STOCHASTIC PROCESSES AND CONTROL FOR JUMP-DIFFUSIONS*

FLOYD B. HANSON†

Abstract. An applied compact introductory survey of Markov stochastic processes and control in continuous time is presented. The presentation is in tutorial stages, beginning with deterministic dynamical systems for contrast and continuing on to perturbing the deterministic model with diffusions using Wiener processes. Then jump perturbations are added using simple Poisson processes constructing the theory of simple jump-diffusions. Next, marked-jump-diffusions are treated using compound Poisson processes to include random marked jump-amplitudes in parallel with the equivalent Poisson random measure formulation. Otherwise, the approach is quite applied, using basic principles with no abstractions beyond Poisson random measure. This treatment is suitable for those in classical applied mathematics, physical sciences, quantitative finance and engineering, but have trouble getting started with the abstract measure-theoretic literature. The approach here builds upon the treatment of continuous functions in the regular calculus and associated ordinary differential equations by adding non-smooth and jump discontinuities to the model. Finally, the stochastic optimal control of marked-jump-diffusions is developed, emphasizing the underlying assumptions. The survey concludes with applications in biology and finance, some of which are canonical, dimension reducible problems and others are genuine nonlinear problems.

Key words. Jump-diffusions, Wiener processes, Poisson processes, random jump amplitudes, stochastic differential equations, stochastic chain rules, stochastic optimal control

AMS subject classifications. 60G20, 93E20, 93E03

1. Introduction.

There are many important applications of stochastic processes and control in such areas as finance, biology and manufacturing. In various financial markets, for instance, the statistical evidence of jumps is given by Ball and Torous [9] for call option prices, Jarrow and Rosenfeld [38] for the capital asset pricing model, and Jorion [39] for foreign exchange as well as in the stock market. Lately, there has been improved development of nonparametric methods of testing for jumps related to quadratic variation and one example is the bipower variation method of Barndorff-Nielsen and Shephard [10]. In addition, Andersen, Benzoni and Lund [2], and Bates [11] similarly demonstrate that the most reasonable model of stock prices would include both stochastic-volatility and jump-diffusion.

While the financial markets are often modeled by the log-normal distribution using the diffusion process only, a more realistic model would have jumps in price, the distribution usually would be *skewed negatively* compared to the skewless normal and would have fatter tails than the exponentially smaller tails of the normal. Along with fatter tails the distribution consequently has higher peaks, so that the distribution is *leptokurtic*, meaning that the coefficient of kurtosis (the scaled fourth central moment) is greater than the normal value of three [25]. Another feature often missing in many models is the time-dependence of the parameters [28] and then there is the important special case of stochastic-volatility, often using the mean-reverting, square-root noise model of Heston [31] (see also [11, 35, 66, 29]).

There are many fine texts available that almost exclusively treat diffusion processes, such as Arnold [7], Schuss [57], Øksendal [51], Mikosch [50] and Steele [62].

*This work is supported in part by the National Science Foundation under DMS-02-07081 and predecessor grants. Any conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

† Department of Mathematics, Statistics, and Computer Science University of Illinois at Chicago, 851 S Morgan St.; m/c 249, Chicago, IL 60607–7045, USA (hanson@math.uic.edu).

The texts of Çinlar [17], and Snyder and Miller [61] are very useful primarily for jump processes, among other topics. The seminal work of K. Itô [36] is well-known for the foundations of the analysis and treatment of diffusion processes such that the term *Itô processes* refers solely to diffusion processes, but it is not so well-known that this work also includes jumps of Poisson processes making it an early work on jump-diffusions. Kushner and Dupuis's book [44] is nominally about numerical methods for stochastic control, but has substantial information about jump-diffusions, some of it found in the earlier works of Kushner [42, 43]. Although the books of Cont and Tankov [18] and Øksendal and Sulem [52] have the term *jump-diffusion* in the titles, the books emphasize more general and abstract types of processes called *Lévy processes*, that include jump-diffusions which can have finite jump-rates as discussed here, but also those which have infinite jump-rates. The approach is generally through measure theory, martingales and other abstractions. This seems to be necessary from the point of view of the practitioners for advanced analysis, transformations and the proving of theorems for stochastic problems. There are books that are even more abstract-oriented than those mentioned, such as the well-respected works of Applebaum [5], Gihman and Skorohod [20, 21], Jacod and Shiryaev [37], and Protter [53].

Colleagues and graduate students in many areas of the applied sciences find these abstract approaches to stochastic problems inaccessible or at least formidable, having been trained in classical analysis based upon regular and advanced calculus. Part of the difficultly lies in the current treatment of the calculus with an almost exclusive emphasis on functions that are too nice, e.g., continuous, smooth and deterministic functions, in a world where things can be discontinuous, non-smooth and random. Here, we take stochastic and random to mean the same thing, signifying a degree of uncertainty and involving probabilities. A *process* merely means some function of time and a *stochastic or random process* means a time-dependent random function or random trajectory.

The purpose of this survey is to build an applied stochastic calculus of stochastic processes based upon first principles, using modifications of the usual regular calculus to allow for jump-discontinuities, non-smoothness and randomness. One underlying goal is to be able to treat jump-diffusion processes and another goal is to control the dynamics of jump-diffusion processes for useful purposes. The difficulty of such an endeavor is that many of the needed applied results are difficult to find or missing in the existing literature, so have to be supplied new [25].

On the other hand, many of the recent results have come from the abstract approaches and some of these results can be found in the jump-diffusion texts listed above [18, 52] and other references, for example Runggaldier [55] or the applied to abstract bridge, last chapter of the author's forthcoming book [25, Chapter 13]. The applied and abstract approaches complement each other, in that the applied approaches supply *physical* intuition about the underlying stochastic problems and the abstract approaches provide the formalism for rigorous proofs.

In Section 2, the deterministic vector *ordinary differential equation (ODE)* model is considered as a base reference model against which to compare the *stochastic differential equation (SDE)*, technically stochastic ODE, models by treating the deterministic model in a similar way. In Section 3, the vector diffusion process is introduced by way of the mathematical *Wiener process* and is used to perturb the deterministic model with continuous stochastic noise. Also, the more fundamental stochastic integral equation is formulated, since it is more useful for the discretization of dynamic models along with defining mean square convergence conditions. The stochastic dif-

fusion chain rule is derived from a second order Taylor approximation. In addition, a formally exact solution for a scalar, linear diffusion SDE is found and a method for simulating the solution is given along with a sample code. In Section 4, a *simple Poisson process* is used to introduce jumps into the diffusion SDE forming a simple *jump-diffusion SDE*, preserving the Markov property of the model. The instantaneous property of the jumps allows the jump-diffusion change rule to be decomposed into continuous and jump components, while the calculation of the jump change is simplified by the zero-one jump law of the infinitesimal Poisson process. A corresponding scalar, linear example problem is formally solved and a simulations program is developed for it. In Section 5, the simple Poisson process is replaced by a *compound or marked Poisson process* allowing for random jump-amplitudes in place of non-random coefficients. The Poisson random measure formulation is also introduced as an equivalent alternative to the compound Poisson process since it is more convenient in many calculations. Several basic jump-amplitude distributions are given for modeling the underlying independent identically-distributed underlying random variables called marks. Again, a simple problem is formally solved and simulated. In the 6th section, the stochastic optimal control problem is transformed to a functional generalization of a *partial differential equation (PDE)*, called the *Bellman's PDE of stochastic dynamic programming* is formally derived in some detail to reveal the assumptions for later modifications. The *curse of dimensionality*, which arises for large-scale discretization in dynamic programming, is shown to have exponential complexity. In the 7th section, two classical canonical problems are presented that avoid the curse of dimensionality by reducing the dynamic programming problem to ODEs in time with a special template form for the state or spatial variables. These methods are the linear-quadratic, jump-diffusion problem and the optimal portfolio and consumption problem with power-law utilities. In addition, two non-canonical problems are introduced as examples of general, nonlinear models: optimal natural resource harvesting with price dynamics in a jump-diffusion environment and European option pricing in a nonlinear stochastic volatility, jump-diffusion environment.

2. Deterministic Differential Equation (DetDE). We start at a very basic level to exhibit the foundations in the regular calculus so at each incremental level it should be clear what modifications are needed. Let $\mathbf{X}(t) = [X_i(t)]_{n_x\times 1}$ be the vector state of a deterministic dynamical system of dimension n_x, where $\mathbf{f}(\mathbf{X}(t),t) = [f_i(\mathbf{X}(t),t)]_{n_x\times 1}$ is the *vector plant function* that determines the rate of growth of the system given by the following ODE or DetDE,

$$d\mathbf{X}(t) = \mathbf{f}(\mathbf{X}(t),t)dt, \quad 0 \le t \le t_f, \quad \mathbf{X}(0) = \mathbf{x}_0 \in \mathcal{D}_x \subset \mathbb{R}^{n_x}. \tag{2.1}$$

The plant function $\mathbf{f}(\mathbf{X}(t),t)$ is assumed to be an integrable function in t when the state trajectory $\mathbf{X}(t)$ is continuous.

Since the system is deterministic, the conditional infinitesimal expectation or mean and variance are trivially denoted by

$$\begin{aligned}\mathrm{E}[d\mathbf{X}(t)\,|\,\mathbf{X}(t)=\mathbf{x}] &= \mathbf{f}(\mathbf{x},t)dt,\\ \mathrm{Var}[d\mathbf{X}(t)\,|\,\mathbf{X}(t)=\mathbf{x}] &= 0_{n_x\times n_x},\end{aligned}$$

i.e., there is no variance. In addition, the jump in the state is

$$\mathrm{Jump}[\mathbf{X}](t) \equiv [\mathbf{X}](t) \equiv \mathbf{x}\left(t^+\right) - \mathbf{x}\left(t^-\right) = \mathbf{0}_{n_x\times 1},$$

i.e., none.

2.1. Deterministic Integral Equation. A better and alternative form of the model DetDE (2.1) is the corresponding deterministic integral equation (DetIE),

$$\mathbf{X}(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{X}(s), s)ds\,, \tag{2.2}$$

for both analytical and numerical purposes, in particular for higher order approximations. For instance, using the very simple *forward integration* or *Euler approximation*,

$$\mathbf{X}(t) \stackrel{\text{fwd}}{\simeq} \mathbf{x}_0 + \sum_{j=0}^{n} \mathbf{f}(\mathbf{X}_j, t_j)\Delta t_j\,, \tag{2.3}$$

where $\mathbf{X}_j = \mathbf{X}(t_j)$, $t_{j+1} = t_j + \Delta t_j$, $\Delta t_j > 0$, $t_0 \equiv 0$, $t_{n+1} = t$ and the mesh $\delta t_n = \max_{j=0:n}[\Delta t_j] \to 0^+$ as $n \to \infty$, guaranteeing convergence (here $j = 0 : n = 0, 1, \ldots, n$ is a MATLAB™ sequence construct). The crude approximation in (2.3) is also called the left-endpoint rectangular rule or the tangent-line approximation when deduced from the DetDE (2.1). There are many more accurate approximation rules that could be used, since any point t_j^* in $[t_j, t_j + \Delta t_j]$ could be used as an approximation point according to the integration theory of Riemann. However, the forward integration approximation is consistent with the notion of the Markov processes considered in the next section where the dependence is on the present events and not future events that need to be estimated along with the uncertainty involved.

2.2. Change of Variables and Deterministic Chain Rule. Often in solving differential or their corresponding integral equations, a change of variables may be useful. In changing the original state variable $\mathbf{X}(t)$ to a new variable $\mathbf{Y}(t)$, consider for generality the composite function form $\mathbf{Y}(t) = \mathbf{F}(\mathbf{X}(t), t)$ or $Y_i(t) = F_i(\mathbf{X}(t), t)$ for $i = 1 : n_x$ components. The corresponding chain rule becomes

$$\begin{aligned} d\mathbf{Y}(t) = &\quad \mathbf{F}_t(\mathbf{X}(t), t)dt + \left(\mathbf{X}^\top(t)\nabla_x\right)[\mathbf{F}](\mathbf{X}(t), t) \\ = &\quad \left(\mathbf{F}_t + \left(\mathbf{f}^\top\nabla_x\right)[\mathbf{F}]\right)(\mathbf{X}(t), t)dt\,, \end{aligned} \tag{2.4}$$

or by components,

$$dY_i(t) = \left(F_{i,t} + \sum_{j=1}^{n_x} f_j F_{i,x_j}\right)(\mathbf{X}(t), t)dt\,,$$

for $i = 1 : n_x$, where $F_{i,t} = \partial F_i/\partial t$ and $F_{i,x_j} = \partial F_i/\partial x_j$ are partial derivatives while $\mathbf{f}^\top$ denotes the transpose of a column vector into a row vector. For later reference, (2.4) represents the *dt-precision of changes in the system*.

2.3. Scalar, Linear DetDE Example. As an illustration of using a change of variables that transforms the differential equation to a form that can be readily integrated to obtain the solution, consider the $n_x = 1$ scalar, linear model,

$$dX(t) = A(t)X(t)dt, \quad X(0) = x_0 > 0\,, \tag{2.5}$$

when $A(t)$ is a given integrable scalar function. Let

$$Y(t) = F(X(t), t) = \ln(X(t))$$

be a logarithmic state transformation, with partial derivatives $F_t(x,t) = 0$ and $F_x(x,t) = 1/x$, assuming $x > 0$. Then, the transformed ODE by the chain rule (2.4) becomes state-independent,

$$dY(t) = \ln(X(t) + dX(t)) - \ln(X(t)) = A(t)dt\,,$$

so that immediate integration yields

$$Y(t) = Y(0) + \int_0^t A(s)ds\,.$$

Using the initial condition $Y(0) = \ln(x_0)$ and the inverse of the natural logarithm, our solution to the given initial value problem (2.5) is

$$X(t) = x_0 \exp\left(\int_0^t A(s)ds\right), \tag{2.6}$$

consistent with the exponential form of the solution expected for deterministic, linear ODEs. Note that the initial assumption that $x_0 > 0$ implies the solution positivity, $X(t) > 0$ for all $t \geq 0$.

Although the detail may seem overdone for this simple ODE problem, it is needed for reference later to show how much or how little the jump-diffusion adds to this reference deterministic solution (2.6).

3. Diffusion Stochastic Differential Equation (SDE). In order to include stochastic diffusion into the differential equation model, the mathematical representation of a diffusion random process called the *Wiener process* is introduced. A *process* merely denotes a function of time. The n_x-dimensional vector Wiener process,

$$\mathbf{W}(t) \in \mathcal{D}_x \subset \mathbb{R}^{n_w}\,,$$

and its differential expressed as an increment,

$$d\mathbf{W}(t) = \mathbf{W}(t + dt) - \mathbf{W}(t)\,, \tag{3.1}$$

are *normally distributed random processes*, with infinitesimal expectation

$$\mathrm{E}[d\mathbf{W}(t)] = \mathbf{0}_{n_w \times 1}\,,$$

infinitesimal component variance,

$$\mathrm{Var}[dW_i(t)] = dt$$

for $i = 1 : n_w$ and the infinitesimal covariance, as related to the *Wiener correlation matrix*, $R^{(w)}(t)$,

$$\begin{aligned} \mathrm{Cov}\left[d\mathbf{W}(t), d\mathbf{W}^\top(t)\right] &= \mathrm{Corr}\left[d\mathbf{W}(t), d\mathbf{W}^\top(t)\right] dt \\ &\equiv R^{(w)}(t)dt = \left[\rho^{(w)}_{i,j}(t)dt\right]_{n_w \times n_w}\,, \end{aligned} \tag{3.2}$$

where the diagonal part of the correlation coefficient satisfies $\rho^{(w)}_{i,i}(t) \equiv 1$ from the component variance. If there is no correlation between distinct components then $R^{(w)}(t) = I_{n_w}$, the identity of order n_w.

The Wiener process is continuous, so the jump is $\text{Jump}[d\mathbf{W}](t) = \mathbf{0}_{n_w \times 1}$, and by convention $\mathbf{W}(0) \equiv \mathbf{0}_{n_w \times 1}$ with probability one. The $\mathbf{W}(t)$ are Markov processes with no memory of the non-immediate past, i.e., the conditional probability is

$$\text{Prob}[W_i(t + \Delta t) \,|\, W_i(s), 0 \leq s \leq t] = \text{Prob}[W_i(t + \Delta t) \,|\, W_i(t)] \,,$$

for $i = 1 : n_w$. Although $W_i(t)$ is continuous, it is not smooth, which can be motivated by noting that $\text{Var}[dW_i(t)] = dt$, formally implying

$$\text{Var}[dW_i(t)/dt] = 1/dt \to +\infty$$

as $dt \to 0^+$ (see Hanson [25, Chapter 1] for an applied proper version or Mikosch [50, Chapter 1] for a stronger form of this condition or Steele [62, Chapter 5] for an even more precise condition.)

Let the resulting stochastic differential equation (SDE) for the diffusion process be

$$\begin{aligned} d\mathbf{X}(t) &= \mathbf{f}(\mathbf{X}(t), t)dt + g(\mathbf{X}(t), t)d\mathbf{W}(t) \,, \\ 0 \leq t \leq t_f, \quad \mathbf{X}(0) &= \mathbf{x}_0 \in \mathcal{D}_x \subset \mathbb{R}^{n_x}, \end{aligned} \tag{3.3}$$

where the state-dependent Wiener process coefficient is the array

$$g(\mathbf{X}(t), t) = \left[g_{i,j}(\mathbf{X}(t), t) \right]_{n_x \times n_w} \,,$$

representing the magnitude of the random environmental or parameter perturbations. The conditional infinitesimal expectation of the state differential process is

$$\text{E}[d\mathbf{X}(t) \,|\, \mathbf{X}(t) = \mathbf{x}] = \mathbf{f}(\mathbf{x}, t)dt \tag{3.4}$$

from (3.1) and conditional infinitesimal covariance is

$$\text{Cov}\left[d\mathbf{X}(t), d\mathbf{X}^\top(t) \,|\, \mathbf{X}(t) = \mathbf{x} \right] = \left(gR^{(w)}(t)g^\top \right)(\mathbf{x}, t)dt \tag{3.5}$$

from from (3.2).

These infinitesimal moments (3.4-3.5) basically define the *stochastic diffusion process*. The conditional infinitesimal state $d\mathbf{X}(t)$ is formally a multi-dimensional Gaussian distributed process with mean $\mathbf{f}(\mathbf{x}, t)dt$ and covariance $\left(gR^{(w)}(t)g^\top \right)(\mathbf{x}, t)dt$.

3.1. Diffusion Stochastic Integral Equation (SIE). Since the diffusion SDE serves more as a symbolic representation of the state stochastic process and the integral form is more amenable to approximations, the diffusion stochastic integral equation (SIE) is used,

$$\mathbf{X}(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{X}(s), s)ds + \int_0^t g(\mathbf{X}(s), s)d\mathbf{W}(s) \,, \tag{3.6}$$

where the first integral is similar to the usual Riemann integral with a deterministic and continuous time element of integration ds although the first argument of the integrand depends on the stochastic state process. However, the second integral in decidedly not like a Riemann integral but has a nonsmooth element of integration $d\mathbf{W}(s)$ and is a more Stieltjes-like random integral.

In his 1951 AMS seminal memoir [36], Kiyoshi Itô, following the compatibility with the Markov memory properties of the Wiener process, suggests that the forward

integration rule be used to approximate that second integral. Also, for consistency in numerical approximation, we will also use the same approximation for the first, almost deterministic-like integral. See Higham's compact review of SDE simulations [32] for more accurate numerical approximations. Hence, letting $t_0 = 0$, $t_n = t$, $W_0 = 0$, $t_{j+1} = t_j + \Delta t_j$, $\delta t_n = \max_{j=0:n}[\Delta t_j]$ and $\Delta \mathbf{W}_j = \mathbf{W}_{j+1} - \mathbf{W}_j$ for $j = 0 : n$ forward steps,

$$\mathbf{X}_{n+1} \overset{\text{fwd}}{\simeq} \mathbf{x}_0 + \sum_{j=0}^{n} \left(\mathbf{f}_j \Delta t_j + g_j \Delta \mathbf{W}_j \right) , \tag{3.7}$$

where $\mathbf{f}_j \equiv \mathbf{f}(\mathbf{X}_j, t_j)$, $g_j \equiv g(\mathbf{X}_j, t_j)$, $\mathbf{X}_j = \mathbf{X}(t_j)$ and $\mathbf{W}_j = \mathbf{W}(t_j)$.

However, the computational form (3.7) is insufficient for defining convergence as the mesh size $\delta t_n \to 0^+$ as $n \to +\infty$ since differing random simulations of the W_j may not lead to the same answer. Further, the *Itô mean square (IMS) convergence* is introduced for well-posed convergence, i.e.,

$$\mathbf{X}(t) \overset{\text{ims}}{=} \mathbf{Z}(t) \quad \Longleftrightarrow \quad \lim_{\substack{n\to\infty \\ \delta t_n \to 0}} \mathrm{E}\left[\left| \mathbf{X}_{n+1} - \mathbf{Z}_{n+1} \right|^2 \right] = 0 \, , \tag{3.8}$$

plus some technical conditions (e.g., mean square integrability), $t_{n+1} = t$ fixed, the important *independent increment property*,

$$\mathrm{Cov}[\Delta W_{i,j}, \Delta W_{i,k}] = \mathrm{E}[\Delta W_{i,j} \Delta W_{i,k}] = \delta_{k,j} \, , \tag{3.9}$$

where $\Delta W_{i,j} = \Delta W_i(t_j) \equiv W_i(t_j + \Delta t_j) - W_i(t_j)$, $\Delta t_j > 0$, and the *diagonal decomposition*

$$\sum_{j=0}^{n} \sum_{k=0}^{n} a_{j,k} \Delta W_{i,j} \Delta W_{i,k} = \sum_{j=0}^{n} \left(a_{j,j} (\Delta W_{i,j})^2 + \sum_{k \neq j}^{n} \Delta W_{i,j} \Delta W_{i,k} \right) .$$

Here, $\delta_{i,j}$ is the *Kronecker delta*, which is zero if $i \neq j$ and one if $i = j$.

