Stochastic Integrals and Their Expectations

Wilfrid S. Kendall

This article explains how the *Itovsn3* package can be extended to add various properties and rules for `ItoIntegral`, which represents a stochastic or Itô integral. This allows us to introduce a further expectation operator and compute suitable expectations involving Itô integrals.

■ Introduction

This article describes the *Mathematica* package *ItoIntegralRules* that provides facilities to simplify and compute expectations of stochastic integrals. *ItoIntegral-Rules* extends the previous package *Itovsn3* [1, 2] that implements stochastic calculus within *Mathematica*.

Stochastic calculus is famous for providing the foundations for modern mathematical finance and is also used extensively in a large number of other areas of applied probability. The introductory text by Øksendal [3] strikes an excellent balance between theory and accessibility. Here we give a very brief review of the underlying concepts. A central notion for stochastic calculus is that of a (continuous) *semimartingale*: a random process X that can be written as the sum of a *local martingale* M (for example, Brownian motion) and a *drift process* V (a continuous process of locally bounded variation, typically the solution of some conventional differential equation). The decomposition $X = X(0) + M + V$ is unique and can be thought of as a decomposition of X into signal V plus noise M. Fundamental to the theory of stochastic calculus is *Itô's lemma*: if $f(X)$ is a smooth function of the semimartingale X, then

$$f(X) = f(X(0)) + \int f'(X)\, dX + \frac{1}{2} \int f''(X)\, d[X, X],$$

where $[X, X] = [M, M]$ is the quadratic variation, the unique nondecreasing process such that $M^2 - [M, M]$ is a local martingale begun at 0. In the case when M is Brownian motion, we find $[M, M] = t$. Care has to be taken when interpreting the integral $\int f'(X)\, dX$: nontrivial continuous local martingales M do not possess bounded variation, so the component $\int f'(X)\, dM$ of $\int f'(X)\, dX = \int f'(X)\, dM + \int f'(X)\, dV$ must be interpreted as a *stochastic* or *Itô integral*. (Since V is of locally bounded variation, the interpretation of $\int f'(X)\, dV$ is strictly classical.)

Itô's lemma, in conjunction with martingale theory, permits us to calculate effectively with semimartingales in *Mathematica*. In *Itovsn3* [1, 2] the underlying algebra of stochastic calculus is implemented as an algebra of *stochastic differentials* dX, dM, and dV. This has facilitated several investigations into applied probability problems: examples given in [2] include explorations of the statistical theory of shape, coupling of diffusions, and computation of distributions of special random processes. The underlying principle of *Itovsn3* is to recognize a second-order algebraic structure of differentials corresponding to the formula for Itô's lemma. Thus semimartingales X have stochastic differentials dX that can be multiplied together to obtain a differential measure of volatility of the semimartingale (for example, $dX^2 = d[X, X]$) and that possess drift parts that capture the underlying trend (Drift[dX] = dV if $X = X(0) + M + V$).

The need to perform some calculations related to an image analysis problem [4, 5] supplied the initial motivation to extend *Itovsn3* by adding the package *ItoIntegralRules* to more fully implement a notion of *Itô integral*, such as $\int f'(X)\, dM$ or $\int f'(X)\, dX$. Itô integrals $\int g(X)\, dX$ are represented in *Itovsn3* using placeholders `ItoIntegral[g dM]` that possess the bare minimum of properties (loosely speaking, `ItoIntegral[dM]` → `M`). The new package *ItoIntegralRules* adds facilities to simplify expressions involving `ItoIntegral` in various ways and also adds an expectation operator $\mathbb{E}$. In particular, this allows us to address the calculations arising from the image analysis problem, which requires the derivation and further manipulation of formulas for means and variances for integrated Ornstein–Uhlenbeck processes. These specific calculations can of course be performed directly by hand; however, the computational framework provided by *ItoIntegralRules* covers a much wider range of possible calculations, so it should be of use elsewhere.

This article is divided into three sections: the first summarizes the issues of simplification of expressions involving `ItoIntegral`, the second introduces a notion of expectation and its interaction with `ItoIntegral`, and the conclusion discusses possibilities for further work.

