– 327 (revised) –

Symbolic Itô calculus in AXIOM: an ongoing story

by

W.S. Kendall

January 1999

Symbolic Itô calculus in *AXIOM*: an ongoing story

Wilfrid S. Kendall, Statistics, University of Warwick
Supported by EPSRC research grant GR/K71677.

January 13, 1999

Abstract

Symbolic Itô calculus refers both to the implementation of Itô calculus in a computer algebra package and to its application. This article[1] reports on progress in the implementation of Itô calculus in the powerful and innovative computer algebra package *AXIOM*, in the context of a decade of previous implementations and applications. It is shown how the elegant algebraic structure underlying the expressive and effective formalism of Itô calculus can be implemented directly in *AXIOM* using the package's programmable facilities for "strong typing" of computational objects. An application is given of the use of the implementation to provide calculations for a new proof, based on stochastic differentials, of the Mardia-Dryden distribution from statistical shape theory.

Key words: *AXIOM*, computer algebra, coupling of random processes, financial mathematics, Itô formula, Itô calculus, *Itovsn3*, Mardia-Dryden density *Mathematica*, `Macsyma`, `Maple`, *REDUCE*, semimartingale, statistics of shape, stochastic calculus, stochastic integral, symbolic Itô calculus.

AMS 1991 SUBJECT CLASSIFICATION: **58G32**.

Introduction

The Itô calculus, or stochastic calculus for semimartingales, provides a flexible, powerful, and expressive formalism for the study of continuous random processes. Over the past ten years or so, methods have been developed for utilizing computer algebra in stochastic calculus, typically by implementing stochastic calculus within a suitable computer algebra package; *symbolic Itô calculus.* The purpose of this article is to report on the progress of a project to implement Itô calculus in the innovative *AXIOM* computer algebra package [15], which

[1]This is `/home/bach/wsk/ms/Montreal/montreal.tex` (`unix`) version: 1.16. It was last edited on 17:29:24, 12/01/1999.

uses elegant mathematical ideas to provide a powerful structured programming environment, eminently suitable for implementation of the rich formalism of stochastic calculus. This builds on previous implementations in *REDUCE* and *Mathematica*, sharing a common approach and hence collectively termed *Itovsn3*.

The combination of computer algebra and stochastic calculus makes the following tasks possible (all of which have been carried out using various versions of *Itovsn3*):

- searching out informative representations of stochastic processes arising in applications;
- automatic identification of the natural geometry of a diffusion;
- identification of local martingales, and searching out of supermartingales and submartingales – a technique which has been applied to solve difficult and nonstandard stochastic control problems;
- solution of classes of stochastic differential equations;
- combination of the powerful mathematical ideas of Itô calculus with algebraic, graphical and numerical capabilities of the computer algebra package;
- computer-aided preparation of simulation code and automatic simulation of solutions to stochastic differential equations.

In order to deliver a useful, reliable, and flexible implementation we need to take seriously the detailed algebraic issues underlying the Itô calculus mathematical formalism. The strongly-typed programming environment of *AXIOM* makes it feasible to do this in a way which builds efficiently on pre-existing *AXIOM* code. Accordingly, the article commences by describing (in Section 1) this formalism in general terms, eschewing technicalities; the features discussed there are all reflected more-or-less exactly in the detailed implementation of *AXIOM Itovsn3*.

Early implementations of stochastic calculus in various computer algebra packages took implicit, not explicit, account of this formalism; Section 2 surveys the development of these successive implementations. The logic of this development has led to the present implementation in *AXIOM*, of which Section 3 gives a flavour (full technical details are to be found in the associated research report [26]).

The use of computer algebra in statistics and probability can only be justified by successful applications: Section 4 describes a new application of the *AXIOM* version of symbolic Itô calculus to a distributional result in the statistics of shape. Finally Section 5 summarizes future plans and prospects for this implementation and related ideas.

1 An informal description of Itô calculus

Itô calculus is based on the notion of Brownian motion, also called the Wiener process. Recall that the concept of Brownian motion arose from *empirical* science, namely the observations of the botanist Robert Brown [6] in the last century, the calculations of Einstein at the turn of the century (described in book-form in [10]) showing that the phenomenon observed by Brown could be explained by the molecular theory of fluids, and independently the speculations of Bachelier [3] about fluctuations of stock-market prices. From these empirical origins it has grown into a huge and fertile theory. Here we simply cite two recent examples: the work of Watkins [50] and others on stochastic models for leucocyte cell movement, and the excellent elementary exposition of Brownian motion in financial mathematics by Baxter and Rennie [4]. In order to understand how to use computer algebra effectively in a stochastic calculus context, we must first review the beautiful theory of Itô calculus which has been built up from these empirical roots.

The Itô calculus provides a way to use the Brownian motion as a basic building block to construct a vast range of other processes, and as such is fundamental to a large number of modern applications of Brownian motion. The idea dates back to Itô's classic papers [12, 13], and a strong independent contribution from the Russian school. An intuitive appreciation of the idea can be gained by considering Brownian motion as a kind of infinitesimal random walk, as presented in the remainder of this section (though of course for a full appreciation one has to engage more fully with the underlying mathematics; some textbook references are given towards the end of this section).

One can view *Brownian motion* B as the limiting case of an unbiased random walk with jumps of $\pm\sqrt{\Delta t}$ occurring every Δt units of time, as Δt tends to zero. In particular

$$\mathbb{E}[\Delta B|\mathfrak{F}_t] \quad = \quad \mathbb{E}[B(t+\Delta t)|\mathfrak{F}_t] - B(t) \quad = \quad 0\,, \tag{1}$$

where the conditional expectation $\mathbb{E}[\cdot|\mathfrak{F}_t]$ refers to conditioning on the class $\mathfrak{F}_t$ of all events taking place in the past at time t. (We shall make no further mention of the underlying *filtration* $\{\mathfrak{F}_t : t \geq 0\}$, though it is an important and fundamental concept in the mathematical theory.)

Thus B may be viewed as the coordinate of a randomly fluctuating particle, or as an approximation to the time-varying price of a share. Ideas of trading in shares are very useful in conveying precisely the right intuition underlying stochastic calculus! We have constant "randomness" or *volatility* $\mathbb{E}[(\Delta B)^2|\mathfrak{F}_t] = \Delta t$, and the unbiased nature of the Brownian motion means that this share price has no "trend" or drift. Writing this as if it were an equation of differentials, we obtain

$$(\mathtt{d}B)^2 \quad = \quad \mathtt{d}t\,. \tag{2}$$

This curious equation corresponds directly to the very irregular behaviour of the Brownian path, which fails to have finite bounded variation but has bounded *quadratic variation*:

$$\sum_{\text{infinitesimal jumps in } [0,t]} (\Delta B)^2 \quad \to \quad t\,. \tag{3}$$

Stochastic integration with respect to B now corresponds exactly to the idea of trading on a share with price B. If one holds H units of the share, then after time Δt, and a change of share price from B to $B+\Delta B$, the holding has gained value $H\Delta B$. Naturally one imposes a "no-insider-trading" condition on the investment strategy H: at time t the strategy $H(t)$ may be random but must depend only on the strict past (observations of B and other prices only before time t). It is elementary to deduce from this that $\sum H\Delta B$ forms a discrete-time martingale. Under this condition (for suitably integrable investment strategies H) we can proceed to the limit and measure the change in fortune under this strategy as a *stochastic integral*:

$$\sum H\Delta B \quad \to \quad \int H \mathtt{d}B\,. \tag{4}$$

In the limit the "no-insider-trading" condition becomes a requirement that the investment strategy H is *predictable*, as well as satisfying some integrability constraints, and the resulting fortune $\int H\mathtt{d}B$ inherits a trend-free property from B: clearly failure of the predictability condition would mean that the resulting fortune might acquire some trend, but would also raise the serious issue of whether the limiting stochastic integral can exist at all!