3.2. Fundamental IMS Convergence Examples. The most fundamental example of Wiener integrals is

$$\int_0^t (dW_i)^2(s) \overset{\text{ims}}{=} t \quad \Rightarrow \quad (dW_i)^2(t) \overset{\text{dt}}{=} dt \, ; \tag{3.10}$$

the mean square convergence of the former form is shown in detail by Hanson [25, Chapter 2] or Arnold [7, Chapter 3] and the latter form being a symbolic representation of that result. This result should be compared to the deterministic Riemann model where

$$(dt)^2 \overset{\text{dt}}{=} 0$$

applying the Riemann sum rules for $\int_0^t (ds)^2$ in keeping only terms of order δt_n. Also in the direct forward approximation sense that

$$\int_0^t (ds)^2 \quad \overset{\text{fwd}}{\simeq} \quad \sum_{j=0}^{n} (\Delta t_j)^2 \leq \delta t_n \cdot \sum_{j=0}^{n} \Delta t_j = t \cdot \delta t_n \to 0$$

as $n \to \infty$ and $\delta t_n \to 0^+$ with t fixed or trivially in the Itô mean square convergence,

$$\mathrm{E}\left[\left(\sum_{j=0}^{n}(\Delta t_j)^2 - 0\right)^2\right] = \left(\sum_{j=0}^{n}(\Delta t_j)^2\right)^2 \le \left(\delta t_n \cdot \sum_{j=0}^{n}(\Delta t_j)\right)^2 = (t\cdot\delta t_n)^2 \to 0\,.$$

Similarly, in symbolic SDE form,

$$dW_i(t)\cdot dW_j(t) \stackrel{\mathrm{dt}}{=} \rho_{i,j}^{(w)}(t)dt \quad \text{or} \quad d\mathbf{W}(t)d\mathbf{W}^\top(t) \stackrel{\mathrm{dt}}{=} R^{(w)}(t)dt\,;$$
$$dW_i(t)\cdot dt \stackrel{\mathrm{dt}}{=} 0\,;$$
$$(dW_i)^k(t) \stackrel{\mathrm{dt}}{=} 0 \quad \text{if} \quad k>2\,.$$

3.3. Change of Variables and Stochastic Diffusion Chain Rule. For the analysis of changes of variables it is convenient to use dt-precision on *increment (inc) definitions*, e.g., let

$$\mathbf{X}(t+dt) \stackrel{\mathrm{inc}}{=} \mathbf{X}(t) + d\mathbf{X}(t) \quad \Longleftrightarrow \quad d\mathbf{X}(t) \stackrel{\mathrm{inc}}{=} \mathbf{X}(t+dt) - \mathbf{X}(t)\,.$$

Let $\mathbf{Y}(t) = \mathbf{F}(\mathbf{X}(t),t)$ be a change of state variables, so the stochastic diffusion chain rule is

$$\begin{aligned}
d\mathbf{Y}(t) &\stackrel{\mathrm{inc}}{=} \mathbf{Y}(t+dt) - \mathbf{Y}(t) = \mathbf{F}(\mathbf{X}(t+dt),t+dt) - \mathbf{F}(\mathbf{X}(t),t)\\
&\stackrel{\mathrm{inc}}{=} \mathbf{F}(\mathbf{X}(t)+d\mathbf{X}(t),t+dt) - \mathbf{F}(\mathbf{X}(t),t)\\
&\stackrel{\mathrm{taylor}}{=} \mathbf{F}_t(\mathbf{X}(t),t)dt + \left(\mathbf{X}^\top(t)\nabla_x\right)[\mathbf{F}](\mathbf{X}(t),t)\\
&\quad +\tfrac{1}{2}d\mathbf{X}^\top(t)\nabla_x\left[\nabla_x[\mathbf{F}]\right](\mathbf{X}(t),t)d\mathbf{X}(t) + \dots \qquad (3.11)\\
&= \left(\mathbf{F}_t dt + \left((\mathbf{f}dt + gd\mathbf{W}(t))^\top\nabla_x\right)[\mathbf{F}]\right)(\mathbf{X}(t),t)\\
&\quad +\tfrac{1}{2}(gd\mathbf{W})^\top\nabla_x\left[\nabla_x[\mathbf{F}]\right]gd\mathbf{W}(t) + \dots\\
&\stackrel{\mathrm{dt}}{=} \left(\mathbf{F}_t + \mathbf{f}^\top\nabla_x[\mathbf{F}] + \tfrac{1}{2}\left(gR^{(w)}g^\top\right):\nabla_x\left[\nabla_x[\mathbf{F}]\right]\right)dt + \nabla_x^\top[\mathbf{F}]gd\mathbf{W}(t)\,,
\end{aligned}$$

where the *double-dot product* is defined here as the trace

$$A:B \equiv \mathrm{Trace}[AB] = \sum_{j=1}^{n_x}\sum_{k=1}^{n_x} A_{j,k}B_{k,j}\,.$$

In components,

$$dY_i(t) \stackrel{\mathrm{dt}}{=} \left(F_{i,t} + \sum_{j=1}^{n_x} f_j F_{i,x_j} + \frac{1}{2}\sum_{j=1}^{n_x}\sum_{k=1}^{n_x}\sum_{\ell=1}^{n_w}\sum_{m=1}^{n_w} g_{j,\ell}\rho_{\ell,m}^{(w)}g_{k,m}F_{i,x_j,x_k}\right)dt$$
$$+\sum_{j=1}^{n_x}\sum_{k=1}^{n_w} g_{j,k}F_{i,x_j}dW_k(t)\,,$$

for $i = 1{:}n_x$.

Although the transformed first order stochastic diffusion term $\nabla_x^\top[\mathbf{F}]gd\mathbf{W}(t)$ in (3.11) is expected from the appearance of similar transformed first order deterministic term $\mathbf{f}^\top\nabla_x[\mathbf{F}]dt$ in (2.4), the really new term is the second order diffusion term

$$\frac{1}{2}\left(gR^{(w)}g^\top\right):\nabla_x[\nabla_x[\mathbf{F}]]dt\,.$$

This diffusion term implies that stochastic diffusions lead to the same type of second order parabolic PDEs as do physical diffusions, although arising from different underlying models. For stochastic diffusions, this term arises due the fact that the quadratic of the Wiener process is linear in dt from Eq. (3.10), i.e., $(dW_i)^2(t)\stackrel{\text{dt}}{=}dt$, which is a fundamental and surprising deterministic result since the *square of a stochastic process can be deterministic*.

3.4. Scalar, Linear, Diffusion SDE Example. Again, consider the $n_x=1$ and $n_w=1$ scalar, linear model, so the diffusion SDE takes the form,

$$dX(t)=X(t)\cdot(A(t)dt+B(t)dW(t)),\quad X(0)=x_0>0\,, \tag{3.12}$$

where $A(t)$ and $B(t)$ are given integrable functions. Again let

$$Y(t)=F(X(t),t)=\ln(X(t))$$

be the logarithmic state transformation, with partial derivatives

$$F_t(x,t)=0\,,\quad F_x(x,t)=1/x\,,\quad F_{x,x}(x,t)=-1/x^2\,,$$

assuming $x>0$. Then, the transformed diffusion SDE by the chain rule (3.11) becomes state-independent,

$$dY(t)\stackrel{\text{dt}}{=}\ln(X(t)+dX(t))-\ln(X(t))\stackrel{\text{dt}}{=}\left(A(t)-B^2(t)/2\right)dt+B(t)dW(t)\,,$$

to dt-precision, so that immediate integration yields

$$Y(t)\stackrel{\text{ims}}{=}Y(0)+\int_0^t\left(A(s)-B^2(s)/2\right)ds+\int_0^t B(s)dW(s)\,, \tag{3.13}$$

the sum of a Riemann integral and a Wiener integral. The term $B^2(t)/2$ is the *Itô correction or diffusion coefficient*. Inverting the logarithmic transformation yields

$$X(t)=x_0\exp\left(\int_0^t\left(\left(A(s)-B^2(s)/2\right)ds+B(s)dW(s)\right)\right), \tag{3.14}$$

preserving the initial positivity such that $x_0>0$ implies $X(t)>0$ for all $t\geq 0$ as in the deterministic case. The solution (3.14) is formally closed if the integrals are known, else the forward integration approximation can be used along with random simulation of the Wiener process $W(t)$.

3.4.1. Random Simulated Diffusion Solution. For numerical simulation purposes, it is helpful to find a recursive form of the solutions (3.14), but it is numerically preferable to do this for the exponent in (3.13). Hence, using the forward approximation on (3.13) with time steps Δt_j for $j=0:n_t-1$ on for $t\in[0,t_f]=[t_0,t_{n_t}]$ with corresponding Wiener increments of ΔW_j,

$$\begin{aligned}Y_{j+1}&\stackrel{\text{fwd}}{\simeq}\ln(x_0)+\textstyle\sum_{k=0}^j\left((A_j-B_j^2/2)\Delta t_k+B_k\Delta W_k\right)\\&=Y_j+(A_j-B_j^2/2)\Delta t_j+B_j\Delta W_j\,,\end{aligned}$$

so

$$X_{j+1} \overset{\text{fwd}}{\simeq} \exp\left(Y_{j+1}\right).$$

There are no truly random simulations, thus, technically speaking, we need to find *pseudo-random simulations* of the Wiener model of stochastic diffusions. Since the Wiener differential process $dW(t)$ is normally distributed with mean zero, $\mathrm{E}[dW(t)] = 0$, and dt variance, $\mathrm{Var}[dW(t)] = dt$, the distribution has the form

$$\Phi_{dW(t)}(w) = \int_{-\infty}^{w} \phi_{dW(t)}(v)dv\,,$$

where the density of $dW(t)$ is

$$\phi_{dW(t)}(w) = \exp\left(-w^2/(2dt)\right)/\sqrt{2\pi dt}\,.$$

However, theoretical normal pseudo-random number generators are usually available for the standard, zero-mean and unit-variance, normal with density,

$$\phi_Z(z) = \exp\left(-z^2/2\right)/\sqrt{2\pi},$$

where Z is a standard normal variate. A simple change of variables $u = v/\sqrt{dt}$ can change the distribution of $dW(t)$ into that for the standard Z variate as follows,

$$\Phi_{dW(t)}(w) = \int_{-\infty}^{w} \frac{\exp\left(-v^2/(2dt)\right)}{\sqrt{2\pi dt}} = \int_{-\infty}^{w/\sqrt{dt}} \frac{\exp\left(-u^2/2\right)}{\sqrt{2dt}} = \Phi_Z\left(\frac{w}{\sqrt{dt}}\right).$$

Thus, we have shown *in distribution*,

$$dW(t) \overset{\text{dist}}{=} \sqrt{dt}Z\,. \tag{3.15}$$

A corresponding simplified sample code in MATLAB™ (although it could also be coded in Maple™, Mathematica™ or other computing system) follows for a single sample path with comments marked by the symbol "%" sign:

```
function diffusionpaths
% Sample Diffusion SDE Test Code
% Scalar, Linear Model:
%   dX(t) = X(t)*(A(T)*dt+B(t)*dW(t)); X(0)=x0; 0<t<tf;
nt = 1000; t0 = 0; tf = 2.0; dt = (tf-t0)/nt; %sample input: fixed dt;
x0 = 1; % initial state;
t = 0:dt:tf; % set time vector in unit base: t(1)=0; t(nt+1)=tf;
sqrtdt = sqrt(dt); % Wiener scaling;
a = A(t); b = B(t); % assumes vector subfunctions for coefficients;
y = zeros(nt+1); % predeclare for efficiency;
y(1) = log(x0)% log = ln; unit not zero index base;
dw = sqrtdt*randn(nt); % Wiener (0,dt)-normal step matrix;
for j = 1:nt % exponent update:
       y(j+1) = y(j) + (a(j)-b(j)^2/2)*dt+b(j)*dw(j);
end
x = exp(y); % vector state;
%
```

```
% Plot:
plot(t,x),'k-','linewidth',3)
title('Diffusion Simulated Sample Path');
ylabel('X(t),State'); xlabel('t, Time');
%
function av = A(t) % must be vector subfunction, e.g., dot-division;
av = 0.3*t./(1+t); % sample, fill-in for each problem;
%End A
function bv = B(t) % must be vector subfunction, e.g., dot-division;
bv = 0.5*sqrt(t)./(1+t); % sample, fill-in for each problem;
%End B
%
%%End diffusionpaths.m
%
```

A sample illustration using a more complicated MATLAB™ code with four sample paths using different random states (seeds) is given in Figure 3.1.

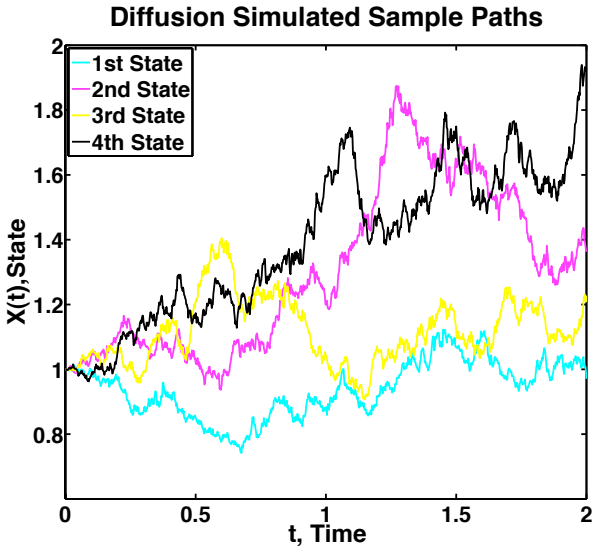

FIG. 3.1. *Diffusion paths for a scalar, linear SDE (3.12) are simulated using MATLAB™ with* $nt = 1000$ *sample points, four* `randn` *states and maximum time* $t_f = 2.0$ *starting at* $x_0 = 1.0$.

4. Simple Jump-Diffusion Stochastic Differential Equation. Although the continuous models of the deterministic and stochastic diffusion models discussed in the prior two sections are suitable for many applications, many other applications exhibit large random fluctuations where diffusions alone would not be suitable. Some applications for which discontinuous components are needed are financial market models having sudden crashes or rallies. Other examples are ecological populations subject to sudden disasters due to such phenomena as earthquakes, related tsunamis, intense storms, sudden temperature changes, predation and disease. Ecological bonanzas are also possible due to a sudden influx or encounter of nutrients and a discovery of many prey. Other examples of problems that can be approximated by discontinuous processes are found in manufacturing, such as machine changes, part changes, repair,

maintenance and strikes. Some manufacturing changes can involve both negative and positive jumps as with stoppages and resumptions of production.

For Markov jump-diffusions, Poisson processes fill the role of the jump part of the jump-diffusions due to their facility in modeling rare events and preserving the Markov property. Let $\mathbf{P}(t) = [P_i(t)]_{n_p \times 1}$ be an n_p-dimensional Poisson vector process with differential component $dP_i(t)$ having a common mean and variance,

$$\mathrm{E}[dP_i(t)] = \lambda_i(t)dt, \quad \mathrm{Var}[dP_i(t)] = \lambda_i(t)dt\,, \tag{4.1}$$

for $i = 1 : n_p$, where $\lambda_i(t) > 0$ is the ith jump-rate or jump-intensity and $\lambda_i(t)dt$ is the mean jump count of the ith Poisson process in $(t, t + dt]$. Along with the Markov property of Poisson processes, they have *independent increments* in time,

$$\begin{aligned} \mathrm{Cov}[dP_i(t_j), dP_i(t_k)] &= \mathrm{E}[(dP_i(t_j) - \lambda_i(t_j)dt)(dP_i(t_k) - \lambda_i(t_k)dt)] \\ &= \mathrm{Var}[dP_i(t_j)]\ \delta_{k,j} = \lambda_i(t_j)dt\ \delta_{k,j}\,, \end{aligned} \tag{4.2}$$

where $\delta_{k,j}$ is the *Kronecker delta*, or if s and t are continuous arguments,

$$\mathrm{Cov}[dP_i(t), dP_i(s)] = \lambda_i(t)dt\delta(s - t)ds\,, \tag{4.3}$$

where $\delta(x)$ is the *Dirac delta function*, i.e., the covariance only has a point concentrated value if $s = t$ represented by $\delta(s - t)ds$ such that $\int_0^t \delta(s - t)ds = 1$.

By convention, $\mathbf{P}(0) = \mathbf{0}_{n_p \times 1}$ with probability one. The $P_i(t)$ are *right-continuous with left-limits (RCLL)* and have unit jumps such that

$$\mathrm{Jump}[P_i](T_{i,j}) \equiv [P_i](T_{i,j}) = P_i\left(T_{i,j}^+\right) - P_i\left(T_{i,j}^-\right) \equiv 1\,,$$

where $T_{i,j}$ is the jth jump of the ith Poisson component process, $T_{i,j}^-$ is the pre-jump time and the post-jump is $T_{i,j}^+ = T_{i,j}$ by right-continuity. Thus, *Poisson jumps are instantaneous*, i.e., $dt \equiv 0$ during a jump and hence there is no time for continuous changes. See Çinlar [17] or Snyder and Miller [61] for more concise definitions of Poisson processes.

The Poisson component process $P_i(t)$ is found by the fundamental integration theorem of calculus,

$$P_i(t) = \int_0^t dP_i(s)$$

and since time is deterministic, the expectation is

$$\mathrm{E}[P_i(t)] = \int_0^t \mathrm{E}[dP_i(s)] = \int_0^t \lambda_i(s)ds \equiv \Lambda_i(t)\,,$$

but in the simpler stationary case when $\lambda_i(t) = \lambda_0$ is a constant, then $\mathrm{E}[P_i(t)] = \lambda_0 t$. Similarly, the Poisson increment, $\Delta P_i(t) = P_i(t + \Delta t) - P_i(t)$, has the expectation

$$\mathrm{E}[\Delta P_i(t)] = \int_t^{t+\Delta t} \lambda_i(s)ds \equiv \Delta\Lambda_i(t)\,.$$

From the independent increment, covariance result (4.3) and the definition of

$P_i(t)$ in terms of $dP_i(t)$,

$$\begin{aligned}\mathrm{Var}[P_i(t)] =\ & E\left[\left(\int_0^t dP_i(s) - \Lambda_i(t)\right)^2\right] = \mathrm{E}\left[\left(\int_0^t (dP_i(s) - \lambda_i(s)ds)\right)^2\right] \\ &= \int_0^t\int_0^t \mathrm{E}\left[(dP_i(s) - \lambda_i(s)ds)\,(dP_i(r) - \lambda_i(s)ds)\right] \\ &= \int_0^t\int_0^t \mathrm{Cov}[dP_i(s)dP_i(r)] = \int_0^t\int_0^t \lambda_i(s)\delta(r-s)drds = \Lambda_i(t)\,,\end{aligned}$$

preserving the expectation-variance symmetry of Poisson processes.

The correlation coefficients of the component processes when $s = t$ are related to the covariance by the definition

$$\mathrm{Corr}[dP_i(t), dP_j(t)] \equiv \frac{\mathrm{Cov}[dP_i(t), dP_j(t)]}{\sqrt{\lambda_i(t)\lambda_j(t)}\ dt}\,.$$

Due to the jump-rate $\sqrt{\lambda_i(t)\lambda_j(t)}$ in the normalization, for simplicity the covariance will be used instead of the correlation coefficients to define the simplified dependence

$$\rho_{i,j}^{(p)}(t)dt \equiv \mathrm{Cov}[dP_i(t), dP_j(t)]\ ,$$

where $R^{(p)}(t) = [\rho_{i,j}^{(p)}(t)]_{n_p\times n_p}$ is the *Poisson covariance matrix*, so $R^{(p)}(t)$ is closer to the Wiener correlation matrix $R^{(w)}(t)$. If there is no correlation between distinct Poisson components, then the Poisson covariance matrix is diagonal, i.e., $R^{(p)}(t) = [\lambda_i(t)\delta_{i,j}]_{n_p\times n_p}$.

Hence, along with assuming independent increments over distinct times, this leads to the result that for distinct components over the same time t,

$$\begin{aligned}\mathrm{Cov}[P_i(t), P_j(t)] =& \mathrm{E}\left[\int_0^t\int_0^t (dP_i(s) - \lambda_i(s)ds)\,(dP_j(r) - \lambda_j(r)dr)\right] \\ &= \int_0^t\int_0^t \mathrm{E}\left[(dP_i(s) - \lambda_i(s)ds)\,(dP_j(r) - \lambda_j(r)dr)\right] \\ &= \int_0^t \mathrm{Cov}[dP_i(s), dP_j(s)] = \int_0^t \rho_{i,j}^{(p)}(s)ds\end{aligned}$$

and consequently for the Poisson vector processes at time t,

$$\mathrm{Cov}\left[\mathbf{P}(t), \mathbf{P}^\top(t)\right] = \int_0^t \left[\mathrm{Cov}[dP_i(s), dP_j(s)]\right]_{n_p\times n_p} = \int_0^t R^{(p)}(s)ds\,.$$

Similar formulas hold for the increments $\Delta P_i(t)$ and $\Delta P_j(t)$.

The Poisson processes $P_i(t)$, $\Delta P_i(t)$ and $\Delta P_i(t)$ all are *Poisson-distributed*, i.e., in symbolic array notation,

$$\mathrm{Prob}\left[\begin{pmatrix} P_i(t) \\ \Delta P_i(t) \\ dP_i(t)\end{pmatrix} = k\right] = \exp\left(-\begin{pmatrix} \Lambda_i(t) \\ \Delta\Lambda_i(t) \\ \lambda_i(t)dt\end{pmatrix}\right)\cdot\begin{pmatrix} \Lambda_i(t) \\ \Delta\Lambda_i(t) \\ \lambda_i(t)dt\end{pmatrix}^k \Big/ k!\,. \tag{4.4}$$

The Poisson distribution can be simplified for the differential $dP_i(t)$ using dt-precision,

$$\mathrm{Prob}[dP_i(t) = k] \overset{\mathrm{dt}}{=} (1 - \lambda_i(t)dt)\delta_{k,0} + \lambda_i(t)dt\delta_{k,1} \tag{4.5}$$

which specifies the *Zero-One Law (ZOL)* for jumps of $dP_i(t)$. In this special case the Poisson distribution reduces to a Bernoulli distribution, i.e., having just two possible events, here zero or one jump. It follows from ZOL (4.5), the *Poisson form of dt-precision*, that

$$\mathrm{E}\left[(dP_i)^m(t)\right] \underset{\text{zol}}{\overset{\text{dt}}{=}} (1-\lambda_i(t)dt)\cdot 0^m + \lambda_i(t)dt\cdot 1^m = \lambda_i(t)dt\,,$$

if the integer $m>0$. So the powers of the Poisson process do not truncate to a finite number, unlike the Wiener process differentials which truncate at the second order so contributing only derivatives up to second order. In contrast, the Poisson differential can contribute derivatives of all orders, usually represented as functional integrals, or delayed as well as advanced arguments causing global dependence rather than local dependence of partial derivatives of finite order.