□ Related Work

There are other implementations of stochastic calculus within a computer algebra package. Steele and Stine [6] adopt a diffusion-based approach, which has been developed further by Mark Fisher in the ItosLemma.m package [7]. Cyganowski [8] describes an approach using Maple, including a solver for stochastic differential equations.

□ Installation

The installation of *Itovsn3* and *ItoIntegralRules* follows the usual procedure for *Mathematica* packages. Unpack the zip archive file Itovsn3.zip (see Additional Material) either in the current working directory or in the Applications subdirectory of *Mathematica*'s AddOns directory (in the second case the packages will load no matter what is the current working directory). This will place the files init.m (which contains the package *Itovsn3*), ItoIntegralRules.m, and ItoIntegral-Tests.nb in the Itovsn3 subdirectory. The accompanying notebook, ItoIntegral-Tests.nb, contains detailed examples and unit tests for *ItoIntegralRules*.

After installation, *Itovsn3* can be loaded and initialized and a single Brownian motion B can be introduced by

In[1]:=
```
Needs["Itovsn3`"];
ItoReset[t, dt];
BrownSingle[B, 0];
```

■ Simplification

ItoIntegralRules implements properties for `ItoIntegral[`*dX*`]` using an approach based on a family of simplification rules, exported by the package ItoIntegral-Rules.m. Rule names are prefixed by `ItoIntegral` or `ItoExpect` to avoid name clashes. The rule-based approach is preferred to canonical simplification techniques because, as a consequence of Itô's lemma, there will typically be inequivalent simplification strategies. For example, if B is Brownian motion, then Itô's lemma can be applied together with $[B, B] = t$ (or, in differential form, $dB^2 = dt$) to show

$$2 \int t\, B\, dB = t\, B^2 - \int B^2\, dt - \int t\, dt,$$

and the preferred choice between the two equivalent forms will depend on context, in particular whether it is more convenient for expressions to contain stochastic or classical integrals. We now survey the major rules and briefly illustrate their use in simplification. More detailed information and unit tests can be found in ItoIntegralTests.nb.

□ Additivity and Linearity

ItoIntegralRules implements additivity: $\int (X + Y)\, dZ = \int X\, dZ + \int Y\, dZ$ is applied automatically once the package is loaded. We exemplify this by considering `ItoIntegral[a dB-b dB]`, representing the stochastic integral $\int a\, dB - \int b\, dB$.

In[4]:=
```
ItoIntegral[a dB - b dB]
```

Out[4]=
```
ItoIntegral[a dB - b dB]
```

In[5]:=
```
Needs["Itovsn3`ItoIntegralRules`"]
```

In[6]:=
```
ItoIntegral[a dB - b dB]
```

Out[6]=
```
ItoIntegral[a dB] - ItoIntegral[b dB]
```

It would normally be convenient to extract constant coefficients a and b: *ItoIntegrationRules* defines `ItoIntegralExpandRule` to perform this. In this particular case, *Itovsn3* can apply the original rules for `ItoIntegral` after linear expansion to deliver a complete solution.

In[7]:=
```
? ItoIntegralExpandRule
```

```
A rule to expand the argument of ItoIntegral,
  pulling out factors which are non-random and non-constant.
```

In[8]:=
```
SetAttributes[{a, b}, Constant];
ItoIntegral[a dB - b dB] /. ItoIntegralExpandRule
```

Out[9]=
```
a B - b B
```

Relationship to Classical Integration

If the Itô integral involves no semimartingale terms other than the time term t (and its differential dt), then it can be rewritten as a classical time integral. *ItoIntegralRules* supplies `ItoIntegralClassicRule` to make this transformation and then the integral may possibly evaluate. (An implementation issue should be noted here. *Itovsn3* is based on the total differentiation `Dt` operation, which assumes dependence unless explicitly stated otherwise. `Integrate` assumes symbols are constant by default. As long as the only quantities to vary in time are semimartingales, which would be the case in normal use of *Itovsn3*, this presents no problems.)

