Evaluating first-passage probabilities for spectrally one-sided Lévy processes

L.C.G. Rogers[1]

Abstract. Fast stable methods for inverting multidimensional Laplace transforms have been developed in recent years by Abate, Whitt and others. We use these methods here to compute numerically the first-passage-time distribution for a spectrally one-sided Lévy process; the basic algorithm is not easy to apply, and we have to develop a variant of it. The numerical performance is as good as the original algorithm.

Keywords: Lévy process, spectrally one-sided, Laplace transform, first pasage distribution

AMS Subject Classifications: 65U05, 60J30, 60J75

[1]Department of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK (phone = +44 1225 826224, fax = +44 1225 826492, e-mail = lcgr@maths.bath.ac.uk). *Journal of Applied Probability* **37**, 2000.

Evaluating first-passage probabilities for spectrally one-sided Lévy processes

L.C.G. Rogers *University of Bath*

1 Introduction.

The celebrated Wiener-Hopf factorisation of a Lévy process X gives an expression linking the laws of the maximum and the minimum of X up to an independent exponential time. In general, it is hard to make use of this factorisation, since in order to find one of the laws, one needs in effect to know the other; however, in the case of a spectrally negative Lévy process (that is, a Lévy process which only jumps down [2]), it is possible to make progress. The reason for this is that the law of the maximum at the independent exponential time is easily shown [3] to be again *exponential*. Thus, apart from the parameter of the distribution, the law of the maximum is known, and the law of the minimum can be deduced. Strictly speaking, we can only deduce the Laplace transform of the law, and we now have to invert what is a double transform, in time and in the spatial variable. It is here that the methods due to Dubner & Abate (1968), and Hosono (1981), developed by Abate & Whitt (1992a), (1995), and extended to the multidimensional setting by Choudhury, Lucantoni & Whitt (1994) may be applied. The idea of the method is basically a Fourier inversion integral, performed by integrating up a suitably-chosen contour $a + i\mathbb{R}$, where the integral is approximated by a trapezium-rule sum, with equally-spaced points. The spacing of the points is chosen so that at the place where we want to evaluate the original function, the complex exponential in the Fourier integral alternates in sign, so that the inversion recipe is the sum of an alternating series; Euler summation is used as an acceleration technique. Choice of the parameter a allows good control over the aliasing error, and the method proves to be fast, accurate and stable.

While this method has all the nice numerical properties one could wish for, in many applications it may be hard to apply because one does not have the Laplace transform of the desired law in closed form; the moment-generating function of the limit law for a super-critical branching process is one good example of this (see, for example, Grimmett & Stirzaker (1992), Chapter 5). A closely-related example in

[2] We talk of spectrally negative Lévy processes to fix our ideas, but of course all that we say applies equally well to spectrally positive Lévy processes.

[3] by a simple strong Markov and lack-of-memory argument - see, for example, Bertoin (1996), p. 190.

queueing theory is the moment generating function of the busy period distribution of an $M/G/1$ queue (see, for example, Grimmett & Stirzaker (1992), Chapter 11). Another example (which is the topic of this paper) is the computation of the first-passage-distribution of a spectrally negative Lévy process; to apply the method, we have to evaluate the Laplace transform at a general complex argument, but the Laplace transform is expressed in terms of the inverse of an analytic function, namely, the Lévy exponent. Only rarely can one find an explicit expression for this inverse function. The solution here is to use Cauchy's Theorem; we integrate up another contour, not the straight line $a + i\mathbb{R}$. Provided we can show that the contribution from the 'ends' of the contour goes to zero, the answer is the same. By making the contour the image of some other nice contour under the Lévy exponent, we do not need to find the inverse. Alternatively, it would be possible to proceed by numerically computing the inverse; Abate & Whitt (1992b) show some situations where this can be done. The need to perform the numerical computations detailed in this article arose in a specific context in credit risk modelling; see Hilberink & Rogers (2000).

The method is very easy to implement; in Section 5 the Matlab code for the calculation of the values is given. Since the calculation can be expressed in matrix terms, the efficient matrix-handling routines in Matlab make the computation extremely rapid.