Let the resulting stochastic differential equation (SDE) for the jump-diffusion process be

$$\begin{gathered} d\mathbf{X}(t) = \mathbf{f}(\mathbf{X}(t),t)dt + g(\mathbf{X}(t),t)d\mathbf{W}(t) + h(\mathbf{X}(t),t)d\mathbf{P}(t)\,, \\ 0\le t\le t_f,\quad \mathbf{X}(0)=\mathbf{x}_0\in\mathcal{D}_x\subset\mathbb{R}^{n_x}, \end{gathered} \tag{4.6}$$

where the Poisson process coefficient or jump-amplitude is the array

$$h(\mathbf{x},t) = \left[h_{i,j}(\mathbf{x},t)\right]_{n_x\times n_p}\,,$$

representing the very large or discontinuous random environmental or parameter perturbations. The element $h_{i,j}(\mathbf{x},t)$ is the jump-amplitude of ith state $X_i(t)$ and the jth Poisson component $P_j(t)$. The conditional infinitesimal expectation of the state differential process is

$$\mathrm{E}[d\mathbf{X}(t)\,|\,\mathbf{X}(t)=\mathbf{x}] = (\mathbf{f}(\mathbf{x},t)+h(\mathbf{x},t)\boldsymbol{\lambda}(t))dt\,,$$

where $\boldsymbol{\lambda}(t)=[\lambda_i(t)]_{n_p\times 1}$ is a jump-rate vector, and conditional infinitesimal covariance is

$$\mathrm{Cov}\left[d\mathbf{X}(t), d\mathbf{X}^\top(t)\,|\,\mathbf{X}(t)=\mathbf{x}\right] = \left(gR^{(w)}(t)g^\top + hR^{(p)}(t)h^\top\right)(\mathbf{x},t)dt\,.$$

Both conditional infinitesimal moments do not completely define the jump-diffusion process, since in addition the jump size must be specified, i.e., for the kth jump of the jth Poisson process,

$$\mathrm{Jump}[\mathbf{X}](T_{j,k}) = \left[h_{i,j}\left(\mathbf{X}\left(T_{j,k}^-\right),T_{j,k}^-\right)\right]_{n_x\times 1}\,. \tag{4.7}$$

Note that Poisson processes model rare events, so as a more general consequence of the zero-one jump law, it is unlikely that there would be more than one jump among all the Poisson components at a given time. Thus, there is no sum involved in (4.7). The Poisson process evaluates the time for the jump-amplitude array at the pre-jump time $T_{j,k}^-$, otherwise the Markov short memory property would be violated using the near-future time $T_{j,k}^+ = T_{j,k}$, the equality stemming from the right-continuity property.

The corresponding stochastic integral equation follows from the formal integration of the jump-diffusion SDE (4.6),

$$\mathbf{X}(t) = \mathbf{x}_0 + \int_0^t \left(\mathbf{f}(\mathbf{X}(s),s)ds + g(\mathbf{X}(s),s)d\mathbf{W}(s) + h(\mathbf{X}(s),s)d\mathbf{P}(s)\right), \tag{4.8}$$

where an alternate form of the jump part of the integral in (4.8) is given by

$$\int_0^t h(\mathbf{X}(s),s)d\mathbf{P}(s) = \sum_{j=1}^{n_p}\sum_{k_j=1}^{P_j(t)} h\Big(\mathbf{X}\left(T_{j,k_j}^-\right), T_{j,k_j}\Big)\,, \tag{4.9}$$

since the jump change in the state must be the sum of all the jumps in the Poisson vector process $\mathbf{P}(s)$ for $s \in (0,t]$. By convention, there is a reverse-sum rule that

$$\sum_{k=1}^{0} a_k \equiv 0\,,$$

since if $P_j(t) = 0$, then there are no jumps of the jth process and $P_j(s) = 0$ on $[0,t]$. Note that the alternate sum form in (4.9) is useful for motivation purposes, but requires extra effort in computing moments due to an extra step required to handle the pre-jump times T_{j,k_j}^- in iterated expectations [25, Chapter 3].

The complication occurs because the distribution of the Poisson counting distribution for $P_j(t)$ and the Poisson jump-time distribution for $T_{j,k}$ are closely related. Consider the less complicated case of a stationary Poisson process where the jump-rate is constant, i.e., $\lambda_j(t) = \lambda_0$. The basic idea [25] is that the probability of the time between jumps $\Delta T_{j,k} = T_{j,k+1} - T_{k,j}$ less than $\Delta t > 0$, conditioned on the prior jump time $T_{j,k}$, is the same as the probability that there be at least one jump in the time interval, which is the same as one minus the probability that there are no jumps in the time interval, i.e.,

$$\begin{aligned}
\text{Prob}[\Delta T_{j,k} \le \Delta t \,|\, T_{j,k}] &= 1 - \text{Prob}[\Delta T_{j,k} > \Delta t \,|\, T_{j,k}] \\
&= 1 - \text{Prob}[\Delta P(T_{j,k}) = 0 \,|\, T_{j,k}] \\
&= 1 - \text{Prob}[P(\Delta t) - P(0) = 0] \\
&= 1 - \text{Prob}[P(\Delta t) = 0] = 1 - e^{-\lambda_0 \Delta t}\,,
\end{aligned}$$

where the stationary property of the simple Poisson process $P(t)$, with λ_0 constant, has been used. Also used are the facts that the probability of the difference depends on the difference in time $\Delta T_{j,k}$ and that $P(0) = 0$ with probability. The jump-time increments distribution is the exponential distribution

$$\Phi_{\Delta T_{j,k}}(\Delta t) = 1 - e^{-\lambda_0 \Delta t}$$

with mean $1/\lambda_0$. For the more general case of nonstationary Poisson process, $\lambda = \lambda(t)$, see Çinlar [17, Chapter 4], who uses a change of time variable that makes this case more amenable.

4.1. Change of Variables and Simple Jump-Diffusion Chain Rule. Since Poisson jumps are discontinuous in time and thus instantaneous, there is no time for any continuous changes, leading to the very basic decomposition of the jump-diffusion state process,

$$d\mathbf{X}(t) = d_{\text{cont}}\mathbf{X}(t) + d_{\text{jump}}\mathbf{X}(t)\,, \tag{4.10}$$

where

$$d_{\text{cont}}\mathbf{X}(t) = \mathbf{f}(\mathbf{X}(t),t)dt + g(\mathbf{X}(t),t)d\mathbf{W}(t)$$

is the Gaussian part and

$$d_{\text{jump}}\mathbf{X}(t) = h(\mathbf{X}(t), t)d\mathbf{P}(t)$$

is the Poisson part. The continuous and jump parts can be calculated independently due to the instantaneous nature of the jumps. Taking the Poisson differential $d\mathbf{P}(t)$ as the condition for a jump, the jump of the state can be generally written at any time t,

$$\text{Jump}[\mathbf{X}](t) = \mathbf{X}(t^+) - \mathbf{X}(t^-) = h(\mathbf{X}(t^-), t^-)d\mathbf{P}(t)\,. \tag{4.11}$$

The two t^-'s on the far right could just as well be plain t's, since it is the Poisson process that picks out the pre-jump time. Also, usually the explicit time argument of the jump-amplitude is continuous, so $h(\mathbf{X}(t^-), t)$ can also be used in (4.11).

Let $\mathbf{Y}(t) = \mathbf{F}(\mathbf{X}(t), t)$ be a state transformation, then

$$\begin{aligned}
\text{Jump}[\mathbf{Y}](t) &= \mathbf{Y}(t^+) - \mathbf{Y}(t^-) = \mathbf{F}(\mathbf{X}(t^+), t) - \mathbf{F}(\mathbf{X}(t^-), t) \\
&= \mathbf{F}(\mathbf{X}(t) + \text{Jump}[\mathbf{X}](t), t) - \mathbf{F}(\mathbf{X}(t^-), t) \\
&= \mathbf{F}(\mathbf{X}(t^-) + h(\mathbf{X}(t^-), t)d\mathbf{P}(t), t) - \mathbf{F}(\mathbf{X}(t^-), t) \\
&\overset{\text{zol}}{=} \sum_{j=1}^{n_p} \left(\mathbf{F}\left(\mathbf{X}(t^-) + \left[h_{i,j}(\mathbf{X}(t^-), t)\right]_{n_x\times 1}, t\right) - \mathbf{F}(\mathbf{X}(t^-), t)\right) dP_j(t)\,,
\end{aligned} \tag{4.12}$$

where the *strong, vector version of the zero-one jump law* has been used, while the pure time arguments of both $\mathbf{F}(\mathbf{x}, t)$ and $h(\mathbf{x}, t)$ have lost their one-sided limit notations, since both are usually continuous in the explicit t argument.

Combining the jump change of variable (4.12) with the stochastic diffusion chain rule (3.11) yields the jump-diffusion chain rule for simple Poisson jumps,

$$\begin{aligned}
d\mathbf{Y}(t) &\overset{\text{inc}}{=} \mathbf{F}(\mathbf{X}(t) + d_{\text{cont}}\mathbf{X}(t), t + dt) - \mathbf{F}(\mathbf{X}(t), t) + \text{Jump}[\mathbf{Y}](t) \\
&\underset{\text{zol}}{\overset{\text{dt}}{=}} \left(\mathbf{F}_t + \mathbf{f}^\top \nabla_x[\mathbf{F}] + \tfrac{1}{2}\left(gR^{(w)}g^\top\right) : \nabla_x\left[\nabla_x[\mathbf{F}]\right]\right) dt + \nabla_x^\top[\mathbf{F}]gd\mathbf{W}(t) \\
&\quad + \sum_{j=1}^{n_p} \left(\mathbf{F}\Big(\mathbf{X}(t) + \left[h_{i,j}(\mathbf{X}(t), t)\right]_{n_x\times 1}, t\Big) - \mathbf{F}(\mathbf{X}(t), t)\right) dP_j(t)\,,
\end{aligned} \tag{4.13}$$

where it is understood that the Poisson jump process $P_j(t)$ picks out the pre-jump time $T_{j,k}^-$ of the state process $\mathbf{X}(t)$ if there is the kth jump at time $T_{j,k}$ of the jth Poisson process. The added complexity from adding simple Poisson jumps to the stochastic diffusion model under a change of variables is that jumps introduce differences rather than partial derivatives into the transformed SDE. For finite, nontrivial jump-amplitude coefficients $h(\mathbf{x}, t)$ that means there is *state global dependence* at fixed t and points $\mathbf{x} + [h_{i,j}(\mathbf{x}, t)]_{n_x\times 1}$ from the jumps and not just *state local dependence* on the present state $\mathbf{x}$ at fixed t due to the partial derivatives arising from the stochastic diffusion.

4.2. Scalar, Linear, Simple Jump-Diffusion SDE Example. Consider the scalar, linear simple jump-diffusion model with $n_x = 1$, $n_w = 1$ and $n_p = 1$,

$$dX(t) = X(t)\cdot(A(t)dt + B(t)dW(t) + C(t)dP(t)), \quad X(0) = x_0 > 0\,, \tag{4.14}$$

where $A(t)$, $B(t)$ and $C(t)$ are given integrable functions. One further essential condition is needed that

$$C(t) > -1\,,$$

since otherwise if $C(t) \leq -1$ with $X(t) > 0$, then the first jump would take the state out of positive values, i.e., the new post-jump state would be $X(t) + \text{Jump}[X](t) = (1 + C(t))X(t) \leq 0$.

Again let

$$Y(t) = F(X(t), t) = \ln(X(t))$$

be the logarithmic state transformation, with partial derivatives

$$F_t(x,t) = 0\,, \quad F_x(x,t) = 1/x\,, \quad F_{x,x}(x,t) = -1/x^2\,,$$

assuming $x > 0$, but now with the conditional jump,

$$\text{Jump}[Y](t) = (\ln(X(t) + C(t)X(t)) - \ln(X(t)))\, dP(t) = \ln(1 + C(t))dP(t)\,,$$

provided $X(t) > 0$ and $C(t) > -1$. Then, the transformed jump-diffusion SDE by the chain rule (4.13) becomes state-independent, by adding the jump-change to the continuous diffusion-changes, and finally results in

$$dY(t) \underset{\text{zol}}{\overset{\text{dt}}{=}} \left(A(t) - B^2(t)/2\right) dt + B(t)dW(t) + \ln(1 + C(t))dP(t)\,,$$

to dt-precision and the zero-one jump law. Immediate integration yields

$$Y(t) \overset{\text{ims}}{=} Y(0) + \int_0^t \left(\left(A(s) - B^2(s)/2\right) ds + B(s)dW(s) + \ln(1+C(s))dP(s)\right), \quad (4.15)$$

the sum of a Riemann integral, a Wiener integral and a Poisson jump integral. Logarithmic inversion leads to

$$\begin{aligned} X(t) &= x_0 \exp\left(\int_0^t \left(\left(A(s) - B^2(s)/2\right) ds + B(s)dW(s) + \ln(1+C(s))dP(s)\right)\right) \\ &= x_0 \exp\left(\int_0^t \left(\left(A(s) - B^2(s)/2\right) ds + B(s)dW(s)\right)\right) \prod_{k=1}^{P(t)} \left(1 + C\left(T_k^-\right)\right), \end{aligned} \quad (4.16)$$

where T_k^- is the kth pre-jump time and the last factor comes from conversion of the alternate sum form $\exp\left(\sum_k \ln\left(1 + C(T_k^-)\right)\right)$ to the product form (see the alternate form in (4.9)). Here, the minus superscript in T_k^- would be unnecessary if C were a continuous function of t. The positivity property of the state is preserved as it is in the deterministic and stochastic diffusion cases.

4.2.1. Random Simulated Simple Jump-Diffusion Solution. Applying the forward approximation to the jump-diffusion exponent solution (4.15) with time steps Δt_j for $j = 0 : n_t - 1$, t on $[t_0, t_{n_t}] = [0, t_f]$, with corresponding Wiener increments of ΔW_j and Poisson increments ΔP_j yields the following recursion for Y_j,

$$Y_{j+1} \overset{\text{fwd}}{\simeq} Y_j + (A_j - B_j^2/2)\Delta t_j + B_j \Delta W_j + C_j \Delta P_j\,,$$

so again

$$X_{j+1} \overset{\text{fwd}}{\simeq} \exp\left(Y_{j+1}\right) .$$

The diffusion part can be simulated as in the previous section. Assuming that $\lambda_j \Delta t_j$ is sufficiently small, then the zero-one Poisson jump law can be used to calculate the Poisson increments ΔP_j given a uniform random number generator like `rand` of MATLAB™. A simple method would be that if the jth uniform sampled variate u_j were such that $u_j \geq 1 - \lambda_j \cdot \Delta t_j$ then $\Delta P_j = 1$ is set, else $\Delta P_j = 0$ for the zero one law, approximating the probability of a jump of $1 - (1 - \lambda_j \cdot \Delta t_j) = \lambda_j \cdot \Delta t_j$.

However, for very small $\lambda_j \Delta t_j$ this is not a good example of numerically well-conditioning due to a bias experienced at the beginning and ending points of the uniform distribution. Further, while the theoretical standard uniform distribution is assumed to be on $[0, 1]$, the computer standard uniform distributions are on the interval $[\epsilon, 1 - \epsilon]$, where ϵ is the *machine epsilon*. The machine epsilon is the smallest positive number such that the floating-point representation of $1 + \epsilon$ is greater than one, e.g., for MATLAB™ with double-precision default, $\epsilon = 2^{-53}$. Hence, a better implementation of the zero-one Poisson jump law would be to set $\Delta P_j = 1$ when the uniform sample point is near the center of the distribution, say on $[(1-\lambda_j \cdot \Delta t_j)/2, (1+\lambda_j \cdot \Delta t_j)/2]$; otherwise set $\Delta P_j = 0$.

In the case that the scaled increment $\lambda_j \Delta t_j$ is not sufficiently small compared to unity, then the Poisson inverse method, described by Glasserman [22, Chapter 3], might be used.

A corresponding simplified sample code in MATLAB™ follows for a single sample trajectory, although this could also be coded in Maple™, Mathematica™ or other computing system:

```
function jumpdiffusionpaths
% Sample Jump-Diffusion SDE Test Code
% Scalar, Linear Model:
%   dX(t) = X(t)*(A(T)*dt+B(t)*dW(t)+C(t)dP(t));
%                X(0)=x0; 0<t<tf; C(t)>-1;
global A0 A1 B0 B1 C0 C1 lam0 lam1
nt = 1000; t0 = 0; tf = 2.0; dt = (tf-t0)/nt; %sample input: fixed dt;
x0 = 1; % initial state;
lam0 = 4.0; lam1 = 1.0; % jump rate parms.;
A0 = 0; A1 = +0.3; % plant function coefficient parms.;
B0 = 0; B1 = +0.5; % diffusion coefficient parms.;
C0 = -0.5; C1 = +0.5; % jump-amplitude coefficient parms.;
t = 0:dt:tf; % set time vector in unit base: t(1)=0; t(nt+1)=tf;
sqrtdt = sqrt(dt); % Wiener scaling;
Av = A(t); Bv = B(t); Cv = C(t);% assumes vector subfunctions;
ldt = 0.5*dt*Lam(t); % time-dependent jump-rate;
y = zeros(nt+1); % predeclare for efficiency;
y(1) = log(x0)% log = ln; unit not zero index base;
dw = sqrtdt*randn(nt); % Wiener (0,dt)-normal step matrix;
du = rand(nt); % Poisson zero-one jump law matrix;
dp = zeros(nt); % pre-declare as zero;
for j = 1:nt % Zero-one jump law and exponent update:
        if (du(j)>=(1-ldt(j))/2) && (du(j)<=(1+ldt(j))/2)
          dp(j) = 1;
```

```
 end
        y(j+1,ks) = y(j,ks) + (a(j)-b(j)^2/2)*dt+b(j)*dw(j)...
            +log(1+c(j))*dp(j);
end
x = exp(y); % vector state;
%
% Plot:
plot(t,x),'k-','linewidth',3)
title('Jump-Diffusion Simulated Sample Path');
ylabel('X(t),State'); xlabel('t, Time');
%
function Av = A(t) % must be vector subfunction, e.g., dot-division;
global A0 A1
Av = A0+A1*t./(1+t); % sample, fill-in for each problem;
%End A
function Bv = B(t) % must be vector subfunction, e.g., dot-division;
global B0 B1
Bv = B0+B1*sqrt(t)./(1+t); % sample, fill-in for each problem;
%End B
function Cv = C(t) % must be vector subfunction, e.g.,  dot-division;
global C0 C1
Cv = C0+C1*sqrt(t)./(1+t); % sample, fill-in for each problem;
%End C
function lv = Lam(t) % must be vector subfunction, e.g., dot-division;
global lam0 lam1
lv = lam0+lam1*sin(pi*t); % sample, fill-in for each problem;
%End Lam
%
%%End jumpdiffusionpaths.m
%
```

A sample illustration using a more complicated MATLAB™ code with four sample paths using different random states (seeds) for each of four trajectories is given in Figure 4.1.

The simple jump-diffusion with simple Poisson processes with only state and time-dependent jump-amplitudes is just too simple. Hence, in the next section a jump-diffusion is considered with an additional random argument in the jump-amplitude coefficient.

5. Marked Jump-Diffusion Stochastic Differential Equation. When the jump-amplitudes of the Poisson process are randomly distributed, then the process is called a *marked-jump process* or *compound Poisson process*, where the *independent, identically distributed (IID)* underlying random variables

$$\mathbf{Q} = [Q_i]_{n_p \times 1}$$

are called the *marks* and here they are taken to be mark vectors. Along with the kth random jump-times $T^-_{j,k}$ of the jth Poisson process, the random mark realizations $Q_{j,k}$ are selected by the jth jumping process $P_j(t; Q_j)$ for the kth instance of jth-process jumps; hence an extra mark argument is given to the Poisson process. It is assumed that the jth Poisson process depends only on the jth random mark component Q_j upon relying on the rareness assumption that only one jump is likely as a vector

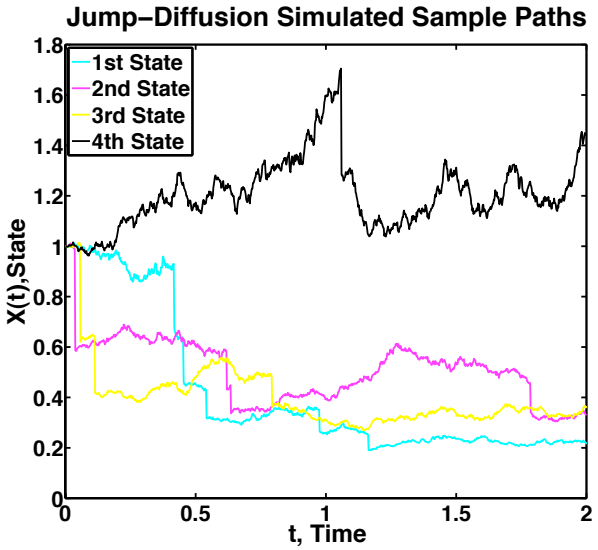

FIG. 4.1. *Simple-jump-diffusion paths for a scalar, linear SDE (4.14) are simulated using MATLAB™ with* $nt = 1000$ *sample points, four* `randn` *normal states for diffusion, four* `rand` *uniform states for the zero-one Poisson jump law and maximum time* $t_f = 2.0$ *starting at* $x_0 = 1.0$.

generalization of the zero-one law; so

$$\mathbf{P}(t;\mathbf{Q}) = [P_i(t;Q_i)]_{n_p\times 1}$$

and the jump-amplitude coefficient also includes the mark vector, but with only one mark component per Poisson component, i.e.,

$$h(\mathbf{X}(t),t,\mathbf{Q}) = [h_{i,j}(\mathbf{X}(t),t,Q_j)]_{n_x\times n_p}.$$

The compound process combination, $h_{i,j}(\mathbf{X}(t),t,Q_j)P_j(t;Q_j)$, is also called a *doubly-stochastic process* or a *marked Poisson process* or a *marked point process* for each i and j [61].