More generally the share price M might remain a *martingale*, with vanishing "trend" or drift $\mathbb{E}[\Delta M|\mathfrak{F}_t] = 0$, but with its volatility ($\mathbb{E}[\Delta M|\mathfrak{F}_t] \approx (\Delta M)^2$) perhaps varying over time. This idea makes sense in the limit, and we may talk of a continuous random process M which has zero drift ($\mathtt{drift}(\mathtt{d}M) = 0$), but whose volatility ($(\mathtt{d}M)^2 = \mathtt{d}[M,M]$) is itself varying with time. However we must take care because, just as a random variable arising from summing infinitely many random variables of zero expectation need not itself have zero expectation, so integrating up a stochastic differential $\mathtt{d}M$ of zero drift need not lead to a process M which is a martingale: in general it will only be a *local martingale* (a process which is a martingale if stopped in a way which keeps it bounded by non-random constants). There are general conditions for a local martingale to be a martingale: these typically involve conditions on the quadratic variation process $[M,M]$. A very simple (but excessively strong) condition, in case $M = \int H\mathtt{d}B$ for a Brownian motion B, is that the predictable integrand H should be bounded.

Note that the bracket process $[M,M]$ can be characterized as the unique predictable process such that

$$M^2 - [M,M] \quad \text{is a local martingale}\,. \tag{5}$$

It is non-trivial that $[M,M]$ exists: see the Doob-Meyer decomposition below at Eq. (8). The bracket process is a (process-valued) quadratic form in M and so one can define the bracket between two different semimartingales M and N *via*

$$[M,N] \quad = \quad \frac{1}{4}\left([M+N,M+N]-[M-N,M-N]\right) . \tag{6}$$

In fact the bracket or quadratic variation is closely analogous to the variance of a Gaussian random variable, and the quadratic covariation is analogous to the covariance.

More generally one considers processes X which may be biased, and whose volatility may vary over time. In the discrete setting it is clear that we may decompose X as the sum $X = V + M$ of a "trend" V and an "unbiased" or martingale term M, each given in difference form by

$$\begin{aligned} \Delta V &= \mathbb{E}[X(t+\Delta t)|\mathfrak{F}_t] - X(t)\,, \\ \Delta M &= \Delta X - \Delta V\,. \end{aligned} \tag{7}$$

It is a remarkable result due to Doob and Meyer that this decomposition persists in the limit: if we can write a continuous process X as the sum

$$X \quad = \quad X(0) + V + M \tag{8}$$

of M a continuous local martingale and V a continuous process with paths of locally bounded variation (so in particular $(\mathtt{d}V)^2 = \mathtt{d}[V,V] = 0$), then this decomposition is unique. Moreover *submartingales* (X such that $\mathbb{E}[X_{t+s}|\mathfrak{F}_t] \geq X_t$) are semimartingales. The square M^2 of an L^2 martingale is a submartingale, and this allows us to define $[M,M]$ using Eq. (5) and the technique known as *localization*. One then defines quadratic variation and covariation of semimartingales by considering their local martingale parts: thus for example $\mathtt{d}[X,X] = \mathtt{d}[M,M]$ in the above.

Such processes are called (continuous) semimartingales. The Doob-Meyer decomposition Eq. (8) can be viewed as a *drift+noise* decomposition, and is thus very natural from a general modelling standpoint. We should mention here that there is a theory of stochastic calculus which permits discontinuities; the general theory is however more involved and does not suit the purposes of implementation in computer algebra, so we omit further discussion and refer the interested reader to [38]. In the remainder of the paper all the semimartingales we mention will be continuous, and we shall not say so explicitly except where we wish to emphasize their continuity.

We note at this point that *real* mathematical finance does *not* assume share prices *etc* to be martingales, but frequently does assume them to be continuous semimartingales. It would take us too far from our course to discuss this here: see for example [4].

Composition of a semimartingale X with a nonlinear smooth function f introduces a further drift term at second order, following the familiar computations used in variance-stabilization in statistics:

$$\Delta f(X) \quad \approx \quad f'(X)\Delta X \;+\; \frac{1}{2}f''(X)(\Delta X)^2 \tag{9}$$

If the process X has any volatility then we cannot ignore the second-order terms in the Taylor expansion for $\Delta f(X)$. At the infinitesimal level this yields the famous *Itô formula* for a smooth (continuously-twice-differentiable) function f of a semimartingale X:

$$\mathtt{d}f(X) \quad = \quad f'(X)\mathtt{d}X \; + \; \frac{1}{2}f''(X)\mathtt{d}[X,X] \tag{10}$$

This generalizes directly to smooth functions of several variables. There are several different ways to prove it in the literature; conventional methods use power series expansions or polynomial approximation, while a recent geometric approach is given in [23]. It should be noted that Eq. (10) generalizes to the case of f which are the difference of convex functions, for example $f(x) = |x|$.

Stochastic calculus theory includes a theory of stochastic differential equations (*sde*)

$$\mathtt{d}X \quad = \quad f(X)\mathtt{d}B \; + \; g(X)\mathtt{d}t \tag{11}$$

with existence and uniqueness for sufficiently regular coefficients f and g; for example the important linear *sde*

$$\mathtt{d}X \quad = \quad \alpha X\mathtt{d}B \; + \; \beta X\mathtt{d}t \tag{12}$$

(for constants α, β) is solved by $X_t = X_0 \exp(\alpha B_t + (\beta - \frac{1}{2}\alpha^2)t)$. (It is an easy exercise in Itô calculus to verify this.) In particular the general theory shows us that the statistical behaviour of a semimartingale X is determined by specifying its drift $\mathtt{drift}(\mathtt{d}X)$ and its quadratic variation $(\mathtt{d}X)^2 = \mathtt{d}[X,X]$.

In some sense the continuous semimartingales are the Gaussian random variables of the world of continuous-processes. The drift $\mathtt{drift}(\mathtt{d}X)$ and bracket $(\mathtt{d}X)^2 = \mathtt{d}[X,X]$ serve the place of mean and variance, while $(\mathtt{d}X)(\mathtt{d}Y) = \mathtt{d}[X,Y]$ plays the part of covariance. Stochastic differential equation theory tells us that specification of drift, bracket, and quadratic covariance (together with initial values) determine the behaviour of the semimartingales when the coefficients are sufficiently regular.