In[10]:=
```
? ItoIntegralClassicRule
```

```
A rule which attempts to convert ItoIntegral into a classical integral.
```

In[11]:=
```
ItoIntegral[t dt + B dB] /. ItoIntegralClassicRule
```

Out[11]=
$$\frac{t^2}{2} + \text{ItoIntegral}[B\,dB]$$

A Simplification Strategy

These rules can be applied in a variety of ways. In important special cases their application can be systematized. Here is a simple example. Consider expressions formed from just one Brownian motion B, as defined earlier, and time t using only addition, multiplication, and (possibly iterated) integration with respect to B and t, which we shall call "polynomial semimartingales." These can be reduced to expressions that involve classical integrals alone (no Itô integrals) by repeated application of specific formulas derived from stochastic integration by parts, itself derived from Itô's lemma:

$$\int\left(\int p\,dt\right) q\,dB = \int p\,dt \int q\,dB - \int\left(\int q\,dB\right) p\,dt.$$

Taking into account the various structural variations (t, t^a, B, … as in the lists `l1`, `l2` following) for monomials p and q, there are 80 different rules to be considered! It is therefore convenient (and more reliable) to construct the various resulting rules automatically as follows (the rule set is also *tested* automatically in the accompanying notebook ItoIntegralTests.nb).

```
In[12]:= ItoIntegralRewriteRuleset =
          Module[{a, b, c, d, conv, l1, l2},
           SetAttributes[{a, b, c, d, conv}, Constant];
           conv = Map[(# → Condition[Pattern[#, Blank[]], Greater[#, -1]]) &,
             {a, b, c, d}];
           l1 = {t, t^a, B, B^b, t B, t B^b, t^a B, t^a B^b};
           l2 = {t, t^c, B, B^d, t B, t B^d, t^c B, t^c B^d};
           Map[((#[[1]] /. conv) → #[[2]]) &, Flatten[
             {
              Map[Solve[
                 # ItoIntegral[dB] == (ItoIntegral[ItoD[# ItoIntegral[dB]]] /.
                    ItoIntegralExpandRule), ItoIntegral[# dB]] &, l1],

              Map[Solve[ItoIntegral[# dt] ItoIntegral[dB] ==
                  (ItoIntegral[ItoD[ItoIntegral[# dt] ItoIntegral[dB]]] /.
                    ItoIntegralExpandRule),
                 ItoIntegral[ItoIntegral[# dt] dB]] &, l2],

              Map[Solve[ItoIntegral[#[[2]] dt] ItoIntegral[#[[1]] dB] ==
                  (ItoIntegral[ItoD[ItoIntegral[#[[2]] dt]
                       ItoIntegral[#[[1]] dB]]] /. ItoIntegralExpandRule),
                 ItoIntegral[ItoIntegral[#[[2]] dt] #[[1]] dB]] &,
               Outer[List, l1, l2], {2}]
             }
            ]
           ]
          ];
```

The rule set must be applied iteratively to suitable expressions until they stop changing, so we use `FixedPoint` to construct an appropriate function. (Note the argument `iter`, controlling maximum number of iterations, is set by default to `Infinity` since there is no *a priori* upper bound on the number of iterations required to simplify a general polynomial semimartingale.)

```
In[13]:= applyItoRewriteRules[x_, iter_: Infinity] :=
          FixedPoint[(# /. ItoIntegralExpandRule /. ItoIntegralRewriteRuleset /.
               ItoIntegralClassicRule // Expand) &, x, iter]
```

We can test this simplification procedure on a famous result from stochastic calculus: the family of Hermite polynomials forms a structure that is preserved by Itô integration.

```
In[14]:= H[k_, x_, t_] := (2 t)^(k/2) HermiteH[k, x/Sqrt[2 t]] // Expand;
```

With this definition, we have

$$\int H[n, B, t]\, dB = \frac{H[n+1, B, t]}{2\,(n+1)}$$

and here we test this for the first 10 values of n.