The basic infinitesimal moments of the Poisson process remain the same,

$$\mathrm{E}[\mathbf{P}(t;\mathbf{Q})] = \boldsymbol{\lambda}(t)dt = [\lambda_i(t)dt]_{n_p\times 1}$$

and

$$\mathrm{Var}[P_j(t;Q_j)] = \lambda_i(t)dt \quad \text{for } j = 1:n_p\,,$$

while

$$\mathrm{Jump}[P_j](T_{j,k};Q_j) = 1 \quad \text{for } j = 1:n_p, \quad \text{any integer } k > 1\,.$$

For different representations, the marked Poisson term can have different forms, some more convenient than others. One representation is the *compound Poisson* form which in differential form is

$$h(\mathbf{X}(t),t,\mathbf{Q})d\mathbf{P}(t;\mathbf{Q}) = \sum_{j=1}^{n_p}\sum_{k=P_j(t;Q_j)+1}^{P_j(t;Q_j)+dP_j(t;Q_j)}\left[h_{i,j}\left(\mathbf{X}\left(T_{j,k}^{-}\right),T_{j,k},Q_{j,k}\right)\right]_{n_x\times 1}, \quad (5.1)$$

where $\sum_{k=P_j+1}^{P_j} a_{i,j,k} \equiv 0$ in the reversed sum convention so zero jumps can be included, adding up the jump-amplitudes on $(t, T+dt]$ over all j Poisson components, but the extra jump-time dependence is not too convenient. However, the integral form is a little bit simpler in the jump count sum as follows,

$$\int_0^t h(\mathbf{X}(t), t, \mathbf{Q}) d\mathbf{P}(t; \mathbf{Q}) = \sum_{j=1}^{n_p} \sum_{k=1}^{P_j(t;Q_j)} \left[h_{i,j} \left(\mathbf{X}\left(T_{j,k}^-\right), T_{j,k}, Q_{j,k} \right) \right]_{n_x \times 1} ; \qquad (5.2)$$

$\sum_{k=1}^{0} a_{i,j,k} \equiv 0$ is understood.

Another representation uses *Poisson random measure (PRM)*,

$$\boldsymbol{\mathcal{P}}(\mathbf{dt}, \mathbf{dq}) = [\mathcal{P}_i(\mathbf{dt}, \mathbf{dq}_i)]_{n_p \times 1} ,$$

where $\mathbf{dt} = (t, t+dt]$ is the *time measure*, $\mathbf{dq}_i = (q_i, q_i + dq_i]$ is the *ith mark measure* and $\mathbf{dq} = [\mathbf{dq}_i]_{n_p \times 1}$ is the *vector mark measure*, such that the marked Poisson term (5.1) has the form

$$h(\mathbf{X}(t), t, \mathbf{Q}) d\mathbf{P}(t; \mathbf{Q}) \overset{\text{zol}}{=} \int_{\mathcal{Q}} h(\mathbf{X}(t), t, \mathbf{q}) \mathcal{P}(\mathbf{dt}, \mathbf{dq}) , \qquad (5.3)$$

where $\mathcal{Q}$ is the mark-space set, noting that $\mathcal{P}(\mathbf{dt}, \mathbf{dq})$ is a random measure so the integral is also random. This form has the advantage that the jump-time does not appear and the mark-argument of the jump-amplitude $h(\mathbf{x}, t, \mathbf{q})$ is a deterministic sample value of the mark. The expectation of the PRM jth component is

$$\mathrm{E}[\mathcal{P}_j(\mathbf{dt}, \mathbf{dq}_j)] = \lambda_j(t) dt \Phi_{Q_j}(\mathbf{dq}_j) \overset{\text{gen}}{=} \lambda_j(t) dt \phi_{Q_j}(q_j) dq_j ,$$

where $\Phi_{Q_j}(q_j)$ is the mark distribution and $\phi_{Q_j}(q_j)$ is the mark density, which is always assumed to exist, here, even in the generalized sense as indicated by the symbol $\left\{\overset{\text{gen}}{=}\right\}$, e.g., a discrete distribution. The PRM is connected to the marked Poisson differential process by the equivalence,

$$\int_{\mathcal{Q}} \boldsymbol{\mathcal{P}}(\mathbf{dt}, \mathbf{dq}) \equiv d\mathbf{P}(t; \mathbf{Q}) .$$

The conditional expectation of the infinitesimal Poisson jump-term is

$$\begin{aligned} \mathrm{E}[h(\mathbf{X}(t), t, \mathbf{Q}) dP(t; \mathbf{q}) \,|\, \mathbf{X}(t) = \mathbf{x}] &\overset{\text{IID}}{=} \sum_{j=1}^{n_p} \int_{\mathcal{Q}_j} [h_{i,j}(\mathbf{x}, t, q_j)]_{n_x \times 1} \phi_{Q_j}(q_j) dq_j \lambda_j(t) dt \\ &\equiv \sum_{j=1}^{n_p} \mathrm{E}_{Q_j}[h_{i,j}(\mathbf{x}, t, Q_j)]_{n_x \times 1} \lambda_j(t) dt , \end{aligned}$$

using the IID property of the marks.

Thus the marked-jump-diffusion SDE can be written in the two alternative forms,

$$\begin{aligned} d\mathbf{X}(t) &= \mathbf{f}(\mathbf{X}(t), t) dt + g(\mathbf{X}(t), t) d\mathbf{W}(t) + h(\mathbf{X}(t), t, \mathbf{Q}) d\mathbf{P}(t; \mathbf{Q}) \\ &= \mathbf{f}(\mathbf{X}(t), t) dt + g(\mathbf{X}(t), t) d\mathbf{W}(t) + \int_{\mathcal{Q}} h(\mathbf{X}(t), t, \mathbf{q}) \boldsymbol{\mathcal{P}}(\mathbf{dt}, \mathbf{dq}) , \\ &\qquad 0 \le t \le t_f, \quad \mathbf{X}(0) = \mathbf{x}_0 \in \mathcal{D}_x \subset \mathbb{R}^{n_x}. \end{aligned} \qquad (5.4)$$

The corresponding stochastic integral equation (SIE) in the PRM form is

$$\mathbf{X}(t) = \mathbf{x}_0 + \int_0^t \left(\mathbf{f}(\mathbf{X}(s), s) ds + g(\mathbf{X}(s), s) d\mathbf{W}(s) + \int_{\mathcal{Q}} h(\mathbf{X}(s), s, \mathbf{q}) \boldsymbol{\mathcal{P}}(\mathbf{ds}, \mathbf{dq}) \right) . \qquad (5.5)$$

5.1. Mark Distribution Examples. The distributions of the marks $\mathbf{Q}$ are the underlying distributions for the jump-amplitudes $h(\mathbf{x}, t, \mathbf{Q})$ which can be quite complex compared to the simpler mark distributions, especially if h depends on the random solution trajectory $\mathbf{X}(t)$.

5.1.1. Uniform Mark Distribution. The use of the uniform mark distribution has been emphasized by Hanson et al. [27, 28, 25, 66, 29] for modeling heavy tailed distributions in financial applications. In the scalar case, the time-dependent uniform density on $[a(t), b(t)]$, $a(t) < b(t)$, is

$$\phi_Q(q;t) = \frac{1}{b(t)-a(t)} \left\{ \begin{array}{ll} 1, & a(t) \leq q \leq b(t) \\ 0, & \text{else} \end{array} \right\}, \tag{5.6}$$

with mean $\mu_j(t) = (b(t) - a(t))/2$ and variance $\sigma_j^2(t) = (b(t) - a(t))^2/12$. In finance with linear models, so often the logarithm of the state is considered. Hence, it would be the *log-uniform distribution* which is used. In financial applications and when $a(t) < 0 < b(t)$, $a(t)$ represents the most extreme market crash while $b(t)$ represents the most extreme rally or buying frenzy. The log-uniform distribution results in the fattest tails, since it can be thought of as all tail. In optimal portfolio and consumption problems using jump-diffusions, the uniform has a big advantage due to finite support implying greater trader flexibility for borrowing and short selling as shown by Zhu and Hanson [68].

5.1.2. Normal Mark Distribution. In his pioneering jump-diffusion options paper, Merton [48] used the normal distribution for the marks modeling the stock log-returns underlying the corresponding options that needed pricing. The log-normal jump-amplitude distribution has also been used by Andersen, Benzoni and Lund [2] in their statistical studies of fitting financial return models. The time-dependent normal density has the form,

$$\phi_Q(q;t) = \frac{e^{-(q-\mu_j(t))^2/(2\sigma_j^2(t))}}{\sqrt{2\pi\sigma_j(t)}}\,, \tag{5.7}$$

where $q \in \mathbb{R}$, $\mu_j(t)$ is the specified jump-mark mean and $\sigma_j^2(t)$ is the specified variance. The normal mark variables give a reasonable approximation of the fat tails in finance problems when used for the log-returns, but the infinite range of the mark distribution results in severe restrictions on flexibility of the trader's stock fraction in the optimal portfolio and consumption problem [68]. No such restrictions appear to occur in the option pricing problem.

5.1.3. Double-Exponential Mark Distribution. In option pricing, Kuo [40] and Kuo and Wang [41] have used the double-exponential distribution such that one exponential density is used for positive values and a mirror image of an exponential density is used for the negative values, but with different parameters. Here, time-dependent exponential density is

$$\phi_Q(q;t) = \left\{ \begin{array}{ll} -\dfrac{p_1(t)}{\mu_1(t)} e^{-q/\mu_1(t)}, & q < 0,\ \mu_1(t) < 0 \\ +\dfrac{p_2(t)}{\mu_2(t)} e^{-q/\mu_2(t)}, & q \geq 0,\ \mu_2(t) > 0 \end{array} \right\}, \tag{5.8}$$

where $q \in \mathbb{R}$, the one-sided means $\mu_1(t)$ and $\mu_2(t)$ are specified, while the one-sided probabilities satisfy $p_1(t) + p_2(t) = 1$, so the composite density is properly normalized.

Earlier, Ramzani and Zeng [54] applied the maximum likelihood method to fit a jump-diffusion model with double-exponential log-jump-amplitudes for security prices. The double-exponential distribution has qualitatively similar properties as its fellow Laplace distribution, i.e., the normal distribution, including being restrictive in a trader's flexibility due to the infinite range of the distribution in the optimal portfolio problem; no such restrictions appear to occur in the option pricing application, as long as the option is not part of an optimal portfolio problem.

5.1.4. Double-Uniform Mark Distribution. In analogy with the use of the double-exponential mark distribution in finance, Zhu [67] for option pricing and Zhu and Hanson [68] for optimal portfolio problems created the double-uniform mark distribution. Since crashes and rallies seem to result from different psychological behaviors, it is a good idea to separate the negative and positive jumps. The time-dependent double-uniform distribution is given by

$$\phi_Q(q;t) = \left\{ \begin{array}{ll} \frac{p_1(t)}{|a|(t)}, & a(t) < q < 0,\ a(t) < 0 \\ \frac{p_2(t)}{b(t)}, & 0 \leq q \leq b(t),\ b(t) > 0 \\ 0, & \text{else} \end{array} \right\}, \tag{5.9}$$

where the one-sided means $|a|(t)/2$ and $b(t)/2$ are specified, while the one-sided probabilities satisfy $p_1(t)+p_2(t) = 1$. This mark distribution with its finite support allows flexibility in short-selling and borrowing in the optimal portfolio and consumption problem, unlike the case of unbounded support for the normal and double-exponential mark distributions [68]. The finite range jump-amplitude distribution also makes sense since the New York Stock Exchange instituted a series of market circuit breakers [6] in 1988 to slow down and completely stop trading in stages to avoid a market crash like the one in 1987.

5.2. Change of Variables and Marked Jump-Diffusion Chain Rule. Letting $\mathbf{Y}(t) = \mathbf{F}(\mathbf{X}(t), t)$ again be the change of variables. Then following the decomposition into continuous and jump parts in the simple jump case (4.13), the marked jump case follows in two alternative forms,

$$\begin{aligned} d\mathbf{Y}(t) &\overset{\underset{\text{zol}}{\text{dt}}}{=} \left(\mathbf{F}_t + \mathbf{f}^\top \nabla_x[\mathbf{F}] + \tfrac{1}{2}\left(gR^{(w)}g^\top\right) : \nabla_x\left[\nabla_x[\mathbf{F}]\right]\right) dt + \nabla_x^\top[\mathbf{F}] g d\mathbf{W}(t) \\ &\quad + \sum_{j=1}^{n_p} \Big(\mathbf{F}\Big(\mathbf{X}(t) + [h_{i,j}(\mathbf{X}(t), t, Q_j)]_{n_x \times 1}, t \Big) - \mathbf{F}(\mathbf{X}(t), t) \Big)\, dP_j(t; Q_j) \\ &= \left(\mathbf{F}_t + \mathbf{f}^\top \nabla_x[\mathbf{F}] + \tfrac{1}{2}\left(gR^{(w)}g^\top\right) : \nabla_x\left[\nabla_x[\mathbf{F}]\right]\right) dt + \nabla_x^\top[\mathbf{F}] g d\mathbf{W}(t) \\ &\quad + \sum_{j=1}^{n_p} \int_{\mathcal{Q}_j} \Big(\mathbf{F}\Big(\mathbf{X}(t) + [h_{i,j}(\mathbf{X}(t), t, q_j)]_{n_x \times 1}, t \Big) - \mathbf{F}(\mathbf{X}(t), t) \Big)\, \mathcal{P}_j(\mathbf{dt}, \mathbf{dq}_j)\,, \end{aligned} \tag{5.10}$$

where again it is understood that the Poisson jump process $P_j(t)$ picks out the pre-jump time $T_{j,k}^-$ of the state process $\mathbf{X}(t)$ if there is the kth jump at time $T_{j,k}$ of the jth Poisson process. Note that the last line of (5.10) can also be written

$$\text{Jump}[\mathbf{Y}](t) = \sum_{j=1}^{n_p} \text{Jump}_j[\mathbf{F}](\mathbf{X}(t), t),$$

in compact form, where

$$\text{Jump}_j[\mathbf{F}](\mathbf{X}(t),t) \equiv \int_{\mathcal{Q}_j} \Big(\mathbf{F}\Big(\mathbf{X}(t) + \widehat{\mathbf{h}}_j(\mathbf{X}(t),t,q_j),t \Big) - \mathbf{F}(\mathbf{X}(t)t) \Big) \boldsymbol{\mathcal{P}}_j(\mathbf{dt},\mathbf{dq}_j)$$

and

$$\widehat{\mathbf{h}}_j(\mathbf{x},t,q_j) \equiv [h_{i,j}(\mathbf{x},t,q_j)]_{n_x \times 1}\,.$$

5.3. Scalar, Linear, Marked Jump-Diffusion SDE Example. Consider the scalar, linear marked-jump-diffusion model with $n_x = 1$, $n_w = 1$ and $n_p = 1$, using the two Poisson forms,

$$\begin{aligned} dX(t) = &\quad X(t) \cdot (A(t)dt + B(t)dW(t) + C(t;Q)dP(t;Q)) \\ = &\quad X(t) \cdot \Big(A(t)dt + B(t)dW(t) + \int_{\mathcal{Q}} C(t;q)\mathcal{P}(\mathbf{dt},\mathbf{dq}) \Big)\,, \end{aligned} \tag{5.11}$$

with $X(0) = x_0 > 0$, where $A(t)$, $B(t)$ and $C(t;Q)$ are given integrable functions. Again, the essential state positivity condition is that

$$C(t;Q) > -1\,,$$

since otherwise the first jump would take the state out of positive values. In the case of one jump when $dP(t;Q) = 1$ and $C(t;Q) > -1$, the new jump is

$$X(t) + \text{Jump}[X](t) = X(t)(1 + C(t;Q)) > 0\,,$$

given jump values for $(t;Q)$. Again, let

$$Y(t) = F(X(t),t) = \ln(X(t))$$

be the logarithmic state transformation, with partial derivatives

$$F_t(x,t) = 0\,, \quad F_x(x,t) = 1/x\,, \quad F_{x,x}(x,t) = -1/x^2\,,$$

assuming $x > 0$ and with the conditional jump,

$$\begin{aligned} \text{Jump}[Y](t) &= (\ln(X(t) + C(t;Q)X(t)) - \ln(X(t)))\, dP(t;Q) \\ &= \ln(1 + C(t;Q))dP(t;Q)\,, \end{aligned}$$

provided $X(t) > 0$ and $C(t;Q) > -1$ while the zero-one law is in effect. Then, the transformed jump-diffusion SDE by the chain rule (5.10) becomes state-independent by adding the jump-change to the continuous diffusion-changes,

$$dY(t) \underset{\text{zol}}{\overset{\text{dt}}{=}} \big(A(t) - B^2(t)/2 \big)\, dt + B(t)dW(t) + \ln(1 + C(t;Q))dP(t;Q)\,,$$

to dt-precision plus the zero-one jump law, so that immediate integration yields

$$\begin{aligned} Y(t) &\overset{\text{ims}}{=} Y(0) + \int_0^t \big(\big(A(s) - B^2(s)/2 \big)\, ds + B(s)dW(s) + \ln(1 + C(s;Q))dP(s;Q) \big) \\ &= Y(0) + \int_0^t \Big(\big(A(s) - B^2(s)/2 \big)\, ds + B(s)dW(s) \\ &\quad + \int_{\mathcal{Q}} \ln(1 + C(s;q))\mathcal{P}(\mathbf{ds},\mathbf{dq}) \Big), \end{aligned} \tag{5.12}$$

the sum of a Riemann integral, a Wiener integral and a Marked Poisson jump integral. Switching notation and logarithmic inversion lead to

$$\begin{aligned} X(t) &= x_0 \exp\left(\int_0^t \left(\left(A(s) - B^2(s)/2 \right) ds + B(s) dW(s) + \ln(1 + C(s;Q)) dP(s;Q) \right) \right) \\ &= x_0 \exp\left(\int_0^t \left(\left(A(s) - B^2(s)/2 \right) ds + B(s) dW(s) \right) \right) \prod_{k=1}^{P(t;Q)} \left(1 + C\left(T_k^-; Q_k \right) \right), \end{aligned} \tag{5.13}$$

where T_k^- is the kth pre-jump time and Q_k is the random jump-amplitude mark. The last factor comes from the law of exponentials of logarithmic exponents using the conversion

$$\exp\left(\sum_{k=1}^{P(t;Q)} \ln\left(1 + C(T_k^-, Q_k) \right) \right) = \prod_{k=1}^{P(t;Q)} \left(1 + C\left(T_k^-; Q_k \right) \right).$$

The positivity property of the state is preserved as it is in the prior cases.

Using the Itô forward integration expansion and mean square convergence, Hanson [25, Chapter 5] shows the following theorem on jump-diffusion exponential expectations, here reformulated:

THEOREM 5.1. ***Exponential Expectations:***

$$\mathrm{E}\left[\exp\left(\int_0^t (B(s) dW(s) - \frac{1}{2} B^2(s) ds) \right) \right] = 1 \tag{5.14}$$

and

$$\begin{aligned} &\mathrm{E}\left[\exp\left(\int_0^t \int_{\mathcal{Q}} \ln(1 + C(s;q)) \mathcal{P}(\mathbf{dt}, \mathbf{dq}) \right) \right] \\ &\qquad = \exp\left(\int_0^t \int_{\mathcal{Q}} C(s;q) \phi_Q(q) dq \lambda(s) ds \right). \end{aligned} \tag{5.15}$$

Thus,

$$\begin{aligned} \mathrm{E}[X(t)] &= x_0 \exp\left(\int_0^t \left(A(t) + \lambda(s) \mathrm{E}_Q[C(s;Q)] \right) ds \right) \\ &= x_0 \exp\left(\int_0^t \mathrm{E}[dX(s)/X(s) \,|\, X(s)] \right), \end{aligned} \tag{5.16}$$

where

$$\mathrm{E}_Q[C(t;Q)] = \int_{\mathcal{Q}} C(t;q) \phi_Q(q) dq \,.$$

Similarly,

$$\mathrm{Var}[X(t)] \stackrel{\mathrm{dt}}{=} \mathrm{E}^2[X(t)] \left(\exp\left(\int_0^t \mathrm{Var}[dX(s)/X(s) \,|\, X(s)] \right) - 1 \right). \tag{5.17}$$

5.3.1. Random Simulated Mark Jump-Diffusion Solution. Using the exponent solution (5.12), the forward, discrete, recursive form of this solution is

$$Y_{j+1} \overset{\text{fwd}}{\simeq} Y_j + \left(A_j - B_j^2/2\right)\Delta t + B_j \Delta W_j + \ln(1+C_j)\Delta P_j\,, \tag{5.18}$$

where $C_j = C(t_j; Q_j)$ and the state solution is given by $X_{j+1} = \exp(Y_{j+1})$ as before.

Since the marks, Q, are the underlying jump-amplitude random variables and it is difficult if not impossible to separate the *outlier* jumps from a background of normal fluctuations [1], it is best to keep the selection of Q simple. So let

$$Q = \ln(1 + C(t;Q)) \quad \text{or} \quad C(t;Q) = \exp(Q) - 1\,, \tag{5.19}$$

and the discrete form (5.18) becomes

$$Y_{j+1} \overset{\text{fwd}}{\simeq} Y_j + \left(A_j - B_j^2/2\right)\Delta t + B_j \Delta W_j + Q_j \Delta P_j\,. \tag{5.20}$$

This may seem to overly restrict the mark-jump-amplitude relation to a time-independent one, but the distribution of the marks still can be time-dependent, such as the use of time-dependent parameters in the mark-distribution examples of Subsection 5.1. Further, the simplest selection of a mark distribution for the logarithm of the state is the *uniform distribution* (5.6) with $Q \in [a(t), b(t)]$, although the other listed example distributions could be used if desired. For instance, with constants a and b in MATLAB™,

$$\texttt{Q} = \texttt{a} + (\texttt{b} - \texttt{a}) * \texttt{rand}(\texttt{1}, \texttt{n});$$

provided a jump has occurred, is used for uniform random simulations on (a, b), where `rand` is the uniform random number generator on $(0, 1)$, approximately.