We have chosen to describe the stochastic calculus theory first in an intuitive and approximate form for processes changing at discrete time-steps Δt, using Eqs. (1,3,4,7,9). In fact this approach actually makes full mathematical sense as written if one calls upon non-standard analysis [1]. Indeed one can then recognize stochastic differential equations exactly as definitions of processes as (infinitesimally) discrete-time processes of nonlinear autoregressive type. More classically, one uses the martingale property to show that stochastic integrals makes sense *via* limiting procedures, and interprets stochastic differentials using integration: see [36, 38, 39, 42] for book expositions at varying levels of detail.

We can now isolate the underlying structure of stochastic calculus, following Itô [14]. The formalism deals with the class $\mathcal{D}_2$ of *Itô differentials* $\mathtt{d}X$ arising from continuous semimartingales X (in fact Itô defines $\mathtt{d}X$ as the equivalence class of X under variation of initial value by addition). Stochastic integration guarantees an algebraic *module* structure for $\mathcal{D}_2$: an Itô differential $\mathtt{d}X$ when multiplied by a bounded predictable function H gives rise to another Itô differential $H\mathtt{d}X$ (the differential of the stochastic integral $\int H\mathtt{d}X$) and so $\mathcal{D}_2$ is a $\mathcal{P}$*-module*, where $\mathcal{P}$ is the algebra of bounded predictable functions.

In fact $\mathcal{D}_2$ also has a multiplication arising from the bracket process: $\mathtt{d}X$ can be multiplied by $\mathtt{d}Y$ to yield a new stochastic differential $\mathtt{d}[X,Y]$. So in algebraic language $\mathcal{D}_2$ is a $\mathcal{P}$*-algebra.* In stochastic calculus terms this $\mathcal{P}$-algebra property amounts to a fundamental identity: for bounded predictable H we have $\mathtt{d}[\int H\mathtt{d}X, Y] = H\mathtt{d}[X,Y]$.

We also have a drift operation $\mathtt{drift} : \mathcal{D}_2 \to \mathcal{D}_2$ which is $\mathcal{P}$-linear. Strictly speaking, `drift` actually maps into the smaller class $\mathcal{D}_1$ of classical, volatility-free, differentials. Again, $\mathcal{P}$-linearity of `drift` amounts to a fundamental identity of stochastic calculus: for bounded predictable H we have $\mathtt{drift}\ H\mathtt{d}X = H\mathtt{drift}\ \mathtt{d}X$.

We can produce Itô differentials from semimartingales using the Itô differential $\mathtt{d} : \mathcal{S} \to \mathcal{D}_2$, where $\mathcal{S}$ is the class of continuous semimartingales: the Itô formula tells us how to compute this for general continuous semimartingales once we know how to do it for basic semimartingales (such as $\mathtt{d} : t \to \mathtt{d}t$, $\mathtt{d} : B \to \mathtt{d}B$).

Of course it is important to connect $\mathcal{D}_2$ back to the world of random processes, and we can do this using Itô integration: $\mathtt{itoIntegral} : \mathcal{D}_2 \to \mathcal{S}$ is a right-inverse to `d`.

This algebraic apparatus provides a good description of the formal manipulations carried out by workers in stochastic calculus. Before the advent of symbolic Itô calculus, using computer algebra packages, the description seemed to be largely a curiosity, though Schwartz [43, 44], Meyer [35] and Émery [11] used a similar description to link stochastic differential geometry to ideas of second-order invariance. However a proper appreciation of the algebraic context of a mathematical problem is manifestly essential if computer algebra is to make a fully effective contribution.

This is a rich and expressive algebraic context for calculations concerning semimartingales in a whole variety of applications. But there is a strong incentive to seek some kind of computer-algebra support, as the Itô formula is *second-order*: the Itô differential applied to a function of n general semimartingales yields a sum of $n + \frac{1}{2}n(n+1)$ different terms. Such a growth in complexity, together with a very effective algebraic framework, strongly motivates the requirement to implement Itô calculus in computer algebra packages.

Before turning to the history of implementations of stochastic calculus in computer algebra, we note an alternative stochastic calculus, *Stratonovich calculus*, which provides a first-order transformation rule instead of the second-order Itô formula Eq. (10), at the double price of (a) strongly restricting the class of integrands, and (b) weakening the connection with martingales – in particular the crucial notion of the *drift* or trend of a stochastic differential ceases to make algebraic sense. In fact there is an explicit (second-order) rule to convert between the two kinds of calculus. However, for the kinds of problems which we will discuss, it makes sense to treat the Itô calculus as primary and to focus on its implementation alone.

2 History of symbolic Itô calculus implementations

Previous surveys [24, 27] have covered the wider field of computer algebra in both probability *and* statistics: so here we confine ourselves to the use of computer algebra in Itô calculus. In this section we describe the history of *implementations*: some discussion of applications is to be found in Section 4.

The first example of implementation of stochastic calculus dates back as far as 1987 [18, 19]. It was carried out in the *REDUCE* computer algebra package. The original motivation arose from noticing that the infinitesimal identity $\mathtt{d}B^2 = \mathtt{d}t$ could be implemented very simply in *REDUCE* as a *substitution rule*

```
LET dB**2 = dt
```

which would be carried out whenever `dB**2` was encountered in a *REDUCE* expression. This observation was the seed for a simple but effective implementation which involved relatively little *REDUCE* code, and which followed the semimartingale theory very closely (building procedures corresponding to `d`, `drift`, `itoIntegral`, *etc*). In effect computer algebra was used to build a "mathematical interface", allowing the probabilist to set up a problem in language corresponding closely to stochastic calculus theory, and then to use the package to carry out involved formal manipulations.

Fairly full descriptions of this implementation in its mature form (referred to as *Itovsn3* in a terminology combining "Itô" with the major version number of the final version of the code) are to be found in [21, 22].

Independently Valkeila carried out some work in `Macsyma` [49, 48], based essentially on computing the Itô formula expansion Eq. (10).

In a rather different vein, Steele and Stine used *Mathematica* to implement a stochastic calculus based on *diffusions* [45, 46]. Subject to suitable regularity conditions, a diffusion X may be viewed as a semimartingale solving a stochastic differential equations in which the coefficients depend only on X at the present time, not on X at some previous time, and not on other semimartingales (Eq. (11) is an example). They have the advantage of a very close link to partial differential equation theory, but have the disadvantage that functions of diffusions are typically *not* diffusions. To see this, consider a semimartingale X which has bracket process

$$\begin{aligned} \mathtt{d}[X,X] &= \frac{1}{2}\mathtt{d}t \quad \text{when } X_t \geq 1\,, \\ & \ \mathtt{d}t \quad \text{when } X_t \leq -1\,, \end{aligned}$$

with some smooth interpolation over the range $[-1, 1]$. (The process X will then be a diffusion, but its absolute value $|X|$ cannot be one, since the quadratic variation can be recovered by inspection of the immediate past. For example, predictions about the future behaviour of $|X|$ when $|X_t| = 2$ can be improved by finding out whether X equals $+2$ or -2.) It is a rather cumbersome restriction not to be able freely to compose with functions, especially in contrast to implementations based on semimartingales, for which the Itô formula Eq. (10) assures us that (smooth) functions of semimartingales *are* semimartingales.

