THE UNIVERSITY OF NEW SOUTH WALES

SCHOOL OF MATHEMATICS AND STATISTICS

November 2013

MATH3311/MATH5335 – Solutions

Mathematical Computing for Finance
Computational Methods for Finance

(1) TIME ALLOWED – 2 HOURS

(2) TOTAL NUMBER OF QUESTIONS – 4

(3) ANSWER **BOTH QUESTION 1 AND QUESTION 2.**
ANSWER **EITHER** QUESTION 3 **OR** QUESTION 4.
(ONLY ONE OF QUESTIONS 3 AND 4 CAN COUNT.)

(4) THE QUESTIONS ARE OF EQUAL VALUE

(5) THIS PAPER MAY BE RETAINED BY THE CANDIDATE

(6) **ONLY** CALCULATORS WITH AN AFFIXED “UNSW APPROVED” STICKER MAY BE USED

(7) **REASONS MUST BE GIVEN FOR ALL ANSWERS**

All answers must be written in ink. Except where they are expressly required pencils may only be used for drawing, sketching or graphical work.

1. i) For n by n matrices, the computational costs of some common operations are given in Table 1.1.

Operation	Flops
Matrix multiplication	$2n^3$
QR factorization	$2n^3 + O(n^2)$
LU factorization	$\frac{2n^3}{3} + O(n^2)$
Cholesky factorization	$\frac{n^3}{3} + O(n^2)$
Back/forward substitution (each)	$n^2 + O(n)$
Tridiagonal solve	$8n + O(1)$

Table 1.1: Computational costs for matrix operations

A high powered workstation has a 2.4 GHz 8 core Xeon processor, where each core can do four floating point operations per clock cycle, 16 GB RAM and 20 MB of cache memory.

A simulation requires the evaluation of the multivariate density function

$$f(\boldsymbol{x}) = \frac{\exp\bigl(-(\boldsymbol{x}-\boldsymbol{\mu})^T C^{-1}(\boldsymbol{x}-\boldsymbol{\mu})/2\bigr)}{\sqrt{(2\pi)^n \det(C)}}$$

at N vectors $\boldsymbol{x}_j \in \mathbb{R}^n, j = 1, \ldots, N$. The vectors are stored as the columns of a matrix $X \in \mathbb{R}^{n\times N}$. $C \in \mathbb{R}^{n\times n}$ is a nonsingular covariance matrix.

a) What is the largest covariance matrix C that can be stored in the **cache** memory of the workstation using IEEE double precision.
Answer: IEEE double precision uses 8 bytes per floating point number and the n by n matrix C has n^2 elements, so for the cache memory

$$8n^2 = 20 \text{ MB}.$$

Using 1 MB $= 2^{20}$ bytes,

$$8n^2 = 20 \times 2^{20} \implies n^2 = 10 \times 2^{18} \implies n = 1.61908616 \times 10^3,$$

so $n = 1,619$ or more sensibly $n \approx 1,600$.
Alternatively, using 1 Mb $= 10^6$ bytes,

$$8n^2 = 20 \times 10^6 \implies n^2 = \frac{5}{42} \times 10^6 \implies n = 1.58113883 \times 10^3,$$

so $n = 1,581$.
A covariance matrix must be symmetric ($C^T = C$ or $C_{ji} = C_{ij}$ for all $i \neq j$), so only the diagonal elements and either the strict lower or

strict upper triangle needs to be stored, requiring $n(n{+}1)/2$ elements. Using 1 MB = 2^{20} bytes, this gives

$$8 \times \frac{n(n+1)}{2} = 20 \times 2^{20} \implies n^2 \approx 5 \times 2^{20} \implies n \approx 2,289.$$

Which ever case, n must be an integer for the size of the matrix.

b) Estimate the speed of the workstation in Gflops per second.
Answer: The speed of the workstation is estimated as

$$\begin{aligned} \text{Speed} &= \text{GHz} \times 10^9 \times \text{number cores} \times \text{flops per cycle per core} \\ &= 2.4 \times 10^9 \times 8 \times 4 = 7.68 \times 10^{10} \text{ flops /sec} \\ &= 76.8 \text{ Gflops/sec.} \end{aligned}$$

GHz and Gflops both use factors of 10^9, not 2^{30} which is used for memory (Cache, RAM).