A corresponding simplified sample code in MATLAB™ for a single sample path follows:

```
function markjumpdiffusionpaths
% Sample Mark Jump-Diffusion SDE Test Code
% Scalar, Linear Model:
%   dX(t) = X(t)*(A(T)*dt+B(t)*dW(t)+C(t;Q)*dP(t;Q));
%             X(0)=x0; 0<t<tf; C(t;Q)=exp(Q)-1 > -1; a(t)<Q<b(t);
global A0 A1 B0 B1 a0 a1 b0 b1 lam0 lam1
nt = 1000; t0 = 0; tf = 2.0; dt = (tf-t0)/nt; %sample input;
x0 = 1; % initial state;
lam0 = 4.0; lam1 = 1.0; % jump rate parms.;
A0 = 0; A1 = +0.3; % plant function coefficient parms.;
B0 = 0; B1 = +0.5; % diffusion coefficient parms.;
a0 = -0.5; a1 = -0.25; b0 = +0.5; b1 = +0.20; % uniform dist. parms.;
t = 0:dt:tf; % set time vector in unit base: t(1)=0; t(nt+1)=tf;
sqrtdt = sqrt(dt); % Wiener scaling;
Av = A(t); Bv = B(t); Cv = C(t);% assumes vector subfunctions;
av = a(t); bv = b(t); % assumes vector subfunctions;
ldt = 0.5*dt*Lam(t); % time-dependent jump-rate;
y = zeros(nt+1); % predeclare for efficiency;
y(1) = log(x0)% log = ln; unit not zero index base;
dw = sqrtdt*randn(nt); % Wiener (0,dt)-normal step matrix;
du = rand(nt); % Poisson zero-one jump law matrix;
```

```
dp = zeros(nt); % pre-declare as zero;
uq = rand(nt); % Could change 'state', but disjoint from du;
q = zeros(nt);
for j = 1:nt % Zero-one jump law and exponent update:
       q(j) = av(j)+(bv(j)-av(j))*uq(j);
        if (du(j)>=(1-ldt(j))/2) && (du(j)<=(1+ldt(j))/2)
          dp(j) = 1;
 end
        y(j+1,ks) = y(j,ks) + (a(j)-b(j)^2/2)*dt+b(j)*dw(j)...
            +q(j)*dp(j);
end
x = exp(y); % vector state;
%
% Plot:
plot(t,x),'k-','linewidth',3)
title('Jump-Diffusion Simulated Sample Path');
ylabel('X(t),State'); xlabel('t, Time');
%
function Av = A(t) % must be vector subfunction, e.g., dot-division;
global A0 A1
Av = A0+A1*t./(1+t); % sample, fill-in for each problem;
%End A
function Bv = B(t) % must be vector subfunction;
global B0 B1
Bv = B0+B1*sqrt(t)./(1+t); % sample, fill-in for each problem;
%End B
function lv = Lam(t) % must be vector subfunction;
global lam0 lam1
lv = lam0+lam1*sin(pi*t); % sample, fill-in for each problem;
%End Lam
function av = a(t) % must be vector subfunction;
global a0 a1
av = a0+a1*sin(pi*t); % sample, fill-in for each problem;
%End a
function bv = b(t) % must be vector subfunction;
global b0 b1
bv = b0+ b1*sin(pi*t); % sample, fill-in for each problem;
%End b
%
%%End markjumpdiffusionpaths.m
%
```

A sample illustration using a more complicated MATLAB™ code with four sample paths using different random states (seeds) is given in Figure 5.1.

6. Stochastic Optimal Control for Mark Jump-Diffusions: Stochastic Dynamic Programming. Studying stochastic dynamics is only half the problem; the other half is controlling the dynamics. Almost all appliances and industrial machines have control systems: DVD-players, TVs, computers, hard-drives, satellites, automobiles, airplanes, ships, trains, etc.

Let $\mathbf{U}(t)$ be the control vector in $\mathcal{D}_u$ at time t and usually found in the *plant or*

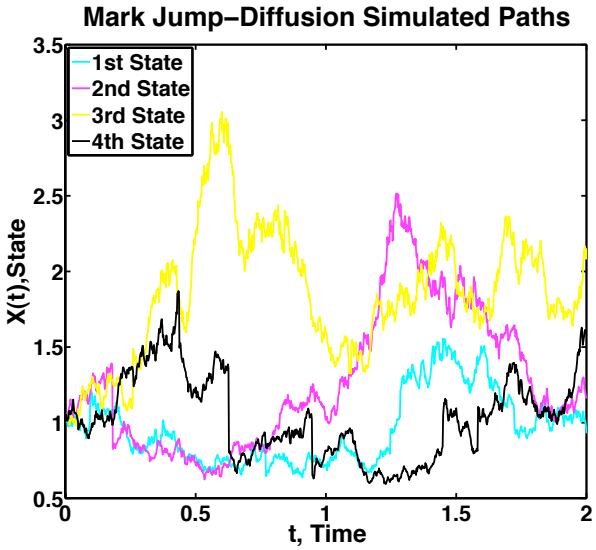

FIG. 5.1. *Mark-jump-diffusion paths for a scalar, linear SDE (5.11) are simulated using MATLAB™ with* $nt = 1000$ *sample points, four* `randn` *normal states for diffusion, four* `rand` *uniform states of double length for the zero-one Poisson jump law as well as for the mark distribution, and maximum time* $t_f = 2.0$ *starting at* $x_0 = 1.0$.

deterministic function, so

$$\mathbf{f}(\mathbf{X}(t), t) \longrightarrow \mathbf{f}(\mathbf{X}(t), \mathbf{U}(t), t)\,.$$

When possible, an optimal, feedback control is preferred, i.e.,

$$\mathbf{U}^*(t) = \widehat{\mathbf{U}}(\mathbf{X}(t), t)\,,$$

where the asterik $(*)$ denotes an optimal solution and the dependence on $\mathbf{X}(t)$ means state-control feedback, i.e., $\mathbf{X} \longleftrightarrow \mathbf{U}$.

Consider the control-dependent jump-diffusion SDE,

$$d\mathbf{X}(t) = \mathbf{f}(\mathbf{X}(t), \mathbf{U}(t), t)dt + g(\mathbf{X}(t), t)d\mathbf{W}(t) + \int_{\mathcal{Q}} h(\mathbf{X}(t), t, \mathbf{q})\boldsymbol{\mathcal{P}}(\mathbf{dt}, \mathbf{dq})\,, \qquad (6.1)$$

as previously with $\mathbf{X}(0) = \mathbf{x}_0$ and $t \in [0, t_f]$, but with the control vector $\mathbf{U}(t)$ in the plant function. The control could also appear in the coefficients $g(\mathbf{x}, t)$ for diffusion or $h(\mathbf{x}, t, \mathbf{q})$ for marked jumps, but that adds too much complexity for this introductory survey and excludes certain important canonical problems such as linear-quadratic problems.

Since the control has been introduced as another variable, another condition is needed to determine it and that is called the **objective** of the optimization, which is to minimize costs or maximize the utility of wealth or profit. Here we will focus on the objective for minimization for definiteness since the difference with the maximum is trivial, i.e., $\max[f(x)] = -\min[-f(x)]$. So consider the scalar objective,

$$V[\mathbf{X}, \mathbf{U}](t) = \int_t^{t_f} \mathcal{C}(\mathbf{X}(s), \mathbf{U}(s), s)ds + S(\mathbf{X}(t_f), t_f)\,, \qquad (6.2)$$

where $V[\mathbf{X},\mathbf{U}](t) \in \mathbb{R}$, $t \in [0,t_f]$, $\mathcal{C}(\mathbf{x}(t),\mathbf{u}(t),t)$ is the integrable instantaneous or running costs and $S(\mathbf{X}(t_f),t_f)$ is the final or salvage costs. As $t \to t_f$ then $V[\mathbf{X},\mathbf{U}](t_f) = S(\mathbf{X}(t_f),t_f)$ for all $\mathbf{X}(t_f) \in \mathbb{R}^{n_x}$.

Direct optimization of the objective (6.2) is ill-posed since (6.2) is a stochastic problem with too many local optima. This suggests the smoothing of the objective by taking the expectation, so the following conditional expectation function is considered as a practical substitute for (6.2) with some flexibility in state and control dependence,

$$\overline{V}(\mathbf{x},\mathbf{u},t) = \underset{(\boldsymbol{W},\boldsymbol{P})(t,t_f]}{\mathrm{E}} \left[V[\mathbf{X},\mathbf{U}](t) \,|\, \mathbf{X}(t) = \mathbf{x}, \mathbf{U}(t) = \mathbf{u} \right], \tag{6.3}$$

then finally taking the optimum of the expected objective (6.3),

$$\overline{V}^*(\mathbf{x},t) = \min_{\mathbf{U}(t,t_f]} \left[\overline{V}(\mathbf{x},\mathbf{u},t) \right], \tag{6.4}$$

over the admissible set $\mathbf{U}(t,t_f] = \{\mathbf{U}(s) \,|\, t < s \le t_f\}$.

6.1. Bellman's Principle of Optimality: Recursive Form. Stochastic dynamic programming according to Richard Bellman [12] depends on the ability to optimize over disjoint time-increments in backward time. This backward decomposition in time is valid for many problems, particularly deterministic problems. The extension to stochastic problems holds for Markov processes due to the independent increment property. In particular, the principle of optimality for stochastic processes depends on the following analytical decompositions [25, Chapter 6]:

- *Additive Decomposition of Cost Integrals*: Let Δt be a time-increment, such that $0 < \Delta t < t_f - t$ and

$$\begin{aligned} V[\mathbf{X},\mathbf{U}](t) &= \int_t^{t_f} \mathcal{C}(\mathbf{X}(s),\mathbf{U}(s),s)ds + S(\mathbf{X}(t_f),t_f) \\ &= \int_t^{t+\Delta t} \mathcal{C}(\mathbf{X}(s),\mathbf{U}(s),s)ds \\ &\quad + \int_{t+\Delta t}^{t_f} \mathcal{C}(\mathbf{X}(s),\mathbf{U}(s),s)ds + S(\mathbf{X}(t_f),t_f) \\ &= \int_t^{t+\Delta t} \mathcal{C}(\mathbf{X}(s),\mathbf{U}(s),s)ds + V[\mathbf{X},\mathbf{U}](t+\Delta t), \end{aligned} \tag{6.5}$$

  yielding a preliminary recursion based upon the usual additive property of integrals.
- *Multiplicative Decomposition of Iterated Expectations*: By applying iterated expectations,

$$\begin{aligned} \overline{V}(\mathbf{x},\mathbf{u},t) &= \underset{(\boldsymbol{W},\boldsymbol{P})(t,t_f]}{\mathrm{E}} \left[V[\mathbf{X},\mathbf{U}](t) \,|\, (\mathbf{X},\mathbf{U})(t) \right] \\ &= \underset{(\boldsymbol{W},\boldsymbol{P})(t,t+\Delta t]}{\mathrm{E}} \left[\int_t^{t+\Delta t} \mathcal{C}(\mathbf{X}(s),\mathbf{U}(s),s)ds + \overline{V}(\mathbf{x},\mathbf{u},t+\Delta t) \right. \\ &\qquad\qquad \left. \Big| \, (\mathbf{X},\mathbf{U})(t) \right], \end{aligned} \tag{6.6}$$

  giving the secondary recursion step.

- *Multiplicative Decomposition of minimum (or maximum)*: Assuming piecewise optimization over time, the principle of optimality is found:

$$\begin{aligned}\overline{V}^*(\mathbf{x},t) = & \min_{\mathbf{U}(t,t_f]}\left[\overline{V}(\mathbf{x},\mathbf{u},t)\right] \\ = & \min_{\mathbf{U}(t,t+\Delta t]}\Bigg[\mathop{\mathrm{E}}_{(\boldsymbol{W},\boldsymbol{P})(t,t+\Delta t]}\Bigg[\int_t^{t+\Delta t}\mathcal{C}(\mathbf{X}(s),\mathbf{U}(s),s)ds \\ & +\overline{V}^*(\mathbf{x}+\Delta\mathbf{X}(t),t+\Delta t)\,\Big|\,(\mathbf{X},\mathbf{U})(t)=(\mathbf{x},\mathbf{u})\Bigg]\Bigg],\end{aligned} \tag{6.7}$$

  where the last term is the shifted optimal, expected objective.

REMARKS 6.1. *When considering more general stochastic optimization problems, knowledge of the details of this sequence facilitates the modification of the principle of optimality (6.7) to the new problem. In addition, it must be emphasized that the optimization decomposition (6.7) is an assumption of the principle, unlike the integral (6.5) and expectation (6.6) decompositions. However, (6.7) holds in many cases, but there are economic counter examples such as those found by Rust [56].*

6.2. PDE of Stochastic Dynamic Programming. Application of the jump-diffusion stochastic chain rule, the forward limit as the time-increment $\Delta t \to 0^+$ with $\Delta t \to dt$ in dt-precision and Taylor approximations lead to a preliminary step in deriving the Bellman's PDE of stochastic dynamic programming,

$$\begin{aligned}\overline{V}^*(\mathbf{x},t) \overset{\substack{\mathrm{dt}\\ \mathrm{zol}}}{=} \min_{\mathbf{u}(t)}\Bigg[& \mathop{\mathrm{E}}_{(\boldsymbol{dW},\boldsymbol{dP})(t)}\Bigg[\mathcal{C}(\mathbf{x},\mathbf{u},t)dt+\overline{V}^*(\mathbf{x},t)+\overline{V}^*_t(\mathbf{x},t)dt \\ & +d_{\mathrm{cont}}\left[\mathbf{X}^\top\right](t)\nabla_x\left[\overline{V}^*\right](\mathbf{x},t)+\tfrac{1}{2}(gd\mathbf{W})^\top\nabla_x\left[\nabla_x^\top\left[\overline{V}^*\right]\right](\mathbf{x},t)(gd\mathbf{W}) \\ & +\sum_{j=1}^{n_p}\int_{\mathcal{Q}_j}\Big(\overline{V}^*\Big(\mathbf{x}+\left[h_{i,j}(\mathbf{x},t,q_j)\right]_{n_x\times 1},t\Big)-\overline{V}^*(\mathbf{x},t)\Big)\mathcal{P}(\mathbf{dt},\mathbf{dq}_j) \\ & \Bigg|\,(\mathbf{X},\mathbf{U})(t)=(\mathbf{x},\mathbf{u})\Bigg]\Bigg].\end{aligned} \tag{6.8}$$

Upon taking the indicated expectations along with the definition of $d_{\mathrm{cont}}\left[\mathbf{X}^\top\right]$, canceling the $\overline{V}^*(\mathbf{x},t)$ on each side of the equation, canceling the remaining common dt, letting $v(\mathbf{x},t)=\overline{V}^*(\mathbf{x},t)$ for simplicity and thus completing a quick derivation of the PDE of stochastic dynamic programming (SDP) leads to

$$\begin{aligned}0= & \ v_t(\mathbf{x},t)+\tfrac{1}{2}(gg^\top):\nabla_x\left[\nabla_x^\top[v]\right](\mathbf{x},t) \\ & +\sum_{j=1}^{n_p}\int_{\mathcal{Q}_j}\Big(v\Big(\mathbf{x}+\left[h_{i,j}(\mathbf{x},t,q_j)\right]_{n_x\times 1},t\Big)-v(\mathbf{x},t)\Big)\phi_{Q_j}(\mathbf{q}_j) \\ & +\min_{\mathbf{u}}[\mathcal{C}(\mathbf{x},\mathbf{u},t)+\mathbf{f}(\mathbf{x},\mathbf{u},t)\nabla_x[v](\mathbf{x},t)]\ .\end{aligned} \tag{6.9}$$

Consider the following properties of the PDE of SDP:

PROPERTIES 6.2.

- *The PDE of SDP (6.9) is a backward, final value problem with final condition as* $t\to t_f^-$,

$$v(\mathbf{x},t_f)=S(\mathbf{x},t_f)$$

  for any $\mathbf{x}\in\mathcal{D}_x$.

- *The boundary and other conditions are state domain $\mathcal{D}_x$ and process dependent, e.g., $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{0}$ is reflecting.*
- *The PDE of SDP (6.9) is also called the HJBE or Hamilton-Jacobi-Bellman equation, since it resembles the Hamilton-Jacobi equations of mechanics, i.e.,*

$$0 = v_t(\mathbf{x}, t) + \mathcal{F}_x[v](\mathbf{x}, t) = v_t(\mathbf{x}, t) + \min_{\mathbf{u}}[\mathcal{H}(\mathbf{x}, \mathbf{u}, t)]\,,$$

  where $\mathcal{H}(\mathbf{x}, \mathbf{u}, t)$ is called the Hamiltonian or pseudo-Hamiltonian in control.
- *For jump-diffusion control, the HJBE is a functional PDE (FPDE) and is also called a partial integro-differential equation (PIDE) due to the jump integral and the control optimization.*
- *The HJBE is a scalar-valued PIDE in 3D-space plus time, but the solution set is a scalar optimal value $v(\mathbf{x}, t)$ plus a vector optimal control*

$$\mathbf{u}^*(\mathbf{x}, t) = \operatorname*{argmin}_{\mathbf{u}}[\mathcal{H}(\mathbf{x}, \mathbf{u}, t)]\,,$$

  called a feedback form of control. The solution is a mapping from the space-time set $\{\mathbf{x}, t\} \in \mathbb{R}^{n_x+1}$ to the optimal value-control set $\{v(\mathbf{x}, t), \mathbf{u}^*(\mathbf{x}, t)\} \in \mathbb{R}^{n_u+1}$.
- *The optimal control in absence of constraints is called the regular optimal control or just regular control $u^{(\mathrm{reg})}(\mathbf{x}, t)$. It is often used in part to find the global optimal control in the presence of constraints by using the regular control when the constraints are automatically satisfied and then calculating the global optimal control with the active constraints to get $u^*(\mathbf{x}, t)$. Thus, $u^*(\mathbf{x}, t)$ is the composite of the regular control and the constraints when they are active or forced. In this case the HJBE is written*

$$0 = v_t(\mathbf{x}, t) + \mathcal{H}^*(\mathbf{x}, t)\,,$$

  where

$$\mathcal{H}^*(\mathbf{x}, t) \equiv \mathcal{H}(\mathbf{x}, \mathbf{u}^*(\mathbf{x}, t), t) = \max_{\mathbf{u}}[\mathcal{H}(\mathbf{x}, \mathbf{u}, t)]\,,$$

  provided the minimum exists.
- *The diffusion coefficient could also depend on the control, i.e., $g = g(\mathbf{x}, \mathbf{u}, t)$, and so could the jump-amplitude coefficient, i.e., $h = h(\mathbf{x}, \mathbf{u}, t, \mathbf{q})$. However, this kind of dependence introduces a great deal of complexity to the more simple stochastic optimal control problems, such as those in the next section. Typically, it is the plant or deterministic function, $\mathbf{f}(\mathbf{x}, \mathbf{u}, t)$, that is controllable.*

6.3. Computational Complexity: Bellman's Curse of Dimensionality. Computational methods for stochastic dynamic programming suffer from an exponential order dimensional complexity that is very similar to that for parabolic PDEs of physical diffusion problems when numerical procedures such as finite differences and finite elements are used. Thus, the computational difficulties are associated with second and higher order PDEs even for a moderate number of space dimensions. For the computational dynamic programming problem, the extreme computational demands are known as *Bellman's curse of dimensionality* [13].

For simplicity, only the finite difference with even step sizes version will be briefly presented. Starting with the n_x-dimensional spatial vector $\mathbf{x} = [x_i]_{n_x}$, let the ith component be represented by

$$X_{i,j_i} = x_{i,0} + (j_i - 1)\Delta X_i \,,$$

for $j_i = 1 : N_i$ nodes in $i = 1 : n_x$ dimensions, where $\Delta X_i = (X_{i,N_i} - X_{i,1})/(N_i - 1)$ is the fixed ith step size. Let $\mathbf{X_j} = [X_{i,j_i}]_{n_x \times 1}$ represent a single, local vector position in space. Let $\mathbf{j} = [j_i]_{n_x \times 1}$ be the local index and $J = [j_1, j_2, \ldots, j_{n_x}]_{N_1 \times N_2 \times \cdots \times N_{n_x}}$ be the $N = \prod_{i=1}^{n_x} N_i$ dimensional global index array. Let the time variable be discretized as $T_k = (k-1)\Delta T$ for $k = 1 : M_t$ equally-spaced time nodes, where $\Delta T = t_f/(M_t - 1)$, though k will not figure into the space dimensional complexity. Similarly, let the mark variable be discretized as values q_ℓ for $\ell = 1 : N_q$ using values appropriate for the mark density, such as the nodes of general Gaussian quadrature [63] developed for jump integrals, but the ℓ will also be ignored in the count.

Using only central finite differences (CFDs) for the state or spatial variable with the time index k and mark index ℓ fixed but with backward march in time appropriate for the final value problem, the value terms have the following representation:

$$\begin{aligned}
v(\mathbf{X_j}, T_k) \to\ & V_{J,k} \equiv [V_{j_1,j_2,\ldots,j_{n_x},k}]_{N_1 \times N_2 \times \cdots \times N_{n_x}} \,, \\
v_t(\mathbf{X_j}, T_k) \to\ & (V_{J,k+1} - V_{J,k})/(-\Delta T) \,, \\
\nabla_x[v](\mathbf{X_j}, T_k) \to\ & \mathbf{DV}_{J,k} \equiv [\mathrm{DV}_{i,j_1,\ldots,j_{n_x},k}]_{n_x \times N_1 \times \cdots \times N_{n_x}} \\
& = \big[\big(V_{j_1+\delta_{i,1},\ldots,j_{n_x}+\delta_{i,n_x},k} \\
& \quad - V_{j_1-\delta_{i,1},\ldots,j_{n_x}-\delta_{i,n_x},k}\big)/\Delta X_i\big]_{n_x \times N_1 \times \cdots \times N_{n_x}} \,, \\
\nabla_{\mathbf{x}}\left[\nabla_{\mathbf{x}}^\top[v]\right](\mathbf{X_j}, T_k) \to\ & \mathrm{DDV}_{J,k} \equiv [\mathrm{DDV}_{i,j,j_1,\ldots,j_{n_x},k}]_{n_x \times n_x \times N_1 \times \cdots \times N_{n_x}} \,, \\
u^{(\mathrm{reg})}(\mathbf{X_j}, T_k) \to\ & \mathbf{UR}_{J,k} \equiv [\mathrm{UR}_{i,j_1,\ldots,j_{n_x},k}]_{n_x \times N_1 \times \cdots \times N_{n_x}} \,, \\
u^*(\mathbf{X_j}, T_k) \to\ & \mathbf{US}_{J,k} \equiv [\mathrm{US}_{i,j_1,\ldots,j_{n_x},k}]_{n_x \times N_1 \times \cdots \times N_{n_x}} \\
& = [\min(\mathrm{UMAX}_i, \max(\mathrm{UMIN}_i, \\
& \quad \mathrm{UR}_{i,j_1,\ldots,j_{n_x},k}))]_{n_x \times N_1 \times \cdots \times N_{n_x}} \,, \\
v\left(\mathbf{X_j} + \widehat{\mathbf{h}}_{J,k,\ell}, T_k\right) \to\ & \mathrm{VH}_{J,k,\ell} \equiv [\mathrm{VH}_{i,j_1,\ldots,j_{n_x},k,\ell}]_{n_x \times N_1 \times \cdots \times N_{n_x}}
\end{aligned} \tag{6.10}$$

where $\delta_{i,j}$ is the Kronecker delta and $\widehat{\mathbf{h}}_{J,k,\ell} = h_{J,\ell}(\mathbf{X}_J, T_k, q_\ell)$, while $\mathrm{UMIN}_i = U_i^{(\min)}$ and $\mathrm{UMAX}_i = U_i^{(\max)}$ are the assumed hypercube constraints on the control for dimensions $i = 1 : n_x$.