In[15]:=
```
Table[
  Expand[H[n + 1, B, t]/(2 (n + 1))] ==
   applyItoRewriteRules[ItoIntegral[H[n, B, t] dB]],
  {n, 1, 10}]
```

Out[15]=
```
{True, True, True, True, True, True, True, True, True, True}
```

Variations on this approach can be devised for polynomial semimartingales based on time t and n independent Brownian motions: see Gaines [9, 10] who applies the notion of Lyndon bases for shuffle products on free algebras. Rather than pursuing this, we now turn to consider expectations of stochastic integrals.

■ Expectation

Itô's lemma may be employed in computation of expectations of semimartingale expressions as follows. If we wish to evaluate $\mathbb{E}[f(X)]$, then we may expand $f(X)$ using Itô's lemma. For well-behaved f (e.g., polynomial functions of Brownian motion), we may replace the differential dX in the expectation of the stochastic integral $\mathbb{E}[\int f'(X)\,dX]$ by its drift dV to obtain $\mathbb{E}[\int f'(X)\,dV]$. If the drift is zero (as is the case for Brownian motion), the integral then vanishes; if the drift is deterministic (e.g., the time process t), then we may simplify further to obtain $\int \mathbb{E}[f'(X)]\,dV$. It follows by induction that we can evaluate an expectation completely if the semimartingale expression X is a combination of linear operations, multiplication, and stochastic integration performed on time t and n independent Brownian motions (what we called a polynomial semimartingale in the previous section).

ItoIntegralRules therefore defines an expectation operator $\mathbb{E}$, which possesses basic linearity properties, and an associated function $\mathbb{EE}$, which applies transformations of the previous form whenever the semimartingale expression is a polynomial semimartingale.

□ Examples

We first consider some simple examples of expectations of polynomial semimartingales. Here is a computation of $\mathbb{E}\left[\left(\int B^4\,dB\right)^2\right]$.

In[16]:=
```
𝔼[ItoIntegral[B^4 dB]^2]
```

Out[16]=
```
𝔼[ItoIntegral[B^4 dB]^2]
```

In[17]:=
```
𝔼[ItoIntegral[B^4 dB]^2] /. 𝔼 → 𝔼𝔼
```

Out[17]=
```
21 t^5
```

Higher-order powers can be dealt with in an equally direct manner, though with increasing computational effort. Here we tabulate $\mathbb{E}[(1+\int B^4\, dB)^n]$ for values of n up to 5.

```
In[18]:= Table[EE[ (1 + ItoIntegral[B^4 dB])^n ], {n, 1, 5}] // TableForm

Out[18]//TableForm=
         1
         1 + 21 t^5
         1 + 63 t^5
         1 + 126 t^5 + 2967183 t^10 / 5
         1 + 210 t^5 + 2967183 t^10
```

Iterated integrals can be disposed of in a similar fashion. Here we evaluate $\mathbb{E}\left[\left(\int e^t \int e^{-t}\, dB\, dt\right)^8\right]$. Note that EE can deal with nonpolynomial functions of t.

```
In[19]:= EE[ItoIntegral[dt e^t ItoIntegral[dB e^-t]]^8]

Out[19]= 105/16 (3 - 4 e^t + e^(2 t) + 2 t)^4
```

Here we evaluate $\mathbb{E}\left[\left(\int \left(\int B \sin(t)\, dt\right)^2 dB\right)^2\right]$.

```
In[20]:= EE[ItoIntegral[ItoIntegral[B Sin[t] dt]^2 dB]^2]

Out[20]= 3/256 (72 t + 96 t^3 + 320 t Cos[2 t] + 28 t Cos[4 t] -
            160 Sin[2 t] + 128 t^2 Sin[2 t] - 25 Sin[4 t] + 8 t^2 Sin[4 t])
```

Nonpolynomial semimartingales are left unsimplified if the nonpolynomial part involves Brownian motions, as in this evaluation of $\mathbb{E}[(B+\sin(B))^4]$.

```
In[21]:= EE[ (B + Sin[B])^4]

Out[21]= 3 t^2 + 4 E[B^3 Sin[B]] + 6 E[B^2 Sin[B]^2] + 4 E[B Sin[B]^3] + E[Sin[B]^4]
```