A defect of the *REDUCE* implementation of *Itovsn3* described above is that the user is not at all protected against inadvertently modifying data structures used by the package to hold information about the semimartingales being used (various lists, such as a list of the semimartingales themselves; a multiplication table carrying information of the form of Eq. (2); essential procedures implementing the Itô formula Eq. (10), *etc.*) The packaging facility of *Mathematica* offers some protection against such hazards, so *Itovsn3* was re-implemented in *Mathematica* using this feature [25].

To conclude this brief survey of implementations, we note that Riccomagno (following an algebraic study of integral representation of Itô integrals [40]) is now working on implementation of *Itovsn3* in `Maple`, while Cyganowski [8, 7] has developed a `Maple` package for the specific purpose of solving stochastic differential equations.

3 Description of *AXIOM* implementation

We have already mentioned the re-implementation of *Itovsn3* in *Mathematica*, undertaken in order to take advantage of *Mathematica*'s packaging facilities, so as to protect against inadvertent overwriting of implementation procedures and lists. Even this has not proved completely satisfactory, because a proper implementation of Itô calculus ought to take account of the intensely algebraic structure ($\mathcal{D}_2$, `drift`, *etc*) described at the end of Section 1. Furthermore *Mathematica* does not offer complete protection against interference between packages and system procedures and definitions (the implementation of *Itovsn3* in *Mathematica* version 2 needs to be rewritten for *Mathematica* version 3 for just this reason). The innovative computer algebra package *AXIOM* offers much in these respects, which motivated the project to use it to implement *Itovsn3*.

The principal issue is that the algebraic structure described in Section 1 requires us to work with entities (stochastic differentials) which are elements of a module (even an algebra) over $\mathcal{P}$ the family of predictable random functions. Stochastic differentials `dX` and `dY` do not fit safely into the usual facilities offered by a computer algebra package. They *could* be viewed as elements of a vector space (albeit typically of infinite real dimension), but the vector space has to permit a multiplication `dX*dY` using the $\mathcal{P}$-algebra structure. They *could* be viewed simply as general algebraic quantities (this is the way in which they are implemented in *Itovsn3* both in *Mathematica* and *REDUCE* forms), but this gives no protection for the unwary user who accidentally constructs the meaningless form `1/dX`, or performs the calculation (meaningless within the domain of Itô calculus) of `dX**0=1`.

We need a computer algebra environment in which quantities are *strongly typed*, not simply as real or complex or vector quantities, but in ways which we can specify to suit the application at hand. This is exactly the strength of *AXIOM*, which provides many tools for the implementation of new types or *domains of computation*. We can then declare a quantity `dX` to possess a type such as "stochastic differential", which then means we can subject `dX` only to those operations which we have implemented and specified as publicly available to that type.

In principle one can arrange this in *Mathematica* (and indeed to some extent in *REDUCE*): however the construction of a whole new domain of computation *ab initio* is unappealing. As well as the standard object-oriented facilities of information hiding *via* private functions *versus* public or exported functions, *AXIOM* offers facilities to use private data representations to inherit private functions from extant *AXIOM* code, and also the notion of *categories*, which systematizes a way of declaring exported functions and properties in a mathematical framework. By way of example we describe a simplified form of the *AXIOM Itovsn3* domain of stochastic differentials.

In the *AXIOM* implementation of *Itovsn3* we use as a building block the very useful computational domain of *integer expressions* (`EXPR INT`), built up recursively of kernel functions (`sin`, `log`, ...) applied to fractions of integer polynomials of variables and of further kernels. Consider for example

$$\sin\left(\frac{a^2+b}{a\,b\,\sin(a/b)}\right)\,.$$

(In fact the analogue of `EXPR INT` is the *de facto* type of many *REDUCE* expressions.) As is common in computer algebra systems, mathematical truth is stretched here: *AXIOM* considers `EXPR INT` to be a field!

At the present stage of implementation *Itovsn3* uses the domain `EXPR INT` to represent the Itô calculus notions of semimartingales and predictable functions, since the major focus of implementation is on building the right domain for stochastic differentials.

To implement stochastic differentials we require the domain of computation `SD` (abbreviating `STOCHASTIC DIFFERENTIAL`) to be both a `Module` over `EXPR INT`, and a "`Rng`" (an outrageous *AXIOM* pun for a ring without identity ...). In fact we describe `SD` to *AXIOM* as a *vector space*, because `EXPR INT` is viewed as a field and we want to make sense of `dX/a`, for example. So category statements for the domain `SD` read as follows:

```
SD:
 Category == Implementation where
```

```
Category ==> Join(Rng, VectorSpace(EXPR INT)) with
 ... (declarations of various exports including drift)
Implementation ==> ...
```

This stipulates that our new computational domain must have a multiplication "*", also addition "+" *etc*, obeying properties arising from the `VectorSpace` and "`Rng`" requirements. As noted, we also add some explicit exports including the `drift` operation.

However this is merely the public *declaration*. We must now choose the private *representation* of `SD`. This depends entirely on computational and programming convenience, as it is hidden from the user: we settle on a representation in terms of *sparse multivariate polynomials* (`SMP`) with `EXPR INT` coefficients. The variables for these sparse polynomials are the *basic stochastic differentials*, members of a further *AXIOM* domain `BSD` whose definition we do not present here. In computational terms `BSD` is merely a private list of basic stochastic differentials, exporting some constructor and accessor functions and a primitive version of the Itô differential `d`.

The choice of the structure `SMP(EXPR INT, BSD)` for implementation domain is convenient because `SMP(...)` carries algebraic structure resembling that of `SD`. We add *multiplication relations* for Itô differentials, such as `dB**2=dt` (for Brownian differential `dB` and time differential `dt`) and `dB*dt=0`, `dt**2=0`. These multiplication relations are kept in a private table belonging to `SD`, maintained by a constructor function `alterQuadVar!` which `SD` exports. (Note it is an *AXIOM* convention, shared by several other list-based computer languages, to postfix a procedure by an exclamation mark `!` if it modifies structures in place.) In the implementation part of the code for `SD` we add a new definition of multiplication "*" which modifies the `SMP` definition; however we add no code at all concerning other algebraic operations ("+", "-", ...) with the result that *AXIOM* simply uses the prior definitions of `SMP`. This economy of definition greatly lightens the task of implementation, while maintaining a reasonable regime of mathematical hygiene.

