G. (2001), “Whassup?,” *Futures Industry Magazine*, February/March.

Cheng, K.S. (1972), “Quantization of a general dynamical system by Feynman’s path integration formulation,” *J. Math. Phys.*, vol. 13, pp. 1723-1726.

CME Market Data API (2001), http://www.cme.com/electronic/mdapi .

Cohen, B. (1994), “Training synaptic delays in a recurrent neural network,” M.Sc. Thesis, Tel-Aviv University, Tel-Aviv, Israel (unpublished).

Cox, J.C. and Ross, S.A. (1976), “The valuation of options for alternative stochastic processes,” *J. Fin. Econ.*, vol. 3, pp. 145-166.

Cozzio-Buëler, R.A. (1995), *The design of neural networks using a prior knowledge*, Ph.D. dissertation, Swiss Fed. Inst. Technol., Zurich, Switzerland.

Dekker, H. (1979), “Functional integration and the Onsager-Machlup Lagrangian for continuous Markov processes in Riemannian geometries,” *Phys. Rev. A*, vol. 19, pp. 2102-2111.

Dekker, H. (1980), “On the most probable transition path of a general diffusion process,” *Phys. Lett. A*, vol. 80, pp. 99-101.

eSignal (Interactive Data Corporation) (2001), http://www.esignal.com .

Eurex Values API (2001), http://www.eurexchange.com/marketplace.

FIX Protocol Organization (2000), http://www.fixprotocol.org .

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.

Gardiner, C. (1983), *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, Springer-Verlag, Berlin, Germany.

Gately, E. (1996), *Neural Networks for Financial Forecasting*, Wiley & Sons, New York, NY.

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images," *IEEE Trans. Patt. Anal. Mac. Int.*, vol. 6, no. 6, pp. 721-741.

Graham, R. (1978), "Path-integral methods in nonequilibrium thermodynamics and statistics," in Garrido, L., Seglar, P., and Shepherd, P.J. (eds.), *Stochastic Processes in Nonequilibrium Systems,* Springer, New York, NY, pp. 82-138.

Haken, H. (1983), *Synergetics,* 3rd ed., Springer, New York.

Harrison, J.M. and Kreps, D. (1979), "Martingales and arbitrage in multiperiod securities markets," *J. Econ. Theory*, vol. 20, pp. 381-408.

Hull, J.C. (2000), *Options, Futures, and Other Derivatives*, 4th ed., Prentice Hall, Upper Saddle River, NJ.

Ilinsky, K. and Kalinin, G. (1997), "Black-Scholes equation from gauge theory of arbitrage," Technical Report, LANL (unpublished), http://xxx.lanl.gov/hep-th/9712034 .

Indiveri, G., Nateri, G., Raffo, L., and Caviglia, D. (1993), "A neural network architecture for defect detection through magnetic inspection," Report, University of Genova, Genova, Italy (unpublished).

Ingber, L. (1984), "Statistical mechanics of nonlinear nonequilibrium financial markets," *Math. Modelling*, vol. 5, no. 6, pp. 343-361, http://www.ingber.com/markets84_statmech.ps.gz .

Ingber, L. (1985), "Statistical mechanics of neocortical interactions: stability and duration of the 7+-2 rule of short-term-memory capacity," *Phys. Rev. A*, vol. 31, pp. 1183-1186, http://www.ingber.com/smni85_stm.ps.gz .

Ingber, L. (1989), "Very fast simulated re-annealing," *Math. Comput. Modelling*, vol. 12, no. 8, pp. 967-973, http://www.ingber.com/asa89_vfsr.ps.gz .

Ingber, L. (1991), "Statistical mechanics of neocortical interactions: a scaling paradigm applied to electroencephalography," *Phys. Rev. A*, vol. 44, no. 6, pp. 4017-4060, http://www.ingber.com/smni91_eeg.ps.gz .

Ingber, L. (1993a), "Adaptive Simulated Annealing (ASA)," Tech. Report "Global optimization C-code," Caltech Alumni Association, Pasadena, CA, http://www.ingber.com/#ASA-CODE .

Ingber, L. (1993b), "Simulated annealing: practice versus theory," *Math. Comput. Modelling*, vol. 18, no. 11, pp. 29-57, http://www.ingber.com/asa93_sapvt.ps.gz .

Ingber, L. (1996a), "Adaptive simulated annealing (ASA): lessons learned," *Control and Cybernetics*, vol. 25, no. 1, pp. 33-54, invited paper on "Simulated Annealing Applied to Combinatorial Optimization," http://www.ingber.com/asa96_lessons.ps.gz .

Ingber, L. (1996b), "Canonical momenta indicators of financial markets and neocortical EEG," in Amari, S.I., Xu, L., King, I., and Leung, K.-S. (eds.), *Progress in Neural Information Processing*, Springer, New York, pp. 777-784, http://www.ingber.com/markets96_momenta.ps.gz .