There are further techniques available for dealing with nonpolynomial semimartingales. For example, consider the evaluation of $\mathbb{E}[\sin(B)^2]$. We could of course evaluate this directly using the density for the random variable B, and this itself can be automated using *mathStatica* [11]. However, we can also make progress using two further *ItoIntegralRules* (`ItoExpectItoIntegralRule`, `ItoExpectExpandRule`), which are components of EE. The first of these implements the interplay between expectation and drift described at the start of this section, while the second expands $\mathbb{E}$ linearly and extracts nonrandom terms.

```
In[22]:= With[{X = Sin[B]^2},
           EE[X] /. ItoExpectItoIntegralRule /. Cos[x_]^2 → 1 - Sin[x]^2 /.
             ItoIntegralExpandRule /. ItoExpectExpandRule]

Out[22]= t - 2 ItoIntegral[dt E[Sin[B]^2]]
```

We have thus obtained a recursive expression for $\mathbb{E}[\sin(B)^2]$ that can now be used to form a differential equation by further developing this code.

```
In[23]:= With[{X = Sin[B]^2},
          DSolve[{D[x[t], t] == D[EE[X] /. ItoExpectItoIntegralRule /.
                   Cos[x_]^2 → 1 - Sin[x]^2 /. ItoIntegralExpandRule /.
                 ItoExpectExpandRule /. E[X] → x[t] /.
               ItoIntegralClassicRule, t],
             x[0] == EE[InitialValue[0, X]]}, x[t], t]]
```

```
Out[23]= {{x[t] → 1/2 ⅇ^(-2 t) (-1 + ⅇ^(2 t))}}
```

So we obtain

$$\mathbb{E}[\sin(B)^2] = \frac{1}{2}(1 - e^{-2t}).$$

Further examples can be found in ItoIntegralTests.nb. This differential equation approach can be applied even when we require the expectation of a quantity that is not a simple function of Brownian motion. See, for example, the treatment of the distribution of the stochastic area integral $\int A\, dB - \int B\, dA$ in [2].

□ Calculations for the Ornstein–Uhlenbeck Process

The original motivation for this work was to provide an environment to aid computations of expectations of quantities associated with the integrated Ornstein–Uhlenbeck process. To illustrate this, we use a pair of stochastic differential equations

$$dU = dB - \frac{\alpha}{2} U \,\mathrm{dt}$$
$$dX = U \,\mathrm{dt}$$

to define an Ornstein–Uhlenbeck process and its integrated variant. To do this in *Itovsn3* we use `Itosde`.

```
In[24]:= SetAttributes[{α, U0, X0}, Constant];
         Itosde[U, dU == dB - α/2 U dt, U0];
         Itosde[X, dX == U dt, X0];
```

Note that it must be stated explicitly that both α and the initial values $U0$, $X0$ are `Constant` in time. It is possible to solve these linear stochastic differential equations in closed form: $U = e^{-\alpha t/2}(U(0) + \int e^{\alpha t/2}\, dB)$ and $X = X(0) + \int U\, dt$. We express the solutions using `ItoIntegral`:

```
In[27]:= UU = ⅇ^(-α t/2) (U0 + ItoIntegral[ⅇ^(α t/2) dB])
```

```
Out[27]= ⅇ^(-t α/2) (U0 + ItoIntegral[dB ⅇ^(t α/2)])
```

and

```
In[28]:= XX = X0 + ItoIntegral[U dt] /. U → UU
```

```
Out[28]= X0 + ItoIntegral[dt ⅇ^(-t α/2) (U0 + ItoIntegral[dB ⅇ^(t α/2)])]
```

and verify directly that they satisfy the relevant stochastic differential equations:

In[29]:=
```
ItoD[UU] == (dB - α/2 UU dt && InitialValue[0, UU] == U0) // Simplify
```
Out[29]=
```
True
```
In[30]:=
```
Simplify[ ItoD[XX] == UU dt && InitialValue[0, XX] == X0 ]
```
Out[30]=
```
True
```