We code this up as follows:

```
SD:
 Category == Implementation where
  Category ==> Join(Rng, VectorSpace(EXPR INT)) with
   ... (various exports including drift)
  Implementation ==> SMP(EXPR INT, BSD) add
   Rep:=SMP(EXPR INT,BSD)
   ... (implementations of private objects, and those exported
        objects - such as drift and * - which are not inherited
        directly from the implementation domain)
```

Of course this is not all. In the full version of the above implementation we add various constructor and accessor functions to maintain and display tables for multiplication and drift rules, for example `alterQuadVar!(dX,dY,dH)`, which sets up or alters the multiplication rule for `dX*dY` to be equal to `dH`. We also need an Itô differential operator `d`, satisfying Itô's formula as well as `d(X)` = `dX`: the Itô formula part of this is implemented in a separate *AXIOM package* `ITO`, both to aid development (it can then be compiled separately) and because this enables us to write code once for an Itô differential to apply to a variety of different algebraic situations (complex-valued stochastic differentials, stochastic differential expressions based on floating-point arithmetic, ...). A further package `ITO2` implements secondary but extremely useful procedures to define stochastic differentials as Brownian differentials, as classical differentials, and as solutions to stochastic differential equations.

The detailed description of all this, together with source code listed in `Nuweb` scraps [5], is to be found in the technical manual reporting on the current stage of implementation of *AXIOM Itovsn3* [26]. Therefore we now turn from this summary description of the implementation to describe how it works in practice.

4 A new application of *Itovsn3* to the statistics of shape

The main focus of this section is to present a brief example of an application of symbolic Itô calculus using the new implementation in *AXIOM*. It should be emphasized that this application is a "toy" application which could easily be performed by hand: the *AXIOM* implementation is in its early stages and therefore the applications are not as weighty as those involving earlier implementations (*REDUCE*, *Mathematica*); this however is likely to change fast now that the *AXIOM* implementation is in a usable form. In any case the application we present below is new, and introduces ideas used elsewhere in a rather more extensive application in the same subject-area [28].

The statistics of shape provided the original motivation for the development of computer algebra methods in stochastic calculus [18, 19, 20]. For example it was used to give a proof that the natural *shape space* Σ^3_d for three independent identically distributed Gaussian points in $\mathbb{R}^d$ (for $d > 2$) is a hemisphere, and that there is a natural shape diffusion associated to this shape space, namely Brownian motion with a drift towards the "north pole" of the hemisphere. This generalized earlier results [16] for the cases of $d = 2$ (the shape space Σ^3_2 is a sphere, the shape diffusion is ordinary spherical Brownian motion) and $d = 3$.

The Mardia-Dryden distribution [33, 34] is a beautifully simple example of the statistics of shape: in its simplest form it describes the *shape density* arising from three independent planar points when the point distributions are planar Gaussian distributions with the same rotationally symmetric dispersion but with different means. It is natural to ask whether the obvious relationship between Gaussian distribution and Brownian motion leads to a stochastic calculus proof, and indeed it does: see [17, 32]. Here we present a different proof which takes advantage of the facilities of symbolic Itô calculus in *AXIOM*, and demonstrates how closely the *Itovsn3* code is linked to the underlying mathematics.

We begin by making *AXIOM* aware of the code for *Itovsn3*, and recording that we are dealing with `pts=3` points in `n=2` dimensions, recording a useful variable `dim`, and introducing the basic time semimartingale `t`. Notice that `t` is set up using a procedure `classicalSD!`: this is one of the secondary procedures from `ITO2` mentioned above.

```
)library BSD SD ITO ITO2
n:= 2; pts := 3; dim := n*(pts-1);
dt := introduce!(t,dt)
classicalSD!(dt)
```

Semicolons are used to separate statements on the same line, and also to suppress output. Here we omit almost all output for the sake of clarity.

In order to conserve space we omit some calculations, well understood from [16, 19, 20], which show that the *size* of the configuration of 3 points is given by a Bessel process `R`, defined as a modification of a Brownian motion in what follows (using another secondary procedure `brownianSD!`), and used to define the stochastic differential `dtau` of a new time-scale which takes account of the fact that the shape of a small configuration will change faster than that of a large one:

```
dR := introduce!(R,dR);
brownianSD!(dR,dt);
alterDrift!(dR,((dim-1)/(2*R))*dt);
dtau := dt/R^2
```

Here (in LaTeX form) is the full output from the last of these statements:

$$\frac{1}{R^2}\ dt$$

Type : `StochasticDifferential Integer`

AXIOM's strong typing is manifest: every output is accompanied by a statement giving the type of the answer; here `StochasticDifferential Integer` signals that the expression is a stochastic differential (defined using a generalization of the domain `SD` described above). This means we can apply the various operations (`drift`, multiplication, ...) available for stochastic differentials, but *AXIOM* will object if we attempt an illegal calculation such as taking the inverse. For space reasons alone we suppress such statements below: however they are a fundamental part of *AXIOM*'s communication with the user.

We continue by introducing a further Brownian motion, this time in the new time-scale `dtau` which we built using `R`. We use the timescale to introduce the process `S` measuring the shape-distance between the shape of the configuration and the shape of an equilateral triangle on the spherical shape space Σ_2^3. The notion of shape-distance comes from general arguments in shape theory; the characteristics of the diffusion `S` can also be computed using computer algebra but we omit this here.

```
dW := introduce!(W,dW)
brownianSD!(dW,dtau)
alterQuadVar!(dR,dW,0)
dS := introduce!(S,dS)
itoSDE!(dS = dW + (cos(2*S)/sin(2*S))*dtau)
```

We use `(cos(2*S)/sin(2*S))` here rather than `cot(2*S)` to facilitate cancellations at a later stage. The procedure `itoSDE!` deduces the correct second-order structure for the basic stochastic differential `dS` from the *sde* on the right-hand side of the equation. As noted in Section 3, the procedure `alterQuadVar!` alters the private multiplication table for Itô differentials. It is actually used many times covertly as a subsidiary procedure to `classicalSD!`, `brownianSD!`, and `itoSDE!`; it is necessary to use it explicitly here to define the relationship between `dR` and `dW` (and hence `dS`), because *Itovsn3* deliberately fails if it encounters an undefined stochastic differential multiplication (or indeed an undefined drift).

All other stochastic multiplications and drifts are now defined, as may be seen from the next invocation:

```
statusIto()
```

which yields output displaying the basic stochastic differentials, their drifts, and their multiplication table:

$$\begin{bmatrix} BSD & : & dR & dS & dW & dt \\ Drift & : & \frac{3}{2\ R}\ dt & \frac{\cos(2\ S)}{R^2\ \sin(2\ S)}\ dt & 0 & dt \\ & & & & & \\ dR & : & dt & 0 & 0 & 0 \\ dS & : & 0 & \frac{1}{R^2}\ dt & \frac{1}{R^2}\ dt & 0 \\ dW & : & 0 & \frac{1}{R^2}\ dt & \frac{1}{R^2}\ dt & 0 \\ dt & : & 0 & 0 & 0 & 0 \end{bmatrix}$$

At this stage *AXIOM* understands the identifiers R and S respectively as standing for the size and the shape-distance from an equilateral triangle of a triad formed by three independent Brownian points on the plane. We can now use *AXIOM Itovsn3* to calculate with these semimartingale quantities so as to find the distribution of S_t.