2 The problem.

We are going to study a spectrally-negative Lévy process X, characterised by

$$E \exp(zX_t) = \exp(t\psi(z)), \tag{2.1}$$

where the Lévy exponent ψ has the (Lévy-Khintchin) representation

$$\psi(z) = \frac{1}{2}\sigma^2 z^2 + bz + \int_{-\infty}^{0} \{e^{zx} - 1 + z(|x| \wedge 1)\}\, \nu(dx), \tag{2.2}$$

where the measure ν satisfies the integrability condition

$$\int_{-\infty}^{0} (|x|^2 \wedge 1)\, \nu(dx) < \infty;$$

see, for example, Bertoin (1996), Chapter VII, or Bingham (1975). We shall use the notation

$$\psi_0(z) = \frac{1}{2}\sigma^2 z^2 + bz,$$

and shall throughout assume that σ *is positive*; as so often with general Lévy processes, the case where $\sigma = 0$ and the process is infinite-variation is the most difficult. We will need at a later stage of the argument that

$$\frac{\psi(z) - \psi_0(z)}{\psi_0(z)} \to 0, \tag{2.3}$$

and this works when $\sigma > 0$, but is not true if $\sigma = 0$, as the case of an asymmetric Cauchy process demonstrates. Introducing the notation

$$\bar{X}_t \equiv \sup_{u \le t} X_u, \qquad \underline{X}_t \equiv \inf_{u \le t} X_u,$$

and letting T_λ denote a random variable with exponential(λ) distribution, independent of X, we have the celebrated *Wiener-Hopf factorisation* of the Lévy process X:

$$\begin{aligned} E\exp(\, zX(T_\lambda)\,) &= \frac{\lambda}{\lambda - \psi(z)} \\ &= \psi_\lambda^+(z).\psi_\lambda^-(z) \\ &\equiv E\exp(\, z\bar{X}(T_\lambda)\,).E\exp(\, z\underline{X}(T_\lambda)\,). \end{aligned} \tag{2.4}$$

See, for example, Bertoin (1996), p. 165, Bingham (1975), Theorem 1, or Rogers & Williams (2000), I.29. In the case of a spectrally negative Lévy process, *the law of* $\bar{X}(T_\lambda)$ *is exponential*; see, for example, Bertoin (1996) for an explanation. Thus we know that for some $\beta^* \equiv \beta^*(\lambda)$ we shall have

$$\psi_\lambda^+(z) \equiv E^0\left[e^{z\bar{X}(T_\lambda)}\right] = \frac{\beta^*}{\beta^* - z}. \tag{2.5}$$

Thus

$$\psi_\lambda^-(z) = \frac{\lambda}{\lambda - \psi(z)}.\frac{\beta^* - z}{\beta^*}. \tag{2.6}$$

Since this function is analytic in the right half-plane, it must be that the apparent pole which occurs when $\lambda = \psi(z)$ is in fact cancelled out by a zero in the numerator, which is to say that the unknown value of $\beta^* = \beta^*(\lambda)$ must be the solution to

$$\psi(\beta) = \lambda. \tag{2.7}$$

The conclusion then is that we may find the Wiener-Hopf factor $\psi_\lambda^-(z)$ explicitly in this situation, at least up to solution of the equation (2.7). This gives us

$$\begin{aligned} \psi_\lambda^-(z) &= \int_0^\infty \lambda e^{-\lambda t} dt \int_{-\infty}^0 z e^{zx} P(\underline{X}_t > x) dx \\ &= \frac{\lambda}{\lambda - \psi(z)}.\frac{\beta^* - z}{\beta^*} \end{aligned}$$

Writing $H_x \equiv \inf\{t : X_t < x\}$, and setting

$$f(t,x) \equiv P(H_{-x} > t) = P(\underline{X}_t > -x),$$

our problem is to invert the transform

$$\begin{aligned} \tilde{f}(\lambda, z) &\equiv \int_0^\infty \int_0^\infty e^{-\lambda t - zx} f(t,x)\, dtdx \\ &= \frac{\beta^*(\lambda) - z}{(\lambda - \psi(z))\beta^*(\lambda) z} \end{aligned} \tag{2.8}$$

The Laplace transform $\tilde{f}$ is analytic in the region where both arguments have strictly positive real parts, and the standard Fourier transform inversion method would give us the unknown function f by Fourier integration:

$$f(t,x) = \int_{\Gamma_1} \frac{d\lambda}{2\pi i} \int_{\Gamma_2} \frac{dz}{2\pi i}\, e^{t\lambda + xz} \tilde{f}(\lambda, z)\ ; \tag{2.9}$$

it is this that we shall approximate.