We now compute mean values

In[31]:=
```
𝔼𝔼[UU]
```
Out[31]= $e^{-\frac{t\alpha}{2}}\,\mathrm{U0}$

In[32]:=
```
𝔼𝔼[XX] // Expand
```
Out[32]= $\mathrm{X0}+\frac{2\,\mathrm{U0}}{\alpha}-\frac{2\,e^{-\frac{t\alpha}{2}}\,\mathrm{U0}}{\alpha}$

and the variance-covariance matrix (using $\mathbb{Cov}[X, Y] = \mathbb{EE}[X, Y] - \mathbb{EE}[X]\,\mathbb{EE}[Y]$).

In[33]:=
```
Outer[ℂov, {B, UU, XX}, {B, UU, XX}] // MatrixForm
```
Out[33]//MatrixForm=

$$\begin{pmatrix} t & \frac{2}{\alpha}-\frac{2e^{-\frac{t\alpha}{2}}}{\alpha} & -\frac{4}{\alpha^2}+\frac{4e^{-\frac{t\alpha}{2}}}{\alpha^2}+\frac{2t}{\alpha} \\ \frac{2}{\alpha}-\frac{2e^{-\frac{t\alpha}{2}}}{\alpha} & \frac{1}{\alpha}-\frac{e^{-t\alpha}}{\alpha} & \frac{2}{\alpha^2}+\frac{2e^{-t\alpha}}{\alpha^2}-\frac{4e^{-\frac{t\alpha}{2}}}{\alpha^2} \\ -\frac{4}{\alpha^2}+\frac{4e^{-\frac{t\alpha}{2}}}{\alpha^2}+\frac{2t}{\alpha} & \frac{2}{\alpha^2}+\frac{2e^{-t\alpha}}{\alpha^2}-\frac{4e^{-\frac{t\alpha}{2}}}{\alpha^2} & -\frac{12}{\alpha^3}-\frac{4e^{-t\alpha}}{\alpha^3}+\frac{16e^{-\frac{t\alpha}{2}}}{\alpha^3}+\frac{4t}{\alpha^2} \end{pmatrix}$$

Computation of the fourth central moment is equally direct.

In[34]:=
```
𝔼𝔼[(XX - 𝔼𝔼[XX])^4]
```
Out[34]= $\frac{48\,e^{-2t\alpha}\left(-1+4\,e^{\frac{t\alpha}{2}}+e^{t\alpha}\,(-3+t\,\alpha)\right)^2}{\alpha^6}$

The image analysis application required the derivation of the conditional distribution of X and U at a specified time s given the values of X and U at 0 and t, for $0 < s < t$. Since (X, U) is a Gaussian process, we can find this by straightforward use of the `Statistics`MultinormalDistribution`` package once the means and variance-covariance matrix are calculated for the various values of X and U at times 0, s, and t. Of course it is possible to derive the conditional distribution by hand; an advantage of working in *Mathematica* is that we are then able to proceed directly to simulation experiments and so forth.

■ Conclusion

We have shown in this article (and in ItoIntegralTests.nb) how the package *Itovsn3* can be extended to provide the ability to manipulate stochastic integrals and compute expectations of stochastic calculus quantities. Further extensions are possible—we have already noted Gaines' work [9, 10] on simplification of polynomial semimartingales based on several independent Brownian motions. A more ambitious extension would be to generalize and automate the differential equation strategy described here for computing expectations, such as $\mathbb{E}[\sin(B)^2]$. This would be a demanding project involving the automatic recognition of differential equations satisfied by the expectation in question. We have also noted the possibility of interaction with other packages such as *mathStatica*. For example, Colin Rose has pointed out how the definition of $\mathbb{EE}$ can be modified so as to invoke *mathStatica*'s `Expect` function when the expression has been determined not to be a polynomial semimartingale. This would provide a direct means of computing $\mathbb{E}[\sin(B)^2]$; however, differential equation techniques would still be required when dealing with expressions involving stochastic integrals.