Invariance arguments can be used to motivate the definition of the following quantity as an appropriate surrogate for time, and an associated equation to be used later for substitutions:

```
kappa := R^2/(4*(t1-t))
kappaSub := first solve(kappa=k,t1)
```

This last command has output

$$t_1 = \frac{4\ k\ t + R^2}{4\ k}\ .$$

It is now possible to derive the Kolmogorov backwards partial differential equation for the shape `density`; the invariance considerations noted just above show the density can be given by

```
f := operator 'f
density := f(cos(2*S),kappa)
```

(where `f` represents the unknown smooth function we'd like to find), and the backwards equation is derived as

```
bde :=
  0 = numer subst(coefficient(drift d density,dt), kappaSub)
```

(here `numer` extracts the numerator, while `subst` performs the substitution indicated by `kappaSub`) with output

$$\begin{aligned} 0 = k^2\ f_{,2,2}\left(\cos\left(2\ S\right),\ k\right) + \sin\left(2\ S\right)^2\ f_{,1,1}\left(\cos\left(2\ S\right),\ k\right) + \\ \left(2\ k^2 + 2\ k\right)\ f_{,2}\left(\cos\left(2\ S\right),\ k\right) - 2\ \cos\left(2\ S\right)\ f_{,1}\left(\cos\left(2\ S\right),\ k\right) \end{aligned}$$

Here $f_{,1}$, for example, stands for the first partial derivative of f with respect to its first argument.

Of course this simply rephrases the problem: we do better by actively searching for solutions. Consider a geometrically motivated "flat-space" solution (close to Euclidean at small distance, yet using natural quantities for the spherically curved shape space geometry of Σ_2^3), and a perturbation using small changes in time-like and space directions:

```
p0 := kappa*exp(-kappa*(1-cos(2*S)))
p1 := (a0+a1/kappa+a2*(1-cos(2*S)))*p0
```

In *Itovsn3* it is easy to compute conditions which `a0`, `a1`, `a2` must satisfy:

- a zero-drift condition corresponding to the Kolmogorov backwards partial differential equation

```
zero := coefficient(drift d p1,dt)/p0
zeroDrift :=
  0 = numer simplify subst(zero,reference)
```

yielding

$$0 = \frac{4\ a_2 - 4\ a_1 + 4\ a_0}{R^2}$$

- a unit mass condition using the *AXIOM* `integrate` facility

  ```
  mass := integrate(subst(p1,kappaSub)*sin(2*S),S)
  unitMass := simplify(subst(mass,S=%pi/2)-subst(mass,S=0)) -1
  ```

  yielding

  $$\frac{((-2\ a_2 - a_0)\ k - a_2 - a_1)\ e^{(-2\ k)} + a_0\ k + a_2 + a_1}{2\ k} = 1$$

  (there *is* an *AXIOM* facility for definite integration, but it is hard to apply to this particular expression, for wholly virtuous reasons to do with strong typing, and *AXIOM*'s excellent ability to deal carefully with mathematically ambiguous situations);

- and a further condition derived by noting that the `unitMass` condition must apply for all time when `a0` *etc* are constant, so we may differentiate with respect to `k` and set it to zero:

  ```
  constancy := subst(D(2*k*unitMass,k),k=0)
  ```

  yielding

  $$(2\ a_1 - a_0)\ e^0 + a_0 = 2$$

We can now solve these three equations:

```
solution := solve([constancy, unitMass, zeroDrift], [a0,a1,a2])
```

and so obtain

$$[[a_0 = 2,\ a_1 = 1,\ a_2 = -1]] \tag{13}$$

Note this is a list of lists because this `solve` facility allows for the possibility of multiple solutions!

Substituting into `p1` we actually obtain the Mardia-Dryden density

$$(k\ \cos(2\ S) + k + 1)\ e^{(k\ \cos(2\ S) - k)} \tag{14}$$

and we can compute with `p1` to verify that this is indeed the required transition density. So *AXIOM Itovsn3* has derived the Mardia-Dryden density.

Before leaving this application we must note that, while the code is written to investigate solutions to more general situations (for example *more* than three points in two dimensions), and while Mardia and Dryden have generalized their result to such cases, the empirical approach described above is less rewarding, since the geometry becomes much more involved.

There are many further applications of various forms of symbolic Itô calculus. We have already mentioned the other applications to statistics of shape, the most recent of which [28] uses ideas similar to the above, and also the powerful computer algebra technique of *Gröbner basis algorithms*. Steele and Stine used their diffusion-based implementation to derive the celebrated Black-Scholes formula from mathematical finance [45, 46], as a demonstration of capability rather than an original application; the *Mathematica* implementation of *Itovsn3* has likewise been used to derive a result on hedging [25]. It is expected that there is a bright future for financial symbolic Itô calculus: for example current work includes the use of *Itovsn3* to explore distributions of values of perpetuities under various assumptions on inflation.

Other applications include specific calculations in probability theory: for example to compute the formula for the distribution of stochastic area of two-dimensional Brownian motion [24] (an example very similar to the application described in detail above), and to investigate the beautiful dynamical structure of the roots of Brownian polynomials [37].

Perhaps the most substantial application to date has been to coupling theory (the art of constructing paired, non-independent, copies of a given random process, typically arranged so as to meet or *couple* at some given future time). In [2] use is made of *Itovsn3* in its *REDUCE* incarnation to show the existence of a highly counter-intuitive coupling of copies of the three-dimensional process formed by two-dimensional Brownian motion together with its stochastic area. It is worth noting that this application combines orthodox computer algebra together with graphical and numerical facilities (driven from the computer algebra package itself) to seek out and to find a solution to an unorthodox control problem (a reduction of the original coupling problem), which could then be verified by hand. This last step laid to rest any possible concerns concerning correctness of the proof of existence, and provides a useful model for future applications of computer algebra: one should always seek independent verification of any complicated proof, and computer-algebra proofs are often very complicated!

The topic of applications in stochastic calculus should not be left without mention of the use of computer algebra in the simulation of diffusions. Talay [47] uses `Maple` to generate `Fortran` code for simulation of *sde*. Kloeden and Scott [31] used `Maple` to construct numerical schemes following the exposition of [29]. In as-yet unpublished work *Itovsn3* has been used to drive "automatic simulation": once a semimartingale has been defined in *Itovsn3* then in principle complete information has been given on how to simulate it, and it is a programming exercise to produce code to carry out simulation. It is on the other hand a research project, comparable to the work of Talay cited above, to produce code to deliver non-naïve simulations. For not only is it non-trivial to choose between the different methods of discretization of *sde*, as described in [29, 30], but also a major research question for computer algebra applied to numerical analysis is how to use packages to produce numerically well-conditioned code (though how to produce *optimized* code is now understood reasonably well).