Ingber, L. (1996c), "Statistical mechanics of nonlinear nonequilibrium financial markets: applications to optimized trading," *Math. Computer Modelling*, vol. 23, no. 7, pp. 101-121, http://www.ingber.com/markets96_trading.ps.gz .

Ingber, L. (2000), "High-resolution path-integral development of financial options," *Physica A*, vol. 283, no. 3/4, pp. 529-558, http://www.ingber.com/markets00_highres.ps.gz .

Ingber, L. and Mondescu, R.P. (2001), "Optimization of trading physics models of markets," *IEEE Trans. on Neural Networks*, vol. 12, no. 4, pp. 776-790, http://www.ingber.com/markets01_optim_trading.pdf .

Ingber, L. and Rosen, B. (1993), "Genetic algorithms and very fast simulated reannealing: a comparison," *Oper. Res. Management Sci.*, vol. 33, no. 5, p. 523.

Ingber, L., Wehner, M.F., Jabbour, G.M., and Barnhill, T.M. (1991), "Application of statistical mechanics methodology to term-structure bond-pricing models," *Math. Comput. Modelling*, vol. 15, no. 11, pp. 77-98, http://www.ingber.com /markets91_interest.ps.gz .

Jensen, M.C. (1978), "Some anomalous evidence regarding market efficiency, an editorial introduction," *J. Finan. Econ.*, vol. 6, pp. 95-101.

Johansen, A., Sornette, D., and Ledoit, O. (1999), "Predicting financial crashes using discrete scale invariance," *J. Risk*, vol. 1, no. 4, pp. 5-32.

Kaufman, P.J. (1998), *Trading Systems and Methods*, 3rd ed., John Wiley & Sons, New York, NY.

Kirkpatrick, S., Gelatt Jr., C.D., and Vecchi, M. (1983), "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680.

Laloux, L., Cizeau, P., Bouchaud, J.-P., and Potters, M. (1999), "Noise dressing of financial correlation matrices," *Phys. Rev. Lett.*, vol. 83, pp. 1467-1470.

Langouche, F., Roekaerts, D., and Tirapegui, E. (1979), "Discretization problems of functional integrals in phase space," *Phys. Rev. D*, vol. 20, pp. 419-432.

Langouche, F., Roekaerts, D., and Tirapegui, E. (1980), "Short derivation of Feynman Lagrangian for general diffusion process," *J. Phys. A*, vol. 113, pp. 449-452.

Langouche, F., Roekaerts, D., and Tirapegui, E. (1982), *Functional Integration and Semiclassical Expansions*, Reidel, Dordrecht, The Netherlands.

Mandelbrot, B.B. (1971), "When can price be arbitraged efficiently? A limit to the validity of the random walk and martingale models," *Rev. Econ. Statist.*, vol. 53, pp. 225-236.

Mandelbrot, B.B. (1997), *Fractals and Scaling in Finance*, Springer-Verlag, New York, NY.

Mantegna, R.N. and Stanley, H.E. (1996), "Turbulence and financial markets," *Nature*, vol. 383, pp. 587-588.

Mayer, D.G., Pepper, P.M., Belward, J.A., Burrage, K. and Swain, A.J. (1996), "Simulated annealing – a robust optimization technique for fitting nonlinear regression models," *Proceedings "Modelling, Simulation and Optimization" Conference*, International Association of Science and Technology for Development (IASTED), 6-9 May, Gold Coast.

Merton, R.C. (1973), "An intertemporal capital asset pricing model," *Econometrica*, vol. 41, pp. 867-887.

Nelder, J.A. and Mead, R. (1964), "A simplex method for function minimization," *Computer J. (UK)*, vol. 7, pp. 308-313.

Oksendal, B. (1998), *Stochastic Differential Equations*, Springer, New York, NY.

Peters, E. (1991), *Chaos and Order in the Capital Markets*, Wiley & Sons, New York, NY.

Pliska, S.R. (1997), *Introduction to Mathematical Finance*, Blackwell, Oxford, UK.

Rogue Wave Software (2001), http://www.roguewave.com .

Sakata, S. and White, H. (1998), "High breakdown point conditional dispersion estimation with application to S&P 500 daily returns volatility," *Econometrica*, vol. 66, pp. 529-567.

Szu, H. and Hartley, R. (1987), "Fast simulated annealing," *Phys. Lett. A*, vol. 122, no. 3/4, pp. 157-162.

Taylor, S.J. (1982), "Tests of the random walk hypothesis against a price-trend hypothesis," *J. Finan. Quant. Anal.*, vol. 17, pp. 37-61.

Wofsey, M. (1993), "Technology: shortcut tests validity of complicated formulas," *The Wall Street Journal*, vol. 222, no. 60, p. B1.