■ Acknowledgments

This work was supported by EPSRC grant GR/M75785. I am grateful for many helpful discussions with my collaborators on this grant: Abhir Bhalerao, Elke Thönnes, and Roland Wilson. It is a pleasure to acknowledge the hospitality of David Nualart and the Statistics Department of the University of Barcelona during part of the writing process. Thanks are also due to the insightful remarks of two referees, and I am particularly indebted to Colin Rose for his enthusiastic editorial contributions, which ranged from incisive stylistic advice all the way to source-code optimization!

Calculations were carried out on a Sun Ultra-10 workstation using *Mathematica* 4.1.

■ References

[1] W. S. Kendall, "Itovsn3: Doing Stochastic Calculus with *Mathematica*," *Economic and Financial Modeling with Mathematica* (H. R. Varian, ed.), New York: Springer-Verlag, 1993 pp. 214–239.

[2] W. S. Kendall, "Stochastic Calculus in *Mathematica:* Software and Examples," *University of Warwick Department of Statistics Research Report* 333 (Jan 2003) www.warwick.ac.uk/go/wsk/ppt/333.pdf.

[3] B. K. Øksendal, *Stochastic Differential Equations*, 5th ed., Berlin: Springer Universitext, 1998.

[4] A. Bhalerao, E. Thönnes, W. S. Kendall, and R. G. Wilson, "Inferring Vascular Structure from 2D and 3D Imagery," in *Medical Image Computing and Computer-Assisted Intervention, Proceedings MICCAI 2001*, (W. J. Niessen and M. A. Viergever, eds.), pp. 820–828; *Lecture Notes in Computer Science 2208*, Springer, 2001 pp. 820–828; *University of Warwick Department of Statistics Research Report* 392 (Jan 2002) www.warwick.ac.uk/go/wsk/ppt/392.pdf.

[5] E. Thönnes, A. Bhalerao, W. S. Kendall, and R. G. Wilson, "A Bayesian Approach to Inferring Vascular Tree Structure from 2D Imagery," in *International Conference on Image Processing*, *Proceedings ICIP 2002*, (W. J. Niessen and M. A. Viergever, eds.), pp. 937–939; *University of Warwick Department of Statistics Research Report* 391 (Oct 2001) www.warwick.ac.uk/go/wsk/ppt/391.pdf.

[6] J. M. Steele and R. A. Stine, "*Mathematica* and Diffusions," *Economic and Financial Modeling with Mathematica*, (H. R. Varian, ed.), New York: Springer-Verlag, 1993 pp. 192–214.

[7] M. Fisher, "ItosLemma.m" *MathSource* (May 20, 2002) library.wolfram.com/database/MathSource/1170 (or www.markfisher.net/~mefisher/mma/mathematica.html).

[8] S. Cyganowski, "Solving Stochastic Differential Equations with Maple" *Computer Algebra: Maple Tech*, **3**(2), 1986 pp. 35–40.

[9] J. G. Gaines, "The Algebra of Iterated Stochastic Integrals," *Stochastics and Stochastics Reports,* **49**(3–4), 1994 pp. 169–179.

[10] J. G. Gaines, "A Basis for Iterated Stochastic Integrals," *Mathematics and Computers in Simulation* [online], **38**(1–3), 1995 pp. 7–11. dx.doi.org/10.1016/0378-4754(93)E0061-9.

[11] C. Rose and M. D. Smith, *Mathematical Statistics with Mathematica*, Springer Texts in Statistics, New York: Springer-Verlag, 2002.

■ Additional Material

Itovsn3.zip (zip archive containing the files init.m, ItoIntegralRules.m, and ItoIntegralTests.nb).

Available at www.mathematica-journal.com/issue/v9i4/download.

About the Author

Wilfrid S. Kendall is a professor of statistics at the University of Warwick, with varied research interests primarily concerning the way computing and geometry influence probability and statistics.

Wilfrid S. Kendall
Department of Statistics
University of Warwick
Coventry CV4 7AL, UK
w.s.kendall@warwick.ac.uk
www.warwick.ac.uk/go/wsk