3 The solution.

The difficulty in implementing the Fourier inversion recipe (2.9) is in evaluating the function β^* at a general complex argument. We sidestep this difficulty by integrating up the contour Γ_0 instead of Γ_1, where

$$\Gamma_0 = (\psi \circ \psi_0^{-1})(\Gamma_1).$$

The rationale for this choice is that for large z, $\psi(z)$ and $\psi_0(z)$ are similar in the sense of (2.3), so the contour Γ_0 will be close to Γ_1. Are they close enough that the integral up Γ_0 agrees with the integral up Γ_1 in (2.9)? Inspection of (2.8) reveals that for each fixed z,

$$\lambda \tilde{f}(\lambda, z) \to 1$$

as $|\lambda| \to \infty$. Thus if we integrate up Γ_1 from $a - iN$ to $a + iN$ then cross over to Γ_0 and integrate down from $(\psi \circ \psi_0^{-1})(a + iN)$ to $(\psi \circ \psi_0^{-1})(a - iN)$ and then cross back to $a - iN$, we can bound the contribution from the little crossings. Indeed,

$$\left| \frac{a + iN - (\psi \circ \psi_0^{-1})(a + iN)}{a + iN} \right| = \left| \frac{\psi_0(z_n) - \psi(z_n)}{\psi_0(z_n)} \right|$$

where $z_n = \psi_0^{-1}(a + iN)$, which tends to zero as $N \to \infty$ by (2.3).

The function ψ_0^{-1} is available in closed form:

$$\psi_0^{-1}(z) = \frac{\sqrt{b^2 + 2\sigma^2 z} - b}{\sigma^2},$$

where we take the function $\sqrt{\cdot}$ to be real-positive on $\mathbb{R}^+$, with a cut along $(-\infty, 0]$. If we now abbreviate $g \equiv \psi \circ \psi_0^{-1}$, the inversion formula (2.9) becomes

$$\begin{aligned} f(t,x) &= \int_{\Gamma_1} \frac{d\zeta}{2\pi i} \int_{\Gamma_2} \frac{dz}{2\pi i}\, g'(\zeta) \exp\{tg(\zeta) + xz\} \tilde{f}(g(\zeta), z) \\ &= \int_{\Gamma_1} \frac{d\zeta}{2\pi i(\sigma^2 \psi_0^{-1}(\zeta) + b)} \int_{\Gamma_2} \frac{dz}{2\pi i}\, \psi'(\psi_0^{-1}(\zeta)) \exp\{tg(\zeta) + xz\} \tilde{f}(g(\zeta), z) \end{aligned}$$

There is now no problem in evaluating the integrand in the inversion formula, because

$$\tilde{f}(g(\zeta), z) = \frac{\psi_0^{-1}(\zeta) - z}{(g(\zeta) - \psi(z))\psi_0^{-1}(\zeta) z}$$

Fixing $t > 0$ and $x > 0$, we approximate the double integral for $f(t,x)$ by the double sum

$$\begin{aligned} S_N \equiv \frac{h_1 h_2}{4\pi^2} \sum_{n=-N}^{N} \sum_{m=-N}^{N} & g'(a_1 + inh_1) \tilde{f}(g(a_1 + inh_1), a_2 + imh_2) \\ & \times \exp\{tg(a_1 + inh_1) + x(a_2 + imh_2)\}, \end{aligned}$$

and then (following Choudhury, Lucantoni & Whitt (1994)) we take $a_1 = A_1/(2tl_1)$, $a_2 = A_2/(2xl_2)$, and $h_1 = \pi/(tl_1)$, $h_2 = \pi/(xl_2)$, where l_1 and l_2 are positive integers, and A_1 and A_2 are positive reals chosen large enough to control the aliasing error, as explained in Choudhury, Lucantoni & Whitt (1994). For the calculations used as test cases in this article, it seemed that taking $A_1 = A_2 = 22.0$, and $l_1 = l_2 = 1$ gave satisfactory results. The raw value S_N may not be a very good approximation, but by using Euler summation (as advocated by Simon, Stroot & Weiss (1972)) to smooth the values of the (nearly) alternating sums, we were able to obtain good accuracy. The approximation to $f(t,x)$ finally is

$$f(t,x) \doteqdot \sum_{k=0}^{M} 2^{-M} \binom{M}{k} S_{N+k}.$$

4 Some numerical values

We took three examples to test the method, and the answers are quoted to 4 decimal places. Higher accuracy could be obtained at the expense of time, but

certainly 'basis point accuracy' (that is, one percent of one percent) is good enough in the finance industry.