5 Summary

In this article we have described how the innovative features of *AXIOM* can be used to exploit the elegant formalism of the Itô calculus, producing an implementation [26] of *Itovsn3* substantially improving on its predecessors [21, 22, 25]. We have briefly described implementation issues and have shown how to use the *AXIOM* version of *Itovsn3* to produce a new stochastic calculus derivation of the Mardia-Dryden distribution. The *AXIOM* implementation of *Itovsn3* is now operational, and uses *AXIOM*'s strong typing facilities to good effect, but much remains to be done. Certainly the present code will be further developed to improve efficiency and utilization of *AXIOM*'s novel mathematically-based programming facilities. In addition it will be re-worked so as to allow investigations into semimartingales taking values in more complicated mathematical structures: for example this is required to carry forward the investigation begun in [2]. *AXIOM*'s packaging facilities make it possible to do this in a very economical manner: for example an algorithm for the Itô differential `d` can be written once in generalized form so as to apply to a large number of rather different domains of computation.

Major advances in symbolic Itô calculus are to be expected only by application to actual problems: it is planned to apply *AXIOM Itovsn3* to investigate behaviour of multi-species epidemics as a supporting part of an interdisciplinary investigation co-funded by EPSRC and BBSRC. A particularly challenging part of this work will be to examine to what extent one can extend ideas of differential algebra, already successfully applied to this problem in a deterministic context [41], when the operation of differentiation with respect to time is replaced by the Itô stochastic differential.

Finally, and speculating over a longer time period, the unpublished experiments in automatic simulation described at the end of Section 4 suggest the possibility of using *AXIOM* as a language to specify and to control complicated statistical simulations. A possible agenda for this prospect is as follows: to build computer algebra software which allows the user to specify the simulation system and simulation objectives in entirely mathematical terms using the facilities of *AXIOM*. These being defined, the process of simulation is *in principle* well-defined, and one can envisage being able to construct the simulation code following general principles. This is exactly a description of the experimental automatic simulation facility already developed in earlier implementations of *Itovsn3*. The developers of *AXIOM* are already aware of similar possibilities, as is evidenced by their re-christening of the *AXIOM* compiler as `Aldor`, abbreviating "A Language for Describing Objects and Their Relationships" [9].

References

[1] R.M. Anderson. A non-standard representation for Brownian motion and Ito integration. *Israel Journal of Mathematics*, 25:15–46, 1976.

[2] G. Ben Arous, M. Cranston, and W.S. Kendall. Coupling constructions for hypoelliptic diffusions: Two examples. In M. Cranston and M. Pinsky, editors, *Stochastic Analysis: Summer Research Institute July 11-30, 1993.*, volume 57, pages 193–212, Providence, RI Providence, 1995. American Mathematical Society.

[3] L. Bachelier. Théorie de la spéculation. *Annales Scientifiques de l'Ecole Normale Supérieure*, III-17:21–86, 1900.

[4] M. Baxter and A. Rennie. *Financial Calculus: an introduction to derivative pricing.* Cambridge University Press, Cambridge, 1996.

[5] P. Briggs. `Nuweb` Version 0.87b: A simple literate programming tool. Published on the World-Wide Web by `preston@cs.rice.edu`, 1992.

[6] R. Brown. A brief account of microscopical observations made in the months of June, July, and August 1827, on the particles contained in the pollen of plants and on the general existence of active molecules inorganic and inorganic bodies. *Philosophical magazine (2nd series)*, 4:161–173, 1828.

[7] S. Cyganowski. A `Maple` package for stochastic differential equations. In A. Easton and R. May, editors, *Computational Techniques and Applications*, Singapore, 1996. World Science.

[8] S. Cyganowski. Solving stochastic differential equations with `Maple`. *MapleTech* (`Maple` *newsletter*), 3(2):38–40, 1996.

[9] M. Dewar. `Aldor` - the language for computer algebra. *Solve: The Newsletter for AXIOM Users*, 3:1, 1997.

[10] A. Einstein. *Investigations on the theory of Brownian movement (edited by R. Fürth).* Dover, New York, 1956.

[11] M. Émery. *Stochastic Calculus in Manifolds (with an appendix by P.-A. Meyer).* Springer-Verlag, New York - Berlin, 1989.

[12] K. Itô. Stochastic integral. *Proc. Imp. Acad. Tokyo*, 20:519–524, 1944.

[13] K. Itô. On a stochastic integral equation. *Proc. Imp. Acad. Tokyo*, 22:32–35, 1946.

[14] K. Itô. Stochastic differentials. *Applied Mathematics and Optimization*, 1:374–381, 1975.

[15] R.D. Jenks and R.S. Sutor. *AXIOM the scientific computation system.* Springer-Verlag, New York and Berlin, 1992.

[16] D.G. Kendall. The diffusion of shape. *Advances in Applied Probability*, 9:428–430, 1977.

[17] D.G. Kendall. The Mardia-Dryden shape distribution for triangles – a stochastic calculus approach. *Journal of Applied Probability*, 28:225–230, 1991.

[18] W.S. Kendall. Discussion of read paper by Clifford, Green and Pilling. *Journal of the Royal Statistical Society (Series B: Methodological)*, 49:286–287, 1987.

[19] W.S. Kendall. Symbolic computation and the diffusion of shapes of triads. *Advances in Applied Probability*, 20:775–797, 1988.

[20] W.S. Kendall. The Euclidean diffusion of shape. In D. Welsh and G. Grimmett, editors, *Disorder in Physical Systems*, pages 203–217, Oxford, 1990. Oxford University Press.

[21] W.S. Kendall. Symbolic Itô calculus: An introduction. Research Report 217, Department of Statistics, University of Warwick, 1991.

[22] W.S. Kendall. Symbolic Itô calculus: An overview. In N. Bouleau and D. Talay, editors, *Probabilités Numeriques*, pages 186–192, Rocquencourt, 1991. INRIA.

[23] W.S. Kendall. A remark on the proof of Itô's formula for C^2 functions of continuous semimartingales. *Journal of Applied Probability*, 29:216–221, 1992.

[24] W.S. Kendall. Computer algebra in probability and statistics. *Statistica Neerlandica*, 47:9–25, 1993.

[25] W.S. Kendall. Doing stochastic calculus with *Mathematica.* In H. Varian, editor, *Economic and Financial Modeling with* Mathematica, pages 214–238. Springer-Verlag, New York, 1993.

[26] W.S. Kendall. *Itovsn3 in AXIOM: modules, algebras and stochastic differentials.* Department of Statistics, University of Warwick, 1998. Research Report 328.

[27] W.S. Kendall. Computer Algebra. In P. Armitage and T. Colton, editors, *Encyclopedia of Biostatistics*, volume 1, pages 839–845. John Wiley & Sons, Chichester, 1998.

[28] W.S. Kendall. A diffusion model for Bookstein triangle shape. *Advances in Applied Probability*, 30(2):317–334, 1998.

[29] P.E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations.* Springer-Verlag, New York-Heidelberg-Berlin, 1992.

[30] P.E. Kloeden, E. Platen, and H. Schurz. *Numerical solution of SDE through computer experiments.* Springer-Verlag, New York-Heidelberg-Berlin, 1994.

[31] P.E. Kloeden and W. Scott. Construction of stochastic numerical schemes through `Maple`. *MapleTech* (`Maple` *newsletter*), 10:60–65, 1993.

[32] H.L. Le. A stochastic calculus approach to the shape distribution induced by a complex normal model. *Mathematical Proceedings of the Cambridge Philosophical Society*, 109:221–228, 1991.

[33] K.V. Mardia. Discussion of paper by D.G. Kendall. *Statistical Science*, 4:108–111, 1989.

[34] K.V. Mardia and I.L. Dryden. Shape distributions for landmark data. *Advances in Applied Probability*, 21:742–755, 1989.

[35] P.A. Meyer. Géométrie stochastique sans larmes. *Séminaire de Probabilités*, XV:44–102, 1981. Springer Lecture Notes in Mathematics 850.

[36] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications.* Springer-Verlag, New York and Berlin, 1985. (Now in 5th edition).

[37] C.J. Price. *Zeros of Brownian polynomials and Coupling of Brownian areas.* PhD thesis, Department of Statistics, University of Warwick, 1996.

[38] Ph. Protter. *Stochastic Integration and Differential Equations: A New Approach.* Springer-Verlag, New York - Berlin, 1990.

[39] D. Revuz and M. Yor. *Continuous martingales and Brownian motion.* Springer-Verlag, New York-Heidelberg-Berlin, 1991.

[40] E. Riccomagno. Rational noncommutative formal power series and iterated integral representation of a class of Ito processes. *Bollettino UMI, VII. Ser. B*, 10(1):25–50, 1996.

[41] E. Riccomagno and L.J. White. Multi-strain species modeling via differential algebra reduction. *Submitted to J. Symbolic Computation*, 1997.

[42] L.C.G. Rogers and D. Williams. *Diffusions, Markov Processes, and Martingales Volume 2: Itô Calculus*. John Wiley & Sons, Chichester and New York, 1987.

[43] L. Schwartz. Géométrie différentielle du 2ème ordre, semi-martingales et équations différentielle stochastique sur une variété différentielle. *Séminaire de Probabilités*, XVI:1–150, 1982. Springer Lecture Notes in Mathematics 921.

[44] L. Schwartz. *Semimartingales and their Stochastic Calculus on Manifolds*. Les Presses de l'Université de Montréal, Montréal, 1984.

[45] J.M. Steele and R.A. Stine. Applications of *Mathematica* to the stochastic calculus. In *ASA Proceedings of the Statistical Computing Section*, pages 11–19, Washington DC, 1991. American Statistical Association.

[46] J.M. Steele and R.A. Stine. *Mathematica* and diffusions. In H. Varian, editor, *Economic and Financial Modeling with* Mathematica, pages 192–213. Springer-Verlag, New York, 1993.

[47] D. Talay. Presto: a software package for the simulation of diffusion processes. *Statistics and Computing*, 4(4):247–252, 1994.

[48] E. Valkeila. Computer algebra and stochastic analysis – some possibilities. *CWI Newsletter*, 4:229–238, 1991.

[49] E. Valkeila. Some remarks on computerized stochastic analysis. In H. Apiola, M. Laine, and E. Valkeila, editors, *Proceedings of the Workshop on Symbolic and Numeric Computation*, pages 173–175, Helsinki, 1991. Computing Centre, University of Helsinki.

[50] J.C. Watkins. Mechanical models for cell movement - locomotion, translocation, migration. *Journal of Applied Probability*, 34(4):827–846, 1998.

Other University of Warwick Department of Statistics
Research Reports authored or co–authored by W.S. Kendall.

161: The Euclidean diffusion of shape.
162: Probability, convexity, and harmonic maps with small image I: Uniqueness and fine existence.
172: A spatial Markov property for nearest–neighbour Markov point processes.
181: Convexity and the hemisphere.
202: A remark on the proof of Itô's formula for C^2 functions of continuous semimartingales.
203: Computer algebra and stochastic calculus.
212: Convex geometry and nonconfluent Γ-martingales I: Tightness and strict convexity.
213: The Propeller: a counterexample to a conjectured criterion for the existence of certain convex functions.
214: Convex Geometry and nonconfluent Γ-martingales II: Well–posedness and Γ-martingale convergence.
216: *(with E. Hsu)* Limiting angle of Brownian motion in certain two–dimensional Cartan–Hadamard manifolds.
217: Symbolic Itô calculus: an introduction.
218: *(with H. Huang)* Correction note to "Martingales on manifolds and harmonic maps."
222: *(with O.E. Barndorff-Nielsen and P.E. Jupp)* Stochastic calculus, statistical asymptotics, Taylor strings and phyla.
223: Symbolic Itô calculus: an overview.
231: The radial part of a Γ-martingale and a non-implosion theorem.
236: Computer algebra in probability and statistics.
237: Computer algebra and yoke geometry I: When is an expression a tensorΓ
238: *Itovsn3*: doing stochastic calculus with *Mathematica*.
239: On the empty cells of Poisson histograms.
244: *(with M. Cranston and P. March)* The radial part of Brownian motion II: Its life and times on the cut locus.
247: Brownian motion and computer algebra (Text of talk to BAAS Science Festival '92, Southampton Wednesday 26 August 1992, with screenshots of illustrative animations).
257: Brownian motion and partial differential equations: from the heat equation to harmonic maps (Special invited lecture, 49th session of the ISI, Firenze).
260: Probability, convexity, and harmonic maps II: Smoothness via probabilistic gradient inequalities.
261: *(with G. Ben Arous and M. Cranston)* Coupling constructions for hypoelliptic diffusions: Two examples.
280: *(with M. Cranston and Yu. Kifer)* Gromov's hyperbolicity and Picard's little theorem for harmonic maps.

292: Perfect Simulation for the Area-Interaction Point Process.
293: *(with A.J. Baddeley and M.N.M. van Lieshout)* Quermass-interaction processes.
295: On some weighted Boolean models.
296: A diffusion model for Bookstein triangle shape.
301: COMPUTER ALGEBRA: an encyclopaedia article.
308: Perfect Simulation for Spatial Point Processes.
319: Geometry, statistics, and shape.
321: From Stochastic Parallel Transport to Harmonic Maps.
323: *(with E. Thönnes)* Perfect Simulation in Stochastic Geometry.
325: *(with J.M. Corcuera)* Riemannian barycentres and geodesic convexity.
327: Symbolic Itô calculus in *AXIOM*: an ongoing story.
328: *Itovsn3* in *AXIOM*: modules, algebras and stochastic differentials.
331: *(with K. Burdzy)* Efficient Markovian couplings: examples and counterexamples.
333: Stochastic calculus in *Mathematica*: software and examples.
341: Stationary countable dense random sets.

Also see the following related preprints
317: E. Thönnes: Perfect Simulation of some point processes for the impatient user.
334: M.N.M. van Lieshout and E. Thönnes: A Comparative Study on the Power of van Lieshout and Baddeley's J-function.

If you want copies of any of these reports then please mail your requests to the secretary of the Department of Statistics, University of Warwick, Coventry CV4 7AL, UK.