c) What is the largest value of n so that a linear system $C\boldsymbol{x} = \boldsymbol{b}$ can be solved in 1 minute on this workstation?
Answer: As C is a non-singular covariance matrix, C must be positive definite, so the major cost in solving the linear system $C\boldsymbol{x} = \boldsymbol{b}$ is the Cholesky factorization requiring $\frac{n^3}{3}$ flops. Thus in 1 minute = 60 seconds,

$$\frac{n^3}{3} = 76.8 \times 10^9 \times 60 \implies n^3 = 1.3824 \times 10^{13} \implies n \approx 24,000.$$

d) **Analyst 1** claims that the simulation will take $2Nn^3/3$ flops while **Analyst 2** claims the simulation will take $n^3/3 + O(Nn^2)$ flops. Explain why you agree with one or neither of the analysts.
Answer: The key observation is that the matrix C does not change with each different point $\boldsymbol{x}_j$, so the Cholesky factorization with a cost of $\frac{n^3}{3}$ flops should only be calculated once. Thus **Analyst 1** is incorrect. **Analyst 1** has also incorrectly used the LU factorization requiring $\frac{2n^3}{3}$ flops, rather than exploiting the structure of the covariance matrix.
For each of the N points $\boldsymbol{x}_j, j = 1, \ldots, N$, evaluating $f(\boldsymbol{x}_j)$ can be done in $O(n^2)$ flops, once the Cholesky factorisation has been calculated. Thus **Analyst 2** is correct.

e) Explain how you would efficiently calculate $f(\boldsymbol{x}_j)$, $j = 1, \ldots, N$.
Answer: The key steps in evaluating $f(\boldsymbol{x}_j)$ for $j = 1, \ldots, N$ are

- Calculate the Cholesky factorization $C = R^T R$, where R is upper triangular, once.
- Calculate $c_0 = \sqrt{\det(C)} = \sqrt{\det(R^T)\det(R)} = \det(R) = \prod_{j=1}^n R_{jj}$
- Calculate $c_1 = (2\pi)^{-n/2}/c_0$.
- For each $j = 1, \ldots, N$

- Solve $R^T\boldsymbol{y} = (\boldsymbol{x}_j - \boldsymbol{\mu})$ by forward substitution.
- Calculate $f(\boldsymbol{x}_j) = c_1 \exp(-\boldsymbol{y}^T\boldsymbol{y}/2)$, where $\boldsymbol{y}^T\boldsymbol{y} = \sum_{i=1}^n y_i^2$.

Only the last two steps are repeated for each value of $j = 1, \ldots, N$. As $C^{-1} = R^{-1}R^{-T}$,

$$\boldsymbol{y}^T\boldsymbol{y} = (\boldsymbol{x}_j - \boldsymbol{\mu})^T R^{-1}R^{-T}(\boldsymbol{x}_j - \boldsymbol{\mu}) = (\boldsymbol{x}_j - \boldsymbol{\mu})^T C^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}).$$

ii) Let C be an n by n covariance matrix. A Matlab session gives the following output:

```
[R, posdef] = chol(C);
posdef
ans =
  0
Cdet = det(C)
Cdet =
  0
rc = rcond(C)
rc =
  2.0128e-10
```

a) Is the matrix C nonsingular?
Answer: The matrix C is not nonsingular as $\det(C) = 0$, indicating it is singular.

b) Is the matrix C positive definite?
Answer: The second output argument for the Matlab function `chol` is 0 if and only if the matrix is positive definite. Thus as `posdef` has the value 0, C is positive definite. (If C is not positive definite this argument has the index in $\{1, \ldots, n\}$ when the Cholesky factorization failed).

c) Are your answers to parts a) and b) consistent? If not, explain any inconsistencies.
Answer: A positive definite matrix has all eigenvalues $\lambda_i > 0$, $i = 1, \ldots, n$, so $\det(C) = \prod_{i=1}^n \lambda_i > 0$. Thus C being positive definite contradicts the results that $\det(C) = 0$, so they are not consistent.
If n is large, the product of a large number of small values (but still larger than the machine precision ϵ), may underflow to zero. Matlab uses the Cholesky (or LU when the matrix is not symmetric positive definite) factorization to calculate the determinant, so

$$\det(C) = \det(R)^2 = \left(\prod_{i=1}^n R_{ii}\right)^2 < \texttt{realmin} \approx 2 \times 10^{-308},$$

even though $R_{ii} > \epsilon = 2.2 \times 10^{-16}$ for all $i = 1, \ldots, n$.

d) What does the output tell you about the condition number $\kappa(C)$?

Please see over ...