(i) We took the continuous example $X_t = \sigma W_t + bt$, which is the only case where the answer is known in closed form; indeed,

$$P(H_x \le t) = \Phi(\frac{a - ct}{\sqrt{t}}) + e^{2ca}\Phi(\frac{a + ct}{\sqrt{t}}) \tag{4.1}$$

for $x < 0$, where Φ is the $N(0,1)$ distribution function, and $a \equiv x/\sigma$, $c \equiv b/\sigma$. This can be confirmed by differentiating with respect to t, to recover the well-known first passage density for a drifting Brownian motion - see, for example, Borodin & Salminen (1996), formula 2.2.0.2; The parameter values here were $\sigma = 0.2$, $b = -0.015$, and the results are displayed in Table I. The columns are indexed by the x values, and the rows indexed by the t values. We used $N = 6$ and $M = 9$ in the Euler summation, with aliasing parameters $A_1 = A_2 = 22$. The exact values computed from (4.1) agreed everywhere except occasionally in the final figure. In the middle of the afternoon on a not particularly fast multi-user server, the entire table of 120 entries computed in 62 seconds.

Table I: $P(H_{-x} > t)$ in the Brownian case

	0.01	0.02	0.03	0.04	0.05	0.06	.0.7	0.08	0.09	0.10
0.2	0.0856	0.1708	0.2544	0.3354	0.4130	0.4863	0.5547	0.6177	0.6750	0.7265
0.4	0.0595	0.1191	0.1784	0.2370	0.2944	0.3505	0.4048	0.4570	0.5071	0.5546
0.6	0.0480	0.0961	0.1441	0.1919	0.2392	0.2858	0.3316	0.3763	0.4198	0.4619
0.8	0.0411	0.0823	0.1236	0.1648	0.2057	0.2463	0.2863	0.3257	0.3644	0.4022
1.0	0.0364	0.0729	0.1095	0.1461	0.1826	0.2189	0.2548	0.2903	0.3254	0.3599
1.2	0.0329	0.0660	0.0991	0.1323	0.1654	0.1984	0.2313	0.2638	0.2960	0.3278
1.4	0.0302	0.0606	0.0910	0.1216	0.1521	0.1825	0.2128	0.2429	0.2728	0.3024
1.6	0.0280	0.0562	0.0845	0.1129	0.1412	0.1696	0.1979	0.2260	0.2540	0.2817
1.8	0.0262	0.0526	0.0791	0.1057	0.1323	0.1589	0.1854	0.2119	0.2382	0.2644
2.0	0.0247	0.0495	0.0745	0.0996	0.1247	0.1498	0.1749	0.1999	0.2248	0.2496
2.2	0.0234	0.0469	0.0706	0.0943	0.1181	0.1419	0.1657	0.1895	0.2133	0.2369
2.4	0.0222	0.0446	0.0671	0.0897	0.1124	0.1351	0.1578	0.1805	0.2031	0.2257

(ii) The second example used jumps which had an exponential distribution with parameter $c = 4$, arriving at rate $a = 2$. The other parameters of the problem were $\sigma = 0.2$ and $b = -0.015$ as in the first example.

Table II: $P(H_{-x} > t)$ in the exponential jumps case

	0.01	0.02	0.03	0.04	0.05	0.06	.0.7	0.08	0.09	0.10
0.2	0.0618	0.1235	0.1843	0.2436	0.3009	0.3556	0.4072	0.4556	0.5003	0.5414
0.4	0.0325	0.0651	0.0978	0.1303	0.1626	0.1945	0.2258	0.2566	0.2866	0.3157
0.6	0.0202	0.0406	0.0611	0.0817	0.1023	0.1229	0.1435	0.1641	0.1844	0.2047
0.8	0.0136	0.0273	0.0412	0.0551	0.0691	0.0833	0.0975	0.1118	0.1262	0.1406
1.0	0.0096	0.0192	0.0290	0.0388	0.0488	0.0588	0.0690	0.0793	0.0897	0.1002
1.2	0.0069	0.0139	0.0210	0.0281	0.0354	0.0427	0.0502	0.0578	0.0654	0.0732
1.4	0.0051	0.0103	0.0155	0.0208	0.0262	0.0317	0.0373	0.0429	0.0487	0.0546
1.6	0.0039	0.0077	0.0117	0.0157	0.0197	0.0239	0.0281	0.0324	0.0368	0.0413
1.8	0.0029	0.0059	0.0089	0.0120	0.0151	0.0182	0.0215	0.0248	0.0281	0.0316
2.0	0.0023	0.0045	0.0069	0.0092	0.0116	0.0141	0.0166	0.0191	0.0217	0.0244
2.2	0.0018	0.0035	0.0053	0.0072	0.0090	0.0109	0.0129	0.0149	0.0169	0.0190
2.4	0.0014	0.0028	0.0042	0.0056	0.0071	0.0086	0.0101	0.0117	0.0133	0.0149