The diffusion Hessian array has the most complexity counting the number of

elements and is generally nonsymmetric, so is given by

$$\begin{aligned}
\mathrm{DDV}_{J,k} \equiv\ & \left[\mathrm{DDV}_{i,j,j_1,\ldots,j_{n_x},k}\right]_{n_x\times n_x\times N_1\times\cdots\times N_{n_x}} \\
=\ & \left[\left(V_{j_1+\delta_{i,1},\ldots,j_{n_x}+\delta_{i,n_x},k}-2V_{j_1,\ldots,j_{n_x},k}+V_{j_1-\delta_{i,1},\ldots,j_{n_x}-\delta_{i,n_x},k}\right)\delta_{i,j}\,/\Delta X_i^2\right. \\
& +0.25\left(V_{j_1+\delta_{i,1}+\delta_{j,1},\ldots,j_{n_x}+\delta_{i,n_x}+\delta_{j,n_x},k}\right. \\
& -V_{j_1-\delta_{i,1}+\delta_{j,1},\ldots,j_{n_x}-\delta_{i,n_x}1+\delta_{j,n_x},k}-V_{j_1+\delta_{i,1}-\delta_{j,1},\ldots,j_{n_x}+\delta_{i,n_x}-\delta_{j,n_x},k} \\
& \left.+V_{j_1-\delta_{i,1}-\delta_{j,1},\ldots,j_{n_x}-\delta_{i,n_x}-\delta_{j,n_x},k}\right) \\
& \left.\cdot(1-\delta_{i,j})\,/(\Delta X_i\Delta X_j)\right]_{n_x\times n_x\times N_1\times\cdots\times N_{n_x}}\ ,
\end{aligned} \tag{6.11}$$

to second order accuracy using central finite differences.

Since the Hessian array is the largest array that needs to be calculated in the straight-forward finite difference method, ignoring the extra integration required for the jump integral, the order of the computational complexity for stochastic dynamic programming finite differences will be the same as the order of the number of elements in the Hessian array, i.e.,

$$\mathrm{O}\left(N_{\mathrm{DDV}}\right)=\mathrm{O}\left(n_x^2\cdot\prod_{i=1}^{n_x}N_i\right)=\mathrm{O}\left(n_x^2\cdot N_g^{n_x}\right)=\mathrm{O}\left(n_x^2\cdot e^{n_x\ln(N_g)}\right)\ , \tag{6.12}$$

where $N_g\equiv\left(\prod_{i=1}^{n_x}N_i\right)^{(1/n_x)}$ is the geometric mean of the spatial node numbers over all n_x dimensions. Hence, the growth of computational complexity is exponential with exponent $n_x\ln(N_g)$, proportional to the logarithm of the geometric mean of the spatial node size. The growth of the curse of dimensionality in the logarithm to the base 2 scale is illustrated in Fig. 6.1 for the Hessian size in bytes, i.e., $\log_2(8\cdot\mathrm{B}\cdot N_{\mathrm{DDV}})$. Note the top scale in the figure is about 60 log(B) and $2^{60}\mathrm{B}=1024^6\mathrm{B}$ is one terabyte (1TB) or 1024^2GB (1GB $=2^{40}$B is one gigabyte, while 1MB $=2^{20}$B is one megabyte) and that is well within the capabilities of our current largest scale computers.

For parallel processing techniques in computational stochastic dynamic programming refer to Hanson's 1996 chapter [23] and more recent paper [24] for more general supercomputing techniques and methods for solving jump-diffusion problems. For other methods, see the Markov chain approximation method, for instance, that is described in Kushner and Dupuis' book [44].

7. Jump-Diffusion Applications. In order to avoid the curse of dimensionality, sometimes, depending on the application, a canonical problem can be found that does not have the computational demands in the state as with stochastic dynamic programming. This means using an algorithm rather than high performance computing hardware to avoid or alleviate large computational demands due to dimensionality.

However, some problems do not fit the canonical dimensional reduction models and a fuller stochastic model and control problem is necessary. For instance, biological dynamical models usually have nonlinear terms due to saturation or other interaction effects. Although, many financial problems are linear, problems with stochastic volatility are usually nonlinear in an essential way. Two examples of these nonlinear problems are given in this section along with the two canonical examples.

7.1. LQJD Canonical Problem: Linear Jump-Diffusion-Dynamics and Quadratic Costs. The linear-quadratic jump-diffusion (LQJD) problem is a canonical optimal control problem with jump-diffusion (JD) dynamics that are linear (L) and costs that are quadratic (Q) in both state and control. Thus, starting from

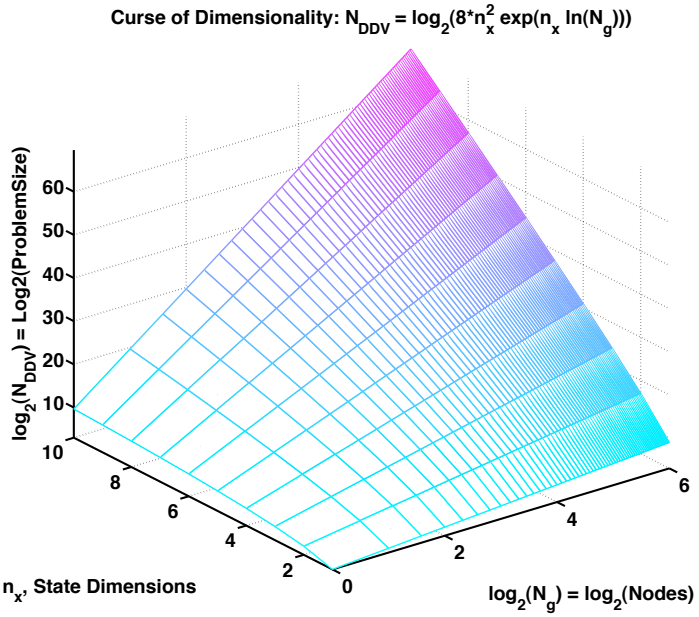

FIG. 6.1. *Estimate of the logarithm to the base 2 of the order of the growth of memory and computing demands using 8 byte words to illustrate the curse of dimensionality in the Hessian case for $n_x = 1{:}10$ dimensions and $N_g = 1{:}64 = 1{:}2^6$ geometric mean nodes per dimension. Note that 1KB or one kilobyte has a base 2 exponent of 10 = $log_2(2^{10})$, while the base 2 exponent is 20 for 1MB, 40 for 1GB and is 60 for 1TB.*

the marked-jump-diffusion control problem dynamics in (6.1), consider the following simplified coefficients,

$$\mathbf{f}(\mathbf{x},\mathbf{u},t) = \mathbf{f}_0(t) + f_1(t)\mathbf{x} + f_2(t)\mathbf{u}\,; \tag{7.1}$$

$$g(\mathbf{x},t) = g_0(t)\,; \tag{7.2}$$

$$h(\mathbf{x},t,\mathbf{q}) = h_0(t,\mathbf{q})\,, \tag{7.3}$$

where all time-dependent plant coefficients are commensurate in multiplication, i.e., $\mathbf{f}_0(t) = [f_{0,i}(t)]_{n_x\times 1}$, $f_1(t) = [f_{1,i,j}(t)]_{n_x\times n_x}$ and $f_2(t) = [f_{2,i,j}(t)]_{n_x\times n_u}$, while there are similar forms for $g_0(t)$ and $h_0(t,\mathbf{q})$. Note that the stochastic noise coefficients are in terms already linear in the noise.

Similarly, the quadratic cost coefficients of the optimal objective (6.2) are given as a general second degree polynomial in state and control variables,

$$\begin{aligned}\mathcal{C}(\mathbf{x},\mathbf{u},t) = \;& C_0(t) + \mathbf{C}_1^\top(t)\mathbf{x} + \mathbf{C}_2^\top(t)\mathbf{u} + \tfrac{1}{2}\mathbf{x}^\top C_{1,1}(t)\mathbf{x} + \mathbf{x}^\top C_{1,2}(t)\mathbf{u} \\ & + \tfrac{1}{2}\mathbf{u}^\top C_{2,2}(t)\mathbf{u}\,;\end{aligned} \tag{7.4}$$

$$S(\mathbf{x},t) = S_0(t) + \mathbf{S}_1^\top(t)\mathbf{x} + \frac{1}{2}\mathbf{x}^\top S_{1,1}(t)\mathbf{x}\,, \tag{7.5}$$

where the quadratic control running costs coefficient is positive-definite,

$$C_{2,2}(t) = [C_{2,2,i,j}]_{n_u\times n_u} > 0\,,$$

but can also be taken to be symmetric without loss of generality [4], since the pure quadratic form $(\mathbf{u}^\top A\mathbf{u})$ has a zero contribution from the antisymmetric part of the matrix (A). Similarly, the other quadratic running costs and final costs time-dependent coefficients will be taken as symmetric. Often $C_{1,1}(t)$ is taken to be positive-definite too, but can be relaxed in most applications.

The usual solution assumption is that the optimal expected value is quadratic in the state,

$$v(\mathbf{x},t) = v_0(t) + \mathbf{v}_1^\top(t)\mathbf{x} + \frac{1}{2}\mathbf{x}^\top v_{1,1}(t)\mathbf{x}\,, \tag{7.6}$$

for all $\mathbf{x} \in \mathcal{D}_x \subset \mathbb{R}^{n_x}$, like the final cost assumption (7.5) and can be justified heuristically or rigorously with much more effort [4]. Since $\mathbf{x}$ is an independent variable for the state, only the three time-dependent coefficients, $v_0(t)$, $\mathbf{v}_1$ and $v_{1,1}(t)$ need be found. Once found, the quadratic template (7.6) can be formed and represents a large reduction in computational demands to find the full state-time solution otherwise. The optimal state $v(\mathbf{x},t)$ satisfies the final condition $v(\mathbf{x},t_f) = S(\mathbf{x},t_f)$ for any $\mathbf{x} \in \mathcal{D}_x$ and by the arbitrariness of the state vector $\mathbf{x}$ and independence of the polynomial terms, the final conditions for the coefficients are

$$v_0(t_f) = S_0(t_f), \quad \mathbf{v}_1(t_f) = \mathbf{S}_1(t_f), \quad v_{1,1}(t_f) = S_{1,1}(t_f)\,. \tag{7.7}$$

By vector calculus, the optimal value solution has the following derivatives and jump-difference,

$$\begin{aligned}
v_t(\mathbf{x},t) &= v_0(t) + (\mathbf{v}_1')^\top(t)\mathbf{x} + \tfrac{1}{2}\mathbf{x}^\top v_{1,1}'(t)\mathbf{x}\,,\\
\nabla_x[v](\mathbf{x},t) &= \mathbf{v}_1(t) + v_{1,1}(t)\mathbf{x}\,,\\
\nabla_x\left[\nabla_x^\top[v]\right](\mathbf{x},t) &= v_{1,1}(t)\,,\\
v\left(\mathbf{x} + \widehat{\mathbf{h}}_{0,j}(t,q_j),t\right) - v(\mathbf{x},t) &= \mathbf{v}_1^\top(t)\widehat{\mathbf{h}}_{0,j}(t,q_j) + \widehat{\mathbf{h}}_{0,j}^\top(t,q_j)v_{1,1}(t)\mathbf{x}\\
&\quad + \tfrac{1}{2}\widehat{\mathbf{h}}_{0,j}^\top(t,q_j)v_{1,1}(t)\widehat{\mathbf{h}}_{0,j}(t,q_j),t)\,,
\end{aligned} \tag{7.8}$$

where $v_{1,1}(t)$ is taken to be symmetric since it appears in a pure quadratic form and where

$$\widehat{\mathbf{h}}_{0,j}^\top(t,q_j) \equiv [h_{0,i,j}(t,q_j)]_{n_x\times 1}$$

for $j = 1:n_p$.

Assembling all the parts, the Hamiltonian has the same form as the running cost,

$$\begin{aligned}
\mathcal{H}(\mathbf{x},\mathbf{u},t) = \mathcal{H}_0(t) + \boldsymbol{\mathcal{H}}_1^\top(t)\mathbf{x} + \boldsymbol{\mathcal{H}}_2^\top(t)\mathbf{u} + \tfrac{1}{2}\mathbf{x}^\top\mathcal{H}_{1,1}(t)\mathbf{x} + \mathbf{x}^\top\mathcal{H}_{1,2}(t)\mathbf{u}\\
+\tfrac{1}{2}\mathbf{u}^\top\mathcal{H}_{2,2}(t)\mathbf{u}\,.
\end{aligned} \tag{7.9}$$

In absence of control constraints, the regular or critical points with respect to the control vector are given by

$$\begin{aligned}
\nabla_u[\mathcal{H}](\mathbf{x},\mathbf{u},t) &= \boldsymbol{\mathcal{H}}_2(t) + \mathcal{H}_{1,2}^\top(t)\mathbf{x} + \mathcal{H}_{2,2}(t)\mathbf{u}\\
&= \mathbf{C}_2(t) + C_{1,2}(t)^\top\mathbf{x} + f_2^\top(t)\left(\mathbf{v}_1(t) + v_{1,1}(t)\mathbf{x}\right) + C_{2,2}(t)\mathbf{u}\\
&= \mathbf{0}_{n_u\times 1}\,,
\end{aligned} \tag{7.10}$$

where the inherited symmetry of $\mathcal{H}_{2,2}(t)$ has been used. Hence, the regular control vector is obtained in the form of linear feedback control,

$$\begin{aligned}\mathbf{u}^{(\mathrm{reg})}(\mathbf{x},t) &= -C_{2,2}^{-1}(t)\left(\mathbf{C}_2(t)+C_{1,2}(t)^{\top}\mathbf{x}+f_2^{\top}(t)\left(\mathbf{v}_1(t)+v_{1,1}(t)\mathbf{x}\right)\right),\\ &\equiv {}_0^{(\mathrm{reg})}(t)+u_1^{(\mathrm{reg})}(t)\mathbf{x}\,,\end{aligned} \tag{7.11}$$

using the invertibility of the positive definite $C_{2,2}(t)$. In the case of constrained control, e.g., hypercube constraints,

$$u_i^{(\min)}(\mathbf{x},t)\le u_i^{*}(\mathbf{x},t)\le u_i^{(\max)}(\mathbf{x},t)$$

for $i=1:n_u$ and $\mathbf{u}(\mathbf{x},t)\in\mathcal{D}_u$ given the $u_i^{(\min)}(\mathbf{x},t)$ and $u_i^{(\max)}(\mathbf{x},t)$, so the optimal control vector components have the continuous composite form,

$$u_i^{*}(\mathbf{x},t)=\left\{\begin{array}{ll} u_i^{(\min)}(\mathbf{x},t), & u_i^{(\mathrm{reg})}(\mathbf{x},t)\le u_i^{(\min)}(\mathbf{x},t)\\ u_i^{(\mathrm{reg})}(\mathbf{x},t), & u_i^{(\min)}(\mathbf{x},t)\le u_i^{(\mathrm{reg})}(\mathbf{x},t)\le u_i^{(\max)}(\mathbf{x},t)\\ u_i^{(\max)}(\mathbf{x},t), & u_i^{(\max)}(\mathbf{x},t)\le u_i^{(\mathrm{reg})}(\mathbf{x},t)\end{array}\right\}. \tag{7.12}$$

The time-dependent coefficients follow from substituting into the HJBE, $0=v_t(\mathbf{x},t)+\boldsymbol{\mathcal{H}}(\mathbf{x},\mathbf{u}^{*}(\mathbf{x},t),t)$, and collecting the different orders of the state vector $|\mathbf{x}|^k$ for $k=2:-1:0$, resulting in a *unidirectionally coupled system* of matrix differential equations subject to the final conditions (7.5). In the regular control region, terms solely order of $|\mathbf{x}|^2/2$ yield a matrix Riccati equation (MRE) of the form

$$0_{n_x\times n_x}=v_{1,1}'(t)-v_{1,1}^{\top}(t)C_{2,2}^{-1}(t)v_{1,1}(t)+F_{2,1}(t)v_{1,1}(t)+F_{2,0}(t) \tag{7.13}$$

for residual functions $F_{2,1}(t)$ and $F_{2,0}(t)$ that can be determined after some algebra [25, Chapter 6]. Given the solution for $v_{1,1}(t)$, terms solely order of $\mathbf{x}$ lead to a linear vector ODE for $\mathbf{v}_1(t)$,

$$0_{n_x\times 1}=\mathbf{v}_1'(t)+F_{1,1}(v_{1,1}(t),t)\mathbf{v}_1(t)+\mathbf{F}_{1,0}(v_{1,1}(t),t)\,, \tag{7.14}$$

for residual functions $F_{1,1}$ and $\mathbf{F}_{1,0}$, which can be found after some algebraic effort. Finally, given the solution coefficients $v_{1,1}(t)$ and $\mathbf{v}_1(t)$, terms order of one lead to a pure integration problem,

$$0=v_0'(t)+F_0(\mathbf{v}_1(t),v_{1,1}(t),t)\,, \tag{7.15}$$

for a residual function $F_0(\mathbf{v}_1(t),v_{1,1}(t),t)$. See Hanson [25, Chapter 6] for a complete description of the residual functions for this set of three matrix ODEs arising from a LQJD optimal control problem. The set of equations, (7.13), (7.14) and (7.15), are all backward equations and are solved in the top-down order presented. Since the computational problem is for an ODE system problem in time alone, the curse of dimensionality (6.12) formula has no meaning for the computation needed here.

See Westman and Hanson [64] for an application of the LQJD system to a multistage manufacturing system and Hanson [25, Chapter 6] for a more general development of the LQJD problem and variants. The LQJD problem has also been called the linear quadratic Gaussian-Poisson (LQJP), the jump linear quadratic Gaussian (JLQG) problem and other names. The classical reference of Anderson and Moore [4] has extensive treatments of the deterministic LQ problem, but also treats the filtering of Gaussian noise for the linear quadratic Gaussian (LQG) problem.

7.2. Optimal Portfolio and Consumption Canonical Problem with Constant Relative Risk Aversion (CRRA) Utility. The optimal portfolio and consumption problem was also pioneered by Merton [45, 46] for the scalar, pure diffusion model (3.12), even before the Black-Scholes-Merton 1973 options paper [15, 47] and before Merton's pioneering 1976 jump-diffusion options paper [48]. For the jump-diffusion optimal portfolio problem here, there is a financial portfolio which consists of a risky asset usually called a stock at price $S(t)$ that is assumed to satisfy the scalar linear version of the marked-jump-diffusion Eq. (5.4),

$$dS(t) = S(t)\left(\mu(t)dt + \sigma(t)dW(t) + \int_{\mathcal{Q}} h_0(q)\mathcal{P}(\mathbf{dt}, \mathbf{dq})\right), \tag{7.16}$$

where $W(t)$ is the Wiener process and a riskless (no noise) asset usually designated as a zero-coupon bond at price $B(t)$ satisfying the deterministic exponential growth equation,

$$dB(t) = r(t)B(t)dt\,, \tag{7.17}$$

where $r(t) > 0$ is the riskless interest rate. Let $U_0(t)$ be the instantaneous portfolio fraction of the bond and let $U(t)$ be the instantaneous fraction of the stock in the portfolio, such that $U_0(t) + U(t) = 1$. For flexibility in trading, the fraction $U(t)$ will not necessarily be in $[0, 1]$, since short selling of the stock could cause U to become negative with the deposited proceeds in bonds increasing U_0 to become greater than one or borrowing from bonds to buy stocks could cause U_0 to become negative and U exceed one, if only instantaneously.

The wealth $Z(t)$ of the portfolio with consumption follows from equating the relative change in the wealth to the relative price changes in the bond plus the stock less the consumption rate $C(t)$ relative to the current wealth, so

$$\begin{aligned} dZ(t) = Z(t)\Bigg(r(t)dt + U(t)\Bigg((\mu(t) - r(t))dt \\ +\sigma(t)dW(t) + \int_{\mathcal{Q}} ho(q)\mathcal{P}(\mathbf{dt}, \mathbf{dq})\Bigg)\Bigg) - C(t)dt\,, \end{aligned} \tag{7.18}$$

where $U_0(t) = 1 - U(t)$ has been substituted to eliminate the bond fraction. The wealth $Z(t)$ is the state variable and it is assumed to be non-negative, $Z(t) \geq 0$, to exclude bankruptcy [60, 59]. Similarly, the consumption will be bounded by wealth, i.e., $0 \leq C(t) \leq C^{(\max)}Z(t)$ where $C^{(\max)} \leq 1$. The consumption $C(t)$ and the stock fraction $U(t)$ are the control variables of the optimal portfolio problem. Also, it will be assumed that the stock fraction $U(t)$ will be bounded, $U^{(\min)} \leq U(t) \leq U^{(\max)}$, to exclude extreme borrowing and short-selling as in the papers of Hanson et al. [28, 68]. It turns out that, in the jump-diffusion optimal portfolio problem, there is an additional natural boundary condition resulting in restrictions on the stock fraction control.

The portfolio objective of the investor is to maximize the conditional, expected current value of the discounted utility $\mathcal{U}(Z(t_f))$ of final wealth at the end of the investment horizon at t_f and the discounted utility of the running consumption preferences

$\mathcal{U}(c)$, using the same utility function, i.e., the optimal value of the portfolio satisfies

$$v(z,t) = \max_{\{u,c\}(t,t_f]} \Big[\mathrm{E}\Big[e^{-\overline{\beta}(t,t_f)} \mathcal{U}(Z(t_f)) + \int_t^{t_f} e^{-\overline{\beta}(t,s)} \mathcal{U}(C(s))\, ds \;\Big|\; Z(t)=z, U(t)=u, C(t)=c \Big] \Big], \tag{7.19}$$

conditioned on the state-control set and $\overline{\beta}(t,s)$ is the cumulative time-discount over time in (t,s) with $\overline{\beta}(t,t)=0$ and discount rate $\beta(t) \equiv \overline{\beta}_s(t,t)$ at time t. The utility function $\mathcal{U}(c)$ is continuous, increasing ($\mathcal{U}'(c) > 0$) and convex ($\mathcal{U}''(c) < 0$, simply stated) for $c > 0$. The optimal value must satisfy the final time conditions $v(z,t_f) = \mathcal{U}(z)$ for all $z > 0$. In order to avoid Merton's [46] difficulties with power utility functions, $\mathcal{U}'(c) \to +\infty$ as $c \to 0^+$ will be assumed for the utility of consumption. The optimal value function must satisfy an absorbing boundary condition at zero wealth [60, 59, 49, 68], so

$$v(0^+,t) = \mathcal{U}(0) e^{-\overline{\beta}(t,t_f)} + \mathcal{U}(0) \int_t^{t_f} e^{-\overline{\beta}(t,s)} ds\,, \tag{7.20}$$

since consumption must vanish with wealth and $\mathcal{U}$ is continuous.

Application of the Itô stochastic chain rule to the principal of optimality corrected for t-dependent discounting factors [25, Chapter 10] (note that the usual running cost function has the form $\mathcal{C}(\mathbf{X}(s),\mathbf{U}(s),s)$ in (6.2), but (7.19) has an extra t-dependence of the form $\widehat{\mathcal{C}}(\mathbf{X}(s),\mathbf{U}(s),s,t)$, but the t-dependence can be separated out), the PIDE of stochastic dynamic programming becomes

$$\begin{aligned} 0 = {} & v_t(z,t) - \beta(t) v(z,t) + \mathcal{U}(c^*) + \left[(r(t) + (\mu(t) - r(t))u^*) z - c^* \right] v_z(z,t) \\ & + \tfrac{1}{2}\sigma^2(t)(u)^2 z^2 v_{zz}(z,t) + \int_{\mathcal{Q}} \left(v(\alpha(u^*,q)z, z) - v(z,t) \right) \phi_Q(q) dq\,, \end{aligned} \tag{7.21}$$

where $u^* = u^*(z,t) \in [U^{(\min)}, U^{(\max)}]$ and $c^* = c^*(z,t) \in [0, C^{(\max)} z]$ are the optimal controls if they exist, while $v_z(z,t)$ and $v_{zz}(z,t)$ are the partial derivatives with respect to wealth z when $0 \le t < t_f$. The wealth changes with a jump by a factor

$$\alpha(u,q) \equiv 1 + (e^q - 1)\, u\,,$$

after taking $h_0(q) = \exp(q) - 1$ as in (5.19) so that the log-change in the jump-amplitude is the mark q, in the post-jump wealth argument of (7.21).

However, since $\alpha(u,q)z$ is a post-jump wealth argument of the optimal value $v(z,t)$, then it must satisfy the non-negativity condition of the wealth, so $\alpha(u,q) \ge 0$ as well. This fact leads to the following control bounds as shown in the marked-jump-diffusion lemma of Zhu and Hanson [68]:

Lemma 7.1 (Bounds on Optimal Stock Fraction due to Non-Neqativity of Wealth Jump Argument). *If the support of $\phi_Q(q)$ is the finite interval $q \in [a(t), b(t)]$ with $a(t) < 0 < b(t)$, then $u^*(z,t)$ is restricted by (7.21) to*

$$\frac{-1}{\left(e^{b(t)} - 1\right)} \le u^*(z,t) \le \frac{1}{\left(1 - e^{a(t)}\right)}\,, \tag{7.22}$$

but if the support of $\phi_Q(q)$ is fully infinite, i.e., $q \in (-\infty, +\infty)$, then $u^*(z,t)$ is restricted by (7.21) to

$$0 \le u^*(z,t) \le 1\,. \tag{7.23}$$

Other cases, such as semi-infinite cases, follow from a combination of (7.22) and (7.23).

In absence of control constraints and optimization with respect to the consumption and stock fraction control variables in (7.21), respectively, the regular controls are given implicitly by

$$\mathcal{U}'\left(c^{(\mathrm{reg})}(z,t)\right) = v_z(z,t)\,, \tag{7.24}$$

and

$$\begin{aligned}\sigma^2(t)z^2v_{zz}(z,t)u^{(\mathrm{reg})}(z,t) = &-(\mu(t)-r(t))zv_z(z,t)\\ &-\lambda(t)z\int_{\mathcal{Q}}(e^q-1)v_z\left(\alpha\left(u^{(\mathrm{reg})}(z,t),q\right)z,t\right)\phi_Q(q)dq\,.\end{aligned} \tag{7.25}$$

Let the common utility of final wealth and running consumption be a power-type called constant relative risk aversion (CRRA) utility (Merton [45, 46, 49]),

$$\mathcal{U}(x) = x^\gamma/\gamma\,,\quad x\geq 0\,,\quad 0<\gamma<1\,, \tag{7.26}$$

where the relative risk aversion is the negative ratio of the second derivative to the average change of the first derivative,

$$\mathrm{RRA} = -\mathcal{U}''(x)/(\mathcal{U}'(x)/x) = 1-\gamma > 0$$

where $\mathcal{U}'(x)$ is called the *marginal utility*. Logarithmic utility ($\gamma = 0$, using the power utility limit of $\mathcal{U}(x) - 1/\gamma$ as $\gamma \to 0$ for $x > 0$) and extreme aversion power utilities ($\gamma < 0$) are considered elsewhere with a compilation by Sethi [59]. The risk-seeking or risk-loving power utilities, $\gamma > 1$, do not satisfy the prior stated convexity condition. The CRRA utilities are related to the more general *hyperbolic absolute risk aversion (HARA)* utilities.

The CRRA utilities (7.26) lead to a canonical dimensional computational reduction form,

$$v(z,t) = \mathcal{U}(z)v_0(t)\,, \tag{7.27}$$

where only the time-dependent function $v_0(t)$ of the template solution (7.27) need be found.

Consequently, the regular optimal consumption (7.24) simplifies to an explicit form linear in wealth,

$$c^{(\mathrm{reg})}(z,t) \equiv zc_0^{(\mathrm{reg})}(t) = z/v_0^{1/(1-\gamma)}(t)\,, \tag{7.28}$$

and the formula (7.25) for the regular stock fraction simplifies to a wealth-independent implicit form,

$$u^{(\mathrm{reg})}(z,t) = \ u_0^{(\mathrm{reg})}(t) \equiv \frac{1}{(1-\gamma)\sigma^2(t)}\left[\mu(t)-r(t)+\lambda(t)I_1\left(u_0^{(\mathrm{reg})}(t)\right)\right], \tag{7.29}$$

where

$$I_1(u) = \int_{\mathcal{Q}}(e^q-1)\alpha^{\gamma-1}(u,q)\phi_Q(q)dq\,. \tag{7.30}$$

The constrained optimal controls are given by

$$c^*(z,t)/z = c_0^*(t) \equiv \max\left[\min\left[c_0^{(\mathrm{reg})}(t), C_0^{(\max)}\right], 0\right],$$

provided $z > 0$, and

$$u^*(z,t) = u_0^*(t) \equiv \max\left[\min\left[u_0^{(\mathrm{reg})}(t), U_0^{(\max)}\right] U_0^{(\min)}\right]$$

is independent of z as with $u_0^{(\mathrm{reg})}(t)$. With these optimal controls, the PIDE of stochastic dynamic programming reduces to an implicitly defined Bernoulli equation in $v_0(t)$,

$$0 = v_0'(t) + (1-\gamma)\left(g_1(t, u_0^*(t))v_0(t) + g_2(t)v_0^{\frac{\gamma}{\gamma-1}}(t)\right), \tag{7.31}$$

where the control and time-dependent coefficients $g_1(t,u)$ and $g_2(t)$ are given in [68] and in [25, Chapter 10] for uniform mark distributions. No explicit solution can be found, but a reasonable amount of iteration can be used to compute solutions of the weakly implicit formulation as shown in the two cited references.

7.3. Coupled Inflationary Price and Nonlinear Renewable Resource Dynamics Control Problem. While many natural resource problems use constant prices, price inflation, depending on the magnitude, can have a significant effect on the optimal production as predicted by the law of supply and demand. Let $X_1(t)$ be the harvested mass of a renewable resource, such as that in an open access ocean or lake fishery, at time t. Assume that the resource, in absence of harvesting and noise, grows logistically with linear intrinsic growth rate r_1 and with carrying-capacity K_1. Let the resource be harvested at a rate $H(t)$ in mass per unit time. The resource is perturbed by Gaussian noise $W_1(t)$ and simple Poisson jump noise $P_1(t)$. Thus,

$$\begin{aligned} dX_1(t) = \; & (r_1 X_1(t)(1 - X_1(t)/K_1) - H(t))\, dt \\ & + \sigma_1 X_1(t)\, dW_1(t) + X_1(t) a_1\, dP_1(t)\,, \end{aligned} \tag{7.32}$$

where $X_1(0) = x_{1,0}$ with $\mathrm{E}[dW_1(t)] = 0$, $\mathrm{Var}[dW_1(t)] = dt$, $\mathrm{E}[dP_1(t)] = \lambda_1 dt$, $\mathrm{Var}[dP_1(t)] = \lambda_1 dt$ and $\mathrm{Cov}[dW_1(t),\ dP_1(t)] = 0$. Let the control $U_1(t)$ be the harvest rate per unit resource mass or harvesting effort corrected by a given efficiency coefficient q, .i.e, $H(t) = qU_1(t)X(t)$.

Let $p(t)$ be the price of harvested mass unit that satisfies the empirical nonlinear, inflationary harvesting supply-demand model relation of Hanson and Ryan [26],

$$p(t)H(t) = (p_0 + p_1 H(t))\, X_2(t)\,, \tag{7.33}$$

where $p(t) \cdot H(t)$ is the gross return rate on the harvest, p_0 is a constant gross return rate coefficient, p_1 is a linear price coefficient and $X_2(t)$ is a fluctuating inflationary factor satisfying the linear SDE

$$dX_2(t) = X_2(t)\left(r_2 dt + \sigma_2 dW_2(t) + a_2 dP_2(t)\right)\,, \tag{7.34}$$

where r_2 is the annual mean rate of inflation, a_2 is the jump-amplitude coefficient, σ_2 is the inflationary volatility and λ_2 is the inflationary jump rate. The background Gaussian noise $W_2(t)$ and the Poisson noise $P_2(t)$ satisfy similar first and second moments as those of the resource mass $X_1(t)$ plus pairwise independence of the resource mass and the inflationary factor noise.

Consider a quadratic performance criterion, such that the maximal, expected current value of future returns is

$$v(x_1,x_2,t) = \max_{\{U_1\}}\left[\mathrm{E}\left[\int_t^{t_f} e^{-\beta_0(s-t)}((p_0+p_1qU_1(s)X_1(s))\,X_2(s)-c(U_1(s)))\,ds \;\middle|\; X_1(t)=x_1, X_2(t)=x_2, U_1(t)=u_1\right]\right], \tag{7.35}$$

where β_0 is the constant nominal discount rate (i.e., not adjusted for inflation, since inflation is being separately modeled) and

$$c(u_1) = c_1u_1 + \frac{1}{2}c_2u_1^2$$

is the quadratic cost function with $c_2 > 0$ to insure minimum costs and a maximum return. Then, the corresponding PIDE of stochastic dynamic programming with correction for the extra t dependence of the discount factor, as in (7.21), is

$$\begin{aligned} 0 = \; & v_t(x_1,x_2,t) - \beta_0 v(x_1,x_2,t) + r_1x_1(1-x_1/K_1)v_{x_1}(x_1,x_2,t) \\ & + \tfrac{\sigma_1^2x_1^2}{2}v_{x_1,x_1} + \lambda_1\left(v\left((1+a_1)x_1,x_2,t\right) - v(x_1,x_2,t)\right) \\ & + r_2x_2v_{x_2} + \tfrac{\sigma_2^2x_2^2}{2}v_{x_2,x_2} + \lambda_2\left(v(x_1,(1+a_2)x_2,t) - v(x_1,x_2,t)\right) \\ & + \max_{u_1}\left[\mathcal{S}(x_1,x_2,u_1,t)\right], \end{aligned} \tag{7.36}$$

where the control switching term from the net running return in (7.35) is

$$\mathcal{S}(x_1,x_2,u_1,t) = p_0x_2 + \left(p_1x_2 - v_x(x_1,x_2,t)\right)qu_1x_1 - c_1u_1 - \frac{1}{2}c_2u_1^2, \tag{7.37}$$

and the final condition is $v(x_1,x_2,t_f) = 0$ for all of the positive state space, in absence of any salvage value. In addition, there must be an absorbing natural boundary condition at extinction,

$$v(x_1,0,t) = -\frac{\left(c_1 + \frac{1}{2}c_2U_1^{(\min)}\right)U_1^{(\min)}}{\beta_0}\left(1 - e^{-\beta_0(t_f-t)}\right), \tag{7.38}$$

for $x_1 \geq 0$, in the case of hypercube control constraints, $U_1^{(\min)} \leq u_1 \leq U_1^{(\max)}$, and assuming that $x_2 = 0$ when $x_1 = 0$ from (7.34). From examining the critical points of (7.37) with respect to the control variable u_1, the regular optimal control, in absence of constraints, is found:

$$u_1^{(\mathrm{reg})}(x_1,x_2,t) = \frac{(p_1x_2 - v_x(x_1,x_2,t))qx_1 - c_1}{c_2}, \tag{7.39}$$

where recall $c_2 > 0$, and with hypercube constraints, the optimal constrained control is the composite function,

$$u_1^*(x_1,x_2,t) = \left\{ \begin{array}{ll} U_1^{(\max)}, & U_1^{(\max)} \leq u_1^{(\mathrm{reg})}(x_1,x_2,t) \\ u_1^{(\mathrm{reg})}(x_1,x_2,t), & U_1^{(\min)} \leq u_1^{(\mathrm{reg})}(x_1,x_2,t) \leq U_1^{(\max)} \\ U_1^{(\min)}, & u_1^{(\mathrm{reg})}(x_1,x_2,t) \leq U_1^{(\min)} \end{array} \right\}. \tag{7.40}$$

For computational PDE procedures to solve this applied optimal stochastic control problem and further references, see Hanson et al. [23, 26]. See also Kushner and Dupuis [44] for the treatment of stochastic boundary conditions and the numerical Markov chain approximation method for solving stochastic control problems, relying more on probability. Andersen and Sutinen [3] review the general field of *stochastic bioeconomics*, i.e., the optimal control of renewable resources.

7.4. European Option Pricing with Stochastic-Volatility, Jump-Diffusions (SVJD). Andersen, Benzoni and Lund [2], Bakshi et al. [8] and Bates [11] similarly demonstrate that the most reasonable model of stock prices would include both stochastic-volatility and jump-diffusion (SVJD). Stochastic volatility or just *volatility* in finance is the standard deviation σ of the stock or other financial instrument per year. It is also the square root of the annual variance rate σ^2 of the return. According to Hull [34], a *European option* is an option that can only be exercised at the terminal time $t_f = T$ at strike price K. A *call option* is an option to buy the option underlying asset, usually a stock, while the *put option* is the option to sell. The gross payoff for the call option is $\max[S(T) - K, 0]$ and the stock price is $S(T)$ at time T, while the payoff for the put option is $\max[K - S(T), 0]$, but a rational option holder will only exercise the option if the payoff is positive, else will walk away from the option. The net payoff is the gross payoff minus the price or premium paid for the option.

7.4.1. SVJD Call Option Model. Unlike the classical Black-Scholes-Merton option pricing model [15, 47] with only one source of stock noise, a diffusion, to hedge away, the jump-diffusion has infinite sources of noise just due to the unbounded jump count of the Poisson counting process, without considering the random marks. Merton, in his pioneering jump-diffusion for options paper [48], specified an approximate option hedge for jump-diffusion called *risk-neutral pricing* in which the underlying stock price of the option is transformed so that the conditional mean rate of return for the transformed stock price SDE is the risk-less rate r, i.e., under the risk-neutral (rn) conditional expectation,

$$\mathrm{E}^{(\mathrm{rn})}[dS(t) \,|\, S(t)] = rS(t)dt\,.$$

A simple transformation can be made by way of an ODE integrating factor that shifts the mean drift rate to r rather than eliminating this mean rate. See Hull [34] for an applied finance explanation or Bingham and Kiesel [14] for an abstract explanation concerning the existence of a risk-neutral measure and an equivalent martingale measure. Thus, using the effective equivalent of the risk-neutral drift $r - \lambda \mathrm{E}[\exp(q) - 1]$ for $\mu(t)$, λ being a constant jump rate, using $\exp(q) - 1$ for $h_0(q)$ and $\sqrt{V(t)}dW_s(t)$ for $\sigma(t)dW(t)$ in (7.16) yields

$$dS(t) = S(t)\left(\left(r - \lambda \mathrm{E}\left[e^{Q} - 1\right]\right)dt + \sqrt{V(t)}dW_s(t) + \int_{\mathcal{Q}} (e^q - 1)\,\mathcal{P}(\mathbf{dt}, \mathbf{dq})\right)\,. \quad (7.41)$$

The risky asset $S(t)$ SDE is coupled through the volatility $\sqrt{V(t)}$ using Heston's [31] stochastic volatility model formulated in terms of an SDE for the variance rate $V(t)$,

$$dV(t) = \gamma_v\,(\theta_v - V(t))\,dt + \sigma_v\sqrt{V(t)}dW_v(t)\,, \quad (7.42)$$

with a mean-reverting drift and square root noise term, where $W_v(t)$ is a Wiener process with correlation

$$\mathrm{Corr}[W_s(t), W_v(t)] = \rho(t)$$

and the $\{\gamma_v, \theta_v, \sigma_v\}$ are constants; γ_v is the mean-reversion scaling rate, θ_v is the mean-reversion value and σ_v is the *volatility of the volatility*. Eqs. (7.41) and (7.42) comprise the SVJD model. Note that the SVJD model (7.41-7.42) is highly nonlinear in $V(t)$.

Although Heston [31] proposed a stochastic volatility model for a pure diffusion model (without jumps), Bates [11] and Scott [58] proposed SVJD models for currency options and European options with stochastic interest rates, respectively. The stochastic volatility model (7.42) was originally proposed by Cox, Ingersoll and Ross [19] just for interest rates, so (7.42) is also called the *CIR model*.

In finding the present value of the price of a European call option at time t it is necessary to use the discounted value of the risk-neutral conditionally expected payoff since the payoff is a final value, therefore

$$C(S(t),V(t),t;K,T)=e^{-r(T-t)}\mathrm{E}^{(\mathrm{rn})}[\max[S(T)-K,0]\,|\,S(t),V(t)]\,, \quad (7.43)$$

where $\exp(-r(T-t))$ is the discount-factor at the risk-neutral rate. Following Heston's [31] analogy with the Black-Scholes [15] decomposition of the pure diffusion call option price into $S(t)$ and K terms, and also Scott [58] for the SVJD model, then

$$\begin{aligned} C(S(t),V(t),t;K,T) = {} & S(t)\Phi_1^{(\mathrm{rn})}(S(t),V(t),t;K,T) \\ & -Ke^{-r(T-t)}\Phi_2^{(\mathrm{rn})}(S(t),V(t),t;K,T)\,, \end{aligned} \quad (7.44)$$

where

$$\Phi_1^{(\mathrm{rn})}(s,v,t;K,T)=e^{-r(T-t)}\,\mathrm{E}^{(\mathrm{rn})}\left[S(T)\mathbf{1}_{\{S(T)\geq K\}}\,\middle|\,S(t)=s,V(t)=v\right]\Big/ s$$

and

$$\Phi_2^{(\mathrm{rn})}(s,v,t;K,T)=\mathrm{E}^{(\mathrm{rn})}\left[\mathbf{1}_{\{S(T)\geq K\}}\,\middle|\,S(t)=s,V(t)=v\right]$$

are conditional risk-neutral tail probability distributions in the variable K and $\mathbf{1}_{\{\mathcal{S}\}}$ is an indicator function for the set $\mathcal{S}$ (see Yan and Hanson [66], for instance). The $\Phi_2^{(\mathrm{rn})}$ is a straight-forward probability distribution, while $\Phi_1^{(\mathrm{rn})}$ is shifted by the relative size of the final asset price $S(T)$.

It is convenient for application purposes to pose and solve this option problem by a brief formulation of a backward Kolmogorov PDE (here PIDE or partial integro-differential equation for the marked-jump-diffusion process) using Dynkin's integral formula introduced in the next subsubsection.

7.4.2. Kolmogorov PIDEs and Dynkin's Formula for Marked Jump-Diffusions. *Dynkin's formula* for marked-jump-diffusons in one state dimension can be written in terms of a conditional expectation which can be expanded by Itô's formula (see Hanson [25, Chapter 7] for details, justifications and further references) as follows: Let $X(t)$ be a marked-jump-diffusion,

$$dX(t)=f(X(t),t)dt+g(X(t),t)dW(t)+\int_{\mathcal{Q}}h(X(t),t,q)\mathcal{P}(\mathbf{dt},\mathbf{dq})\,, \quad (7.45)$$

with continuous coefficients. Let $v(x,t)$ be twice continuously differentiable in x and once in t, then Dynkin's formula can be written

$$\begin{aligned} \widehat{u}(x_0,t_0) = {} & \mathrm{E}[v(X(t),t)\,|\,X(t_0)=x_0] \\ = {} & v(x_0,t_0)+\mathrm{E}\left[\int_{t_0}^{t}\left(\frac{\partial v}{\partial t}(X(s),s)+\mathcal{B}_x[v](X(s),s)\right)ds\,\middle|\,X(t_0)=x_0\right], \end{aligned} \quad (7.46)$$

suppressing the forward t dependence of $\widehat{u}(x_0,t_0)=u(x_0,t_0;t)$ to emphasize its backward nature. Also, the *backward operator or generator* for this marked-jump-diffusion is defined as

$$\begin{aligned}\mathcal{B}_{x_0}[v](x_0,t_0) \equiv\ & f(x_0,t_0)\frac{\partial v}{\partial x_0}(x_0,t_0)+\tfrac{1}{2}g^2(x_0,t_0)\frac{\partial^2 v}{\partial x_0^2}(x_0,t_0)\\ &+\widehat{\lambda}(x_0,t_0)\int_{\mathcal{Q}}\Delta_h[v](x_0,t_0,q)\phi_Q(q)dq\,,\end{aligned}\tag{7.47}$$

where the forward time t is suppressed in $\widehat{\lambda}(x_0,t_0)\equiv\lambda(x_0,t_0;t)$ and the jump-difference is

$$\Delta_h[v](x_0,t_0,q)\equiv v(x_0+h(x_0,t_0,q),t)-v(x_0,t_0)\,.\tag{7.48}$$

For the *backward Kolmogorov equation (BKE)*, let $v=v(x)$ be independent of t and $\widehat{u}(x_0,t_0)=\mathrm{E}[v(X(t))\,|\,X(t_0)=x_0]$. Then, upon taking limits of Dynkin's formula (7.46) results in the backward Kolmogorov equation,

$$0=\frac{\partial\widehat{u}}{\partial x_0}(x_0,t_0)+\mathcal{B}_{x_0}\left[\widehat{u}\right](x_0,t_0)\,,\tag{7.49}$$

with final condition $u(x_0,t^-)=v(x_0)$.

An important application is the marked-jump-diffusion *transition density*

$$\widehat{\phi}(x_0,t_0)\equiv\phi_{X(t)}(x,t;x_0,t_0)$$

with distribution $\Phi_{X(t)}(x,t;x_0,t_0)\equiv\mathrm{Prob}[X(t)<x\,|\,X(t_0)=x_0]$ and backward equation,

$$0=\frac{\partial\widehat{\phi}}{\partial x_0}(x_0,t_0)+\mathcal{B}_{x_0}\left[\widehat{\phi}\right](x_0,t_0)\,,\tag{7.50}$$

with Dirac delta function final condition $\widehat{\phi}(x_0,t^-)=\delta(x_0-x)$, by writing the expectation of Dynkin's formula as

$$\widehat{u}(x_0,t_0)=\int_{-\infty}^{\infty}v(\xi)\phi_{X(t)}(\xi,t;x_0,t_0)d\xi\,,$$

setting $v(\xi)=\delta(\xi-x)$ and using (7.49).

7.4.3. Tail Probability PIDEs. Taking advantage of the geometric property of $S(t)$ in the SDE (7.41), a logarithmic change of variables is made with $L(t)=\ln(S(t))$ and $\kappa=\ln(K)$, so

$$C(S(t),V(t),t;K,T)=\widehat{C}(L(t),V(t),t;\kappa,T)$$

as a process, but

$$\widehat{C}(\ell,v,t;\kappa,T)=C(\exp(\ell),v,t;\exp(\kappa),T)$$

for PIDE representation where the logarithm of the stock price ℓ and the squared volatility v are now independent variables rather than processes as functions of time as in the SDE representation.

Application of the two state dimension form of Dynkin's formula and resulting backward Eq. (7.49) leads to the backward PIDE with backward operator $\mathcal{B}_{\ell,v}$ with respect to the variable set $\{\ell, v\}$,

$$\begin{aligned} 0 = \ & \frac{\partial \widehat{C}}{\partial t} + \mathcal{B}_{\ell,v}\Big[\widehat{C}\Big](\ell, v, t; \kappa, T) \equiv \frac{\partial \widehat{C}}{\partial t} + \big(r - \lambda \mathrm{E}\left[e^q - 1\right] - \tfrac{1}{2}v\big)\frac{\partial \widehat{C}}{\partial \ell} \\ & + \alpha_v(\theta_v - v)\frac{\partial \widehat{C}}{\partial v} + \tfrac{1}{2}v\frac{\partial^2 \widehat{C}}{\partial \ell^2} + \rho\sigma_v v\frac{\partial^2 \widehat{C}}{\partial \ell \partial v} + \tfrac{1}{2}\sigma_v^2 v\frac{\partial^2 \widehat{C}}{\partial v^2} \\ & - r\widehat{C} + \lambda\int_{-\infty}^{\infty}\Big(\widehat{C}(\ell + q, v, t; \kappa, T) - \widehat{C}(\ell, v, t; \kappa, T)\Big)\phi_Q(q)dq\,, \end{aligned} \tag{7.51}$$

subject to the final condition that the final payoff $\widehat{C}(\ell, v, T; \kappa, T) = \max[\ell - \kappa, 0]$, which is assumed to be nonnegative for all variables concerned. Decomposing (7.51) into the corresponding tail probabilities in the current notation produces, for $\Phi_1^{(\mathrm{rn})} = \widehat{\Phi}_1^{(\mathrm{rn})}(\ell, v, t; \kappa, T)$,

$$0 = \frac{\partial \widehat{\Phi}_1^{(\mathrm{rn})}}{\partial t} + \mathcal{B}_{\ell,v}^{(1)}\Big[\widehat{\Phi}_1^{(\mathrm{rn})}\Big](\ell, v, t; \kappa, T) \equiv \frac{\partial \widehat{\Phi}_1^{(\mathrm{rn})}}{\partial t} + \mathcal{B}_{\ell,v}\Big[\widehat{\Phi}_1^{(\mathrm{rn})}\Big] + v\frac{\partial \widehat{\Phi}_1^{(\mathrm{rn})}}{\partial \ell} + \rho\sigma_v v\frac{\partial \widehat{\Phi}_1^{(\mathrm{rn})}}{\partial v} \tag{7.52}$$
$$+ (r - \lambda \mathrm{E}\left[e^q - 1\right])\,\widehat{\Phi}_1^{(\mathrm{rn})} + \lambda\int_{-\infty}^{\infty}(e^q - 1)\,\widehat{\Phi}_1^{(\mathrm{rn})}(\ell + q, v, t; \kappa, T)\phi_Q(q)dq,$$

with indicator function final condition $\widehat{\Phi}_1^{(\mathrm{rn})}(\ell, v, T; \kappa, T) = \mathbf{1}_{\{\ell > \kappa\}}$, i.e., in-the-money (ITM) or $S(t) > K$ for the call, and for the probability $\Phi_2^{(\mathrm{rn})} = \widehat{\Phi}_2^{(\mathrm{rn})}(\ell, v, t; \kappa, T)$,

$$0 = \frac{\partial \widehat{\Phi}_2^{(\mathrm{rn})}}{\partial t} + \mathcal{B}_{\ell,v}^{(2)}\Big[\widehat{\Phi}_2^{(\mathrm{rn})}\Big](\ell, v, t; \kappa, T) \equiv \frac{\partial \widehat{\Phi}_2^{(\mathrm{rn})}}{\partial t} + \mathcal{B}_{\ell,v}\Big[\widehat{\Phi}_2^{(\mathrm{rn})}\Big] + r\widehat{\Phi}_2^{(\mathrm{rn})}\,, \tag{7.53}$$

with the same final condition $\widehat{\Phi}_2^{(\mathrm{rn})}(\ell, v, T; \kappa, T) = \mathbf{1}_{\{\ell > \kappa\}}$.

7.4.4. Fourier Transform Approach to Solving PIDEs. The PIDEs, (7.52) and (7.53), can be transformed to more useful forms by Fourier transforms, but in probability and finance they are called characteristic functions. Here they are relative to the density of the logarithms of the strike price, $\phi_j^{(\mathrm{rn})}(\kappa; s, v, t, T) = \partial \widehat{\Phi}_j^{(\mathrm{rn})}(\ell, v, t; \kappa, T)/\partial \kappa$,

$$f_j(\ell, v, t; y, T) \equiv -\int_{-\infty}^{\infty} e^{iy\kappa}\phi_j^{(\mathrm{rn})}(\kappa; s, v, t, T)d\kappa\,, \tag{7.54}$$

for $j = 1 : 2$, the minus sign accounting for the fact that the distributions are complementary distributions. Due to the form of (7.54), these characteristic functions satisfy the same PIDEs as the $\widehat{\Phi}_j^{(\mathrm{rn})}(\ell, v, t; \kappa, T)$,

$$\frac{\partial f_j}{\partial t}(\ell, v, t; \kappa, T) + \mathcal{B}_{\ell,v}^{(j)}[f_j](\ell, v, t; \kappa, T) = 0\,, \tag{7.55}$$

but satisfy the transformed final conditions $f_j(\ell, v, T; y, T) = +\exp(iy\ell)$, respectively for $j = 1 : 2$. A solution form that works was guessed by Heston [31], which in the marked-jump-diffusion case is an exponential with an affine exponent in terms of the time-to-go $\tau \equiv T - t$,

$$f_j(\ell, v, t; y, t + \tau) = \exp(g_j(\tau) + h_j(\tau)v + \beta_j(\tau) + iy\ell)\,, \tag{7.56}$$

where $g_j(0) = 0 = h_j(0)$ and $\beta_j(\tau) = r\tau\delta_{j,2}$ for $j = 1:2$. With much effort, the coefficient functions can be found; see Yan and Hanson [66] for the details. Upon transforming the complex contour to an equivalent contour on the real line, the inverse transform for the original risk-neutral tail probabilities is

$$\Phi_j^{(\mathrm{rn})}(s,v,t;K,T) = \frac{1}{2} + \frac{1}{\pi}\int_{0^+}^{+\infty} \mathrm{Re}\left[\frac{e^{-iy\ln(K)} f_j(\ln(s),v,t;y,T)}{iy}\right] dy\,, \quad (7.57)$$

for $j = 1:2$, yielding a residue of $1/2$ and a principal value integral from combining the real segment contributions such that the integrand is bounded in the limit to the apparent singularity as $y \to 0^+$.

Once the European call option prices are calculated, the *put-call parity*,

$$P(S(t),V(t),t;K,T) = C(S(t),V(t),t;K,T) + Ke^{-r(T-t)} - S(t)\,, \quad (7.58)$$

can be used to easily compute the European put option price $P(S(t),V(t),t;K,T)$. Hull [34, Chapter 7] and Higham [33] explain how the European put-call parity follows from the maximum function property and the common value, $\max[S(T),K]$, of both options.

Again, see Yan [65] and Yan and Hanson [66] for the details of these numerical procedures. For further background see Heston [31], Bates [11], Scott [58], Carr and Madan [16] and Hull and White [35]. American options are very different from European options, since American options can be exercised any time in the period $[0,T]$ leading to an interesting but complicated free boundary problem. Yan [65] and Hanson and Yan [29] studied American options in an SVJD environment by two different computational methods and further references can be found there.

8. Conclusions. It is possible to understand the basic mathematics of jump-diffusion processes and control based upon the first principles of applied mathematics by supplementing the continuity concepts of classic analysis or calculus with the properties of jump-discontinuities and nonsmoothness of Markov processes. The result is an applied stochastic calculus. This works for many jump-diffusion applications in finance, biology, manufacturing and other stochastic research areas. Otherwise, some scientists and financial analysts would find the areas inaccessible due to the many layers of theorems and results in the more abstract approaches.

However, first understanding the applied side of these areas should help to develop the motivation and intuition to further understand the theorems and results on the abstract side. For instance, the complete market theorem of Harrison and Pliska [30] for continuous processes, i.e., diffusions, relies heavily on martingale theory, so would be very difficult to justify by the applied approach used here, but the result does not hold for jump-diffusion or SVJD. Hence, there are still many applications of jump-diffusions and SVJD in the literature and there are a good number of open problems in this currently very active area. Only a few have been mentioned in this compact survey.

Acknowledgements. This survey grew out of a compact course *Stochastic Processes and Control of Jump Diffusions, with Applications* that the author gave at the Department of Mathematics at the Indian Institute of Science (IISc) in Bangalore, India for the month of February 2007 as part of the *Stochastic Processes and Applications* 2006-2007 theme of the IISc Mathematics Initiative. The author is grateful to Mrinal K. Ghosh and the Department of Mathematics for the generous invitation, as well as local support and joint travel support with the University of Illinois.

REFERENCES

[1] Y. Aït-Sahalia, *Disentangling Diffusion from Jumps*, J. Fin. Economics, 74 (2004), pp. 487-528.

[2] T. G. Andersen, L. Benzoni and J. Lund, *An Empirical Investigation of Continuous-Time Equity Return Models*, J. Finance, 57 (2002), pp. 1239-1284.

[3] P. Andersen and J. G. Sutinen, *Stochastic Bioeconomics: A Review of Basic Methods and Results*, Marine Resource Economics, 1 (1982), pp. 1-10.

[4] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[5] D. Applebaum, *Lévy Processes and Stochastic Calculus*, Cambridge University Press, Cambridge, UK, 2003.

[6] C. A. Aourir, D. Okuyama, C. Lott and C. Eglinton, *Exchanges - Circuit Breakers, Curbs, and Other Trading Restrictions*, 2002, `URL:http://invest-faq.com/articles/exch-circuit-brkr.html` .

[7] L. Arnold, *Stochastic Equations: Theory and Applications*, John Wiley, New York, NY, 1974.

[8] G. Bakshi, C. Cao and Z. Chen, *Empirical Performance of Alternative Option Pricing Models,* J. Finance, 52 (1997), pp. 2003-2049.

[9] C. A. Ball and W. N. Torous, *On Jumps in Common Stock Prices and Their Impact on Call Option Prices*, J. Finance, 40 (1985), pp. 155-173.

[10] O.E. Barndorff-Nielsen and N. Shephard, *Econometrics of Testing for Jumps in Financial Economics Using Bipower Variation*, J. Fin. Econometrics, 9 (2006), pp. 1-30.

[11] D. Bates, *Jump and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options*, Rev. Fin. Studies, 9 (1996), pp. 69-107.

[12] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[13] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ, 1961.

[14] N. H. Bingham and R. Kiesel, *Risk-Neutral Valuation: Pricing and Hedging of Financial Derivatives*, Springer-Verlag, New York, NY, 2004.

[15] F. Black and M. Scholes, *The Pricing of Options and Corporate Liabilities*, J. Political Economy, 81 (Spring 1973), pp. 637-659.

[16] P. Carr and D. B. Madan, *Option Valuation Using the Fast Fourier Transform*, J. Comp. Finance, 2 (1999), pp. 61-73.

[17] E. Çinlar, *Introduction to Stochastic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1975.

[18] R. Cont and P. Tankov, *Financial Modelling with Jump Processes*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[19] J. C. Cox, J. E. Ingersoll and S. A. Ross, *A Theory of the Term Structure of Interest Rates*, Econometrica, 53 (1985), pp. 385-408.

[20] I. I. Gihman and A. V. Skorohod, *Stochastic Differential Equations*, Springer-Verlag, New York, NY, 1972.

[21] I. I. Gihman and A. V. Skorohod, *Controlled Stochastic Processes*, Springer-Verlag, New York, NY, 1979.

[22] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer-NY, New York, NY, 2003.

[23] F. B. Hanson, *Computational Stochastic Dynamic Programming*, in Stochastic Digital Control System Techniques, C. T. Leondes, ed., Control and Dynamic Systems: Advances in Theory and Applications, Vol. 76, Academic Press, New York, NY, 1996, pp. 103-162.

[24] F. B. Hanson, *Local Supercomputing Training in the Computational Sciences Using Remote National Centers*, Future Generation Computer Systems: Special Issue on Education in the Computational Sciences, 19 (2003), pp. 1335-1347.

[25] F. B. Hanson, *Applied Stochastic Processes and Control for Jump-Diffusions: Modeling, Analysis and Computation*, SIAM Books, Philadelphia, PA, to be released September 2007.

[26] F. B. Hanson and D. Ryan, *Optimal Harvesting with Both Population and Price Dynamics*, Math. Biosciences, 148 (1998), pp. 129-146.

[27] F. B. Hanson and J. J. Westman, *Jump-Diffusion Stock Return Models in Finance: Stochastic Process Density with Uniform-Jump Amplitude*, Proc. 15th Int. Sympos. Math. Theory of Networks and Systems, (2002), pp. 1-7.

[28] F. B. Hanson and J. J. Westman, *Portfolio Optimization with Jump–Diffusions: Estimation of Time-Dependent Parameters and Application*, Proc. 41st Conf. on Decision and Control, (2002), pp. 377-382.

[29] F. B. Hanson and G. Yan, *American Put Option Pricing for Stochastic-Volatility, Jump-Diffusion Models*, in Proc. Amer. Control Conf., (2007), 6 pages, invited paper to appear

July 11, 2007.
[30] J. M. Harrison and S. R. Pliska, *Martingales and Stochastic Integrals in the Theory of Continuous Trading*, Stochastic Processes & Appl., 11 (1981), pp. 215-260.
[31] S. L. Heston, *A Closed-form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, Rev. Fin. Studies, 6 (1993), pp. 327-343.
[32] D. J. Higham, *An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations*, SIAM Review, 43 (2001), pp. 525-546.
[33] D. J. Higham, *An Introduction to Financial Option Valuation: Mathematics, Stochastics and Computation*, Cambridge University Press, Cambridge, UK, 2004.
[34] J. C. Hull, *Options, Futures, & Other Derivatives*, 4th Edition, Prentice-Hall, Englewood Cliffs, NJ, 2000.
[35] J. Hull and A. White, *The Pricing of Options with Stochastic Volatilities*, J. Finance, 42 (1987), pp. 281-300.
[36] K. Itô, *On Stochastic Differential Equations*, Mem. Amer. Math. Soc., no. 4 (1951), pp. 1-51.
[37] J. Jacod and A. Shiryaev, *Limit Theorems for Stochastic Processes*, Springer-Verlag, New York, NY, 1987.
[38] R. A. Jarrow and E. R. Rosenfeld, *Jump Risks and the Intertemporal Capital Asset Pricing Model*, J. Business, 57 (1984), pp. 337-351.
[39] P. Jorion, *On Jump Processes in the Foreign Exchange and Stock Markets*, Rev. Fin. Studies, 88 (1989), pp. 427-445.
[40] S. G. Kou, *A Jump Diffusion Model for Option Pricing*, Management Science, 48 (2002, pp. 1086-1101.
[41] S. G. Kou and H. Wang, *Option Pricing Under a Double Exponential Jump Diffusion Model*, Management Science, 50 (2004), pp. 1178-1192.
[42] H. J. Kushner, *Stochastic Stability and Control*, Academic Press, New York, NY, 1967.
[43] H. J. Kushner, *Jump-Diffusions with Controlled Jumps: Existence and Numerical Methods*, J. Math. Anal. Applic., 249 (2000), pp. 179-198.
[44] H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, 2nd Edition, Springer-Verlag, New York, NY, 2001.
[45] R. C. Merton, *Lifetime Portfolio Selection Under Uncertainty: The Continuous-Time Case*, Rev. Economics and Statistics, 51 (1969), pp. 247-257.
[46] R. C. Merton, *Optimum Consumption and Portfolio Rules in a Continuous-Time Model*, J. Econ. Theory, 3 (1971), pp. 373-413. (Reprinted in Merton [49, Chapter 5].)
[47] R. C. Merton, *Theory of Rational Option Pricing*, Bell J. Econ. Mgmt. Science, 4 (Spring 1973), pp.141-183. (Reprinted in Merton [49, Chapter 8].)
[48] R. C. Merton, *Option Pricing When Underlying Stock Returns are Discontinuous*, J. Financial Economics, 3 (1976), pp. 125-144. (Reprinted in Merton [49, Chapter 9].)
[49] R. C. Merton, *Continuous Time Finance*, Blackwell Publishers, Cambridge, MA, 1992.
[50] T. Mikosch, *Elementary Stochastic Calculus: with Finance in View*, World Scientific, Singapore, 1998.
[51] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, Fifth ed., Springer-Verlag, New York, NY, 1998.
[52] B. Øksendal and A. Sulem, *Applied Stochastic Control of Jump Diffusions*, Springer-Verlag, Berlin, 2005.
[53] P. Protter, *Stochastic Integration and Differential Equations: A New Approach*, Springer-Verlag, Berlin, 2004.
[54] C. A. Ramzani and Y. Zeng, *Maximum Likelihood Estimation of Asymmetric Jump-Diffusion Processes: Application to Security Prices*, Social Science Research Network, `http://ssrn.com/abstract=606361`, 1998, pp. 1-32.
[55] W. J. Runggaldier, *Jump-Diffusion Models*, in Handbook of Heavy Tailed Distributions in Finance, S. T. Rachev, ed., Handbooks in Finance, Elsevier/North-Holland, New York, NY, 2003, pp. 169-209.
[56] J. Rust, *Do People Behave According to Bellman's Principle of Optimality?*, Hoover Institution Working Paper E-92-10, 1994, 65 pages.
[57] Z. Schuss, *Theory and Applications of Stochastic Differential Equations*, John Wiley, New York, NY, 1980.
[58] L. O. Scott, *Pricing Stock Options in a Jump-Diffusion Model with Stochastic Volatility and Interest Rates: Applications of Fourier Inversion Methods*, Math. Finance, 7 (1997), pp. 413-424.
[59] S. P. Sethi, *Optimal Consumption and Investment with Bankruptcy*, Kluwer Academic Publishers, Boston, MA, 1997.
[60] S. P. Sethi and M. Taksar, *A Note on Merton's "Optimum Consumption and Portfolio Rules*

in a Continuous-Time Model", J. Econ. Theory, 46 (1988), pp. 395-401.

[61] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*, Second ed., Springer-Verlag, New York, NY, 1991.

[62] J. M. Steele, *Stochastic Calculus and Financial Applications*, Springer, New York, NY, 2001.

[63] J. J. Westman and F. B. Hanson, *Nonlinear State Dynamics: Computational Methods and Manufacturing Example*, Int. J. Control, 73 (2000), pp. 464-480.

[64] J. J. Westman and F. B. Hanson, *The LQGP Problem: A Manufacturing Application*, in Proc. Amer. Control Conf., (1997), pp. 566-570.

[65] G. Yan, *Option Pricing for a Stochastic-Volatility Jump-Diffusion Model*, Ph.D. Thesis in Mathematics, Univ. Illinois at Chicago, Chicago, IL, 23 June 2006, 128 pages.

[66] G. Yan and F. B. Hanson, *Option Pricing for a Stochastic-Volatility Jump-Diffusion Model with Log-Uniform Jump-Amplitudes*, in Proc. Amer. Control Conf., (2006), pp. 2989-2994.

[67] Z. Zhu, *Option Pricing and Jump-Diffusion Models*, Ph.D. Thesis in Mathematics, Univ. Illinois at Chicago, Chicago, IL, 17 October 2005, 165 pages.

[68] Z. Zhu and F. B. Hanson, *Optimal Portfolio Application with Double-Uniform Jump Model* in Control Theory Applications in Financial Engineering and Manufacturing, H. Yan, G. Yin, Q. Zhang eds., International Series in Operations Research and Management Sciences, Vol. 94 Springer/Kluwer, New York, NY, 2006, pp. 331-358, invited chapter.