Answer: The MATLAB function `rcond` gives an estimate of the reciprocal of the 1-norm condition number, so

$$\kappa_1(C) = \frac{1}{\texttt{rcond(C)}} = \frac{1}{2.0128 \times 10^{-10}} \approx 5 \times 10^9.$$

Such a large condition number (on a scale of $1 = 10^0$ to $1/\epsilon \approx 5\times10^{15}$) indicates the problem is ill-conditioned.
The condition number must satisfy $\kappa \geq 1$. An answer $\kappa = 4968203497$ is not really sensible, as it contains many more significant figures that the data!

e) The elements of C are known "exactly", and the computed solution $\boldsymbol{x}$ to $C\boldsymbol{x} = \boldsymbol{b}$ must have at least 4 significant figures. How accurate must $\boldsymbol{b}$ be to achieve this?
Answer: An estimate of the relative error $\mathrm{re}(\boldsymbol{x})$ in the computed solution to $C\boldsymbol{x} = \boldsymbol{b}$ is

$$\mathrm{re}(\boldsymbol{x}) \leq \kappa(C) \left(\mathrm{re}(C) + \mathrm{re}(\boldsymbol{b})\right).$$

The same norm should be used to measure the condition number and all the relative errors.
The computed solution has at least 4 significant figures if

$$\kappa(C) \left(\mathrm{re}(C) + \mathrm{re}(\boldsymbol{b})\right) \leq 0.5 \times 10^{-4}.$$

As C is known exactly, $\mathrm{re}(C) = \epsilon = 2.2 \times 10^{-16}$, so

$$\mathrm{re}(\boldsymbol{b}) \leq \frac{0.5 \times 10^{-4}}{\kappa(C)} - \epsilon = \frac{0.5 \times 10^{-4}}{5 \times 10^9} - 2 \times 10^{-16} \approx 10^{-14}.$$

The required relative error in $\boldsymbol{b}$ is $10^{-14} < 0.5 \times 10^{-13}$ so $\boldsymbol{b}$ should have elements with at least 13 significant figures. This may (just) be possible when using double precision floating point arithmetic, but is not possible if only single precision arithmetic is used.

iii) Consider the matrix

$$B = C + \sum_{k=1}^{K} \alpha_k \boldsymbol{v}_k \boldsymbol{v}_k^T,$$

where $C \in \mathbb{R}^{n\times n}$ is a sparse symmetric positive definite matrix with $n \gg K$, $\alpha_k \geq 0$ and $\boldsymbol{v}_k \in \mathbb{R}^n$ for $k = 1, \ldots, K$.

a) Show that B is positive definite.
Answer: The matrix B is positive definite $\iff$ $\boldsymbol{u}^T B\boldsymbol{u} > 0$ for all $\boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{u} \neq \boldsymbol{0}$. Here

$$\boldsymbol{u}^T B\boldsymbol{u} = \boldsymbol{u}^T \left(C + \sum_{k=1}^{K} \alpha_k \boldsymbol{v}_k \boldsymbol{v}_k^T\right) \boldsymbol{u} = \boldsymbol{u}^T C\boldsymbol{u} + \sum_{k=1}^{K} \alpha_k (\boldsymbol{u}^T \boldsymbol{v}_k)(\boldsymbol{v}_k^T \boldsymbol{u}).$$

As $\boldsymbol{u}, \boldsymbol{v}_k \in \mathbb{R}^n$, $\boldsymbol{v}_k^T \boldsymbol{u} = \boldsymbol{u}^T \boldsymbol{v}_k$ as they are both scalars, so the sum is

$$\sum_{k=1}^{K} \alpha_k (\boldsymbol{u}^T \boldsymbol{v}_k)(\boldsymbol{v}_k^T \boldsymbol{u}) = \sum_{k=1}^{K} \alpha_k (\boldsymbol{u}^T \boldsymbol{v}_k)^2 \geq 0$$

as $\alpha_k \geq 0$ for $k = 1, \ldots, K$. As C is positive definite, $\boldsymbol{u}^T C \boldsymbol{u} > 0$ for all $\boldsymbol{u} \neq \boldsymbol{0}$, so $\boldsymbol{u}^T B \boldsymbol{u} > 0$ for all $\boldsymbol{u} \neq \boldsymbol{0}$ and B is positive definite.

b) Why is explicitly forming B^{-1} not a good idea.
Answer: Explicitly forming B^{-1} is not a good idea as the sparse structure of C will be lost, and the inverse can be dense even if the matrix is sparse. Here B also contains some hidden structure in the low rank corrections in the sum which will be lost if B^{-1} is calculated. Explicitly forming B^{-1} is also computationally more expensive than solving a linear system $B\boldsymbol{x} = \boldsymbol{b}$ using an appropriate matrix factorization (in this case a Cholesky factorization as B is symmetric positive definite).