In contrast to the first example, there is no independent check of the numerical values, but it is easy enough to run the calculations again with higher values of N and M as a check. Repeating the calculations with $N = 12$ and $M = 15$ produced differences of at most 4×10^{-5} in the table; this changes the final digit in a few places.

(iii) In the third example, the jumps were uniformly distributed on the interval $[-0.05, 0]$, coming at rate 2. Once again, $\sigma = 0.2$ and $b = -0.015$. We computed the values in Table III using $N = 6$ and $M = 9$, then recomputed them using $N = 12$ and $M = 15$. The differences in the answers were never bigger than 3.051×10^{-5}, so as with the second example, there were just a few changes in the final figure of some entries.

5 Matlab code

We print here the Matlab code used to generate the results of the previous section. Matlab was chosen because it has the speed of compiled C, but the ease of use of a package. It handles complex numbers with no difficulty, and the matrix/vector calculations are done extremely efficiently. Also, the code is very compact, as the specimen program below demonstrates.

```
global sigma a b c
```

Table III: $P(H_{-x} > t)$ in the uniform jumps case

	0.01	0.02	0.03	0.04	0.05	0.06	.0.7	0.08	0.09	0.10
0.2	0.0759	0.1519	0.2274	0.3018	0.3744	0.4443	0.5107	0.5731	0.6310	0.6840
0.4	0.0496	0.0996	0.1500	0.2006	0.2513	0.3017	0.3515	0.4003	0.4479	0.4940
0.6	0.0380	0.0765	0.1154	0.1548	0.1946	0.2346	0.2746	0.3144	0.3539	0.3930
0.8	0.0312	0.0628	0.0949	0.1275	0.1605	0.1939	0.2275	0.2613	0.2950	0.3287
1.0	0.0266	0.0536	0.0810	0.1089	0.1372	0.1660	0.1950	0.2244	0.2539	0.2835
1.2	0.0233	0.0468	0.0708	0.0952	0.1201	0.1454	0.1710	0.1970	0.2232	0.2496
1.4	0.0207	0.0416	0.0629	0.0847	0.1068	0.1294	0.1524	0.1757	0.1993	0.2230
1.6	0.0186	0.0374	0.0566	0.0762	0.0963	0.1167	0.1375	0.1586	0.1800	0.2016
1.8	0.0169	0.0340	0.0515	0.0693	0.0876	0.1062	0.1251	0.1444	0.1640	0.1839
2.0	0.0155	0.0312	0.0472	0.0635	0.0803	0.0973	0.1148	0.1325	0.1506	0.1689
2.2	0.0143	0.0287	0.0435	0.0586	0.0740	0.0898	0.1059	0.1223	0.1391	0.1560
2.4	0.0132	0.0266	0.0403	0.0543	0.0686	0.0833	0.0983	0.1135	0.1291	0.1449

```
sigma=0.2;c=0.05;a=2.0;b=-0.015;

% Now some parameters of the discretization - dx is the
% spacing of the x values, dt is the spacing of the t values,
% and nx, nt are the numbers of such values to be found .
% The parameters l1 and l2 are as in the Ann Appl Prob 4 article
% of Choudhury, Lucantoni & Whitt

dt=0.2;dx=0.01;nx=10;nt=12;l1=1;l2=1;
N=6;M=9;A1=22.0;A2=22.0;
X=dx:dx:nx*dx;T=dt:dt:nt*dt;

% Next we create the weights to be used in the
% Euler summation of the partial sums:

mx=pascal(M+1);my=fliplr(mx);bn=diag(my)*2^(-M);
weight=ones([2*N+1 1]);head=cumsum(bn);tail=1-cumsum(bn);
tail(M+1,:)=[];head(M+1,:)=[];
weight=[head;weight;tail];

% Now we set the values of the arguments at which the
% transform is to be evaluated:

val1=-(N+M):(N+M);val1=(i*pi*val1+A1/2)/l1;
```

```
val2=-(N+M):(N+M);val2=(i*pi*val2+A2/2)/l2;
r1=ones(1,1+2*(N+M));c1=ones(1+2*(N+M),1);
T_arr=kron(val1',r1);X_arr=kron(c1,val2);
T_inv=(1./T)';X_inv=1./X;
TT=kron(T_inv,ones(1,nx));XX=kron(ones(nt,1),X_inv);
T_args=kron(TT,T_arr);X_args=kron(XX,X_arr);

% Next we evaluate the integrand at all the points where
% it needs to be found:

integrand=FF(T_args,X_args).*exp( ...
diag(kron(T,ones(1,1+2*N+2*M)) )*lev(phi_0(T_args)) ...
 + X_args*diag(kron(X,ones(1,1+2*N+2*M)) )   );

% .. and now we prepare the two matrices which will
% pre- and post-multiply integrand to give us the answer:

left=kron(diag(T_inv),weight')/(2*l1);
right=kron(diag(X_inv),weight)/(2*l2);

% and finally the answer:

sol=real(left*integrand*right)
```