c) How can you calculate $\boldsymbol{y} = B\boldsymbol{x}$ efficiently for a given $\boldsymbol{x} \in \mathbb{R}^n$?
Answer: The question asks for the matrix-vector product (an essential step in many iterative methods for solving a linear system), not the solution of the linear system. Thus

$$\boldsymbol{y} = B\boldsymbol{x} = \left(C + \sum_{k=1}^{K} \alpha_k \boldsymbol{v}_k \boldsymbol{v}_k^T \right) \boldsymbol{x} = C\boldsymbol{x} + \sum_{k=1}^{K} (\alpha_k \boldsymbol{v}_k^T \boldsymbol{x}) \boldsymbol{v}_k.$$

The sparsity of C can be used to calculate the matrix-vector product $C\boldsymbol{x}$ very efficiently. For each k, $\alpha_k \boldsymbol{v}_k^T \boldsymbol{x} \in \mathbb{R}$ is a scalar, so the sum just consists of the inner product $\boldsymbol{v}_k^T \boldsymbol{x}$ ($2n$ flops) to find the scalar, then adding up K scalars times vectors ($3Kn$ flops), both of which are very efficient for $K \ll n$.

2. On 24 August the ASX 30 day intrabank cash rate futures implied yield curve was given by the $m = 18$ data values in Table 2.1, which are plotted in Figure 2.1. Three interns, **Intern 1**, **Intern 2** and **Intern 3**, were asked to perform various calculations using the data (t_i, r_i) where $r_i = r(t_i)$ for $i = 1, \ldots, m$ are the annualised yields as percentages.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
t_i (months)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
r_i (%)	2.545	2.485	2.450	2.380	2.325	2.325	2.305	2.295	2.285	2.290	2.330	2.380	2.425	2.475	2.530	2.600	2.685	2.695

Table 2.1: Implied yield data

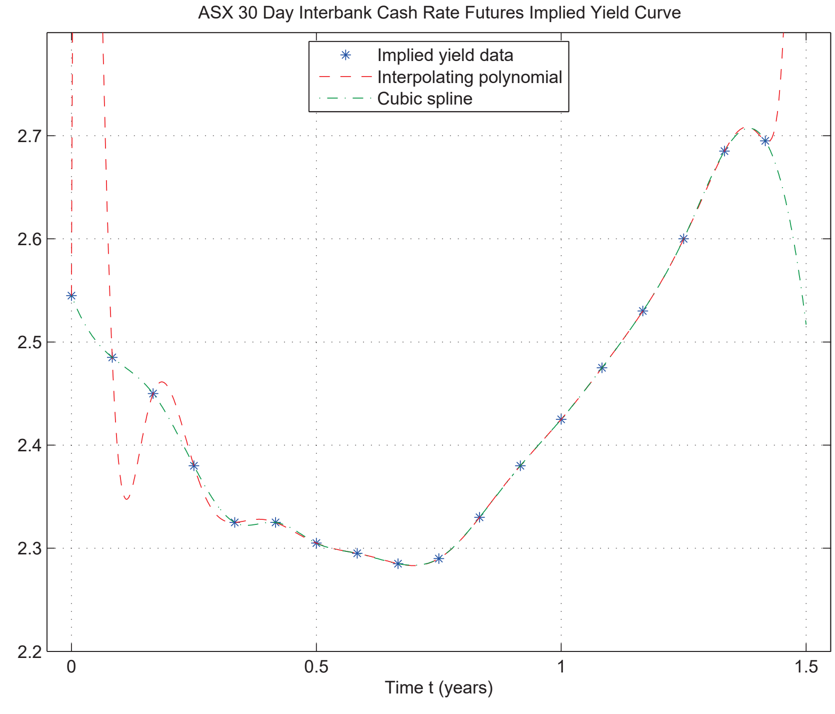

Figure 2.1: Yield curve data and approximations

The data was stored in Matlab in column vectors `tdat` and `rdat`.

i) **Intern 1** approximated the data by the lowest degree polynomial that would interpolate all the data values.

a) What is the degree n of this polynomial?