This code requires the function kept in the file FF.m:

```
function f=FF(z1,z2)
global sigma a b c
z=phi_0(z1);
f1=(z-z2).*dlev(z);
f2=(lev(z)-lev(z2)).*z.*z2.*(b+sigma^2*z);
f=f1./f2;
```

which in turn requires the functions in phi_0.m:

```
function g=phi_0(z)
global sigma  a b c
g= ( sqrt( b^2+2*sigma^2*z) - b)/sigma^2;
```

and the (problem-specific) functions in lev.m:

```
function g=lev(z)
global sigma a b c
g=sigma^2*z.^2/2 + b*z + a*(1-z*c-exp(-z*c))./(c*z);
```

and in dlev.m:

```
function g=dlev(z)
global sigma a b c
g=sigma^2*z+ b +a*exp(-c*z).*(1+c*z-exp(c*z))./(c*z.^2);
```

The functions lev.m and dlev.m given above are for the final example, and the parameter settings used are for the final example.

References.

ABATE, J., and WHITT, W. (1992a), The Fourier-series method for inverting transforms of probability distributions, *Queueing Systems Theory and Applications* **10**, 5–88.

ABATE, J., and WHITT, W. (1992b), Solving probability transform functional equations for numerical inversion, *Operations Research Letters* **12**, 275–281.

ABATE, J., and WHITT, W. (1995), Numerical inversion of Laplace transforms of probability distributions, *ORSA Journal on Computing* **7**, 36–43.

ABATE, J., CHOUDHURY, G. L., and WHITT, W. (1999) An introduction to numerical transform inversion and its application to probability models, *Computational Probability*, W. Grassman (ed.), Kluwer, Boston, pp 257–323.

BERTOIN, J. (1996), *Lévy Processes*, Cambridge University Press, Cambridge.

BINGHAM, N. H. (1975), Fluctuation theory in continuous time, *Advances in Applied Probability* **7**, 705–766.

BORODIN, A. N., and SALMINEN, P. (1996), *Handbook of Brownian Motion - Facts and Formulae*, Birkhäuser, Basel.

CHOUDHURY, G. L., LUCANTONI, D. M., and WHITT, W. (1994), Multidimensional transform inversion with applications to the transient M/G/1 queue, *Annals of Applied Probability* **4**, 719–740.

DUBNER, H., and ABATE, J. (1968), Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform, *Journal of the Association of Computing Machinery* **15**, 115–123.

GRIMMETT, G. R., and STIRZAKER, D. R., (1992), *Probability and Random Processes*, 2nd edition, Oxford University Press, Oxford.

HILBERINK, B., and ROGERS, L. C. G. (2000) Optimal capital structure and endogenous default, University of Bath preprint.

HOSONO, T. (1984), Numerical inversion of Laplace transform and some applications to wave optics, *Radio Science* **16**, 1015–1019.

ROGERS, L. C. G., and WILLIAMS, D. (2000), *Diffusions, Markov Processes and Martingales, Vol. 1*, Cambridge University Press, Cambridge.

SIMON, R. M., STROOT, M. T., and WEISS, G. H. (1972), Numerical inversion of Laplace transforms with application to percentage labeled experiments, *Comput. Biomed. Res.* **6**, 596–607.