Answer: A polynomial $\sum_{k=0}^{n} a_k t^k$ of degree n has $n+1$ coefficients $a_k, k = 0, \ldots, n$. Thus a degree 17 polynomial is required to fit the $m = 18$ data values in the table.

b) **Intern 1** solved the linear system $A\boldsymbol{x} = \boldsymbol{r}$ where A is the Vandermonde matrix with elements $A_{ij} = t_i^{j-1}$. The calculated polynomial goes through all the data points in Figure 2.1, but the MATLAB commands they used gave

```
Acnd = cond(A)
Acnd =
  3.2705e+14
x = A \ rdat;
rpolchk = norm(A*tdat-rdat)
rpolchk =
  1.6801e+03
```

What is your explanation for this?

Answer: The value of `rpolchk = 1.6e+03` indicates that the 2-norm of the residuals, and hence at least some of the residuals, are very large. This contradicts the plot, where the polynomial visually goes through all the data points, so the residuals should be small. Examination of the MATLAB code shows that **intern 1** has **not** calculated the norm of the residuals `r = A*x - rdat`, but incorrectly used the vector `tdat` of data times, rather than the estimated parameters `x`.

c) What else is wrong with **Intern 1**'s approach?

Answer: The choice of a high degree polynomial, which must go to ±∞ as t gets large, and may have large oscillations (see plot for $t \in [0, 0.2]$), is not appropriate for interest rate data.

Secondly the use of the monomial basis $\{1, t, t^2, t^3, \ldots, t^{m-1}\}$ and equally spaced data produces a very ill-conditioned coefficient matrix A with $\kappa_2(A) = 3 \times 10^{14}$. Thus any errors in the data (which is only given to 4 significant figures) may be amplified when solving for the polynomial coefficients.

ii) **Intern 2** approximated the data using the MATLAB commands

```
tspl = linspace(0, 1.5, 1501);
rspl = spline(tdat, rdat, tspl);
```

obtaining the dash-dot curve in Figure 2.1.

a) Explain what an interpolating cubic spline $s(t)$ is.

Answer: An interpolating cubic spline $s(t)$ is a piecewise cubic function such that

- $s(t) = s_i(t) \in \mathbb{P}_3$ for $t \in [t_i, t_{i+1}]$, $i = 1, \ldots, m$, so s_i is a cubic (degree 3 polynomial) on each interval.

- The function $s(t)$, its first derivative $s'(t)$ and second derivative $s''(t)$ are continuous at all the interior knots t_i, $i = 2, \dots, m-1$.
- The spline interpolates the data, $s(t_i) = r_i$ for $i = 1, \dots, m$.

b) The spline approximation gives a low predicted yield at $t = 1.5$ years? Explain why this has occurred.
Answer: The interpolation and continuity conditions still leave 2 degrees of freedom in the choice of the parameters for the cubic spline. When predicting values outside the range of the data (extrapolation), for example at $t = 1.5$, the choice of these two degree of freedom can have a major affect on the approximation. Here MATLAB has used the default not-a-knot end conditions, so a single cubic is used for all $t \geq t_{16} = 15/12 = 1.25$ years, resulting in an unreasonably low prediction of the rate at $t = 1.5$.

c) How could knowledge of the slopes $r'(t_1)$ and $r'(t_{18})$ be used to improve the spline approximation.
Answer: If the slopes $r'(t_1)$ at the first knot and $r'(t_m)$ at the last knot are known, then these can be used to satisfy the two degrees of freedom in the cubic spline. This is known as the "clamped" cubic spline. This **may** help improve the extrapolated value at $t = 1.5$.

iii) The value at time t of \$1 received at time $T \geq t$ (years) is

$$B(t,T) = \exp\left(-\int_t^T r(s)ds\right) \qquad \text{for} \quad 0 \leq t \leq T.$$

a) **Intern 3** used the yield $r_9 = 2.285\%$ at 8 months and the yield $r_{13} = 2.425\%$ at 1 year to provide the bounds $0.2975 \leq B(0.5, 1) \leq 0.3190$. Calculate lower and upper bounds on $B(0.5, 1)$ using r_9 and r_{13}. Do you agree with **Intern 3**?
Answer: For $B(0.5, 1)$ we are only interested in the values of $r(s)$ for $s \in [0.5, 1]$. Over this interval

$$r_9 = 2.285 \approx \min_{s\in[0.5,1])} r(s), \qquad r_{13} = 2.425 \approx \max_{s\in[0.5,1]} r(s).$$

These bounds are only estimates, as $r(s)$ may be lower or higher in between the data points. Hence, using rectangles of width $1 - 0.5 = 0.5$ to approximate the area under the graph,

$$0.5\,\frac{2.285}{100} = 0.011425 \leq \int_{0.5}^1 r(s)\,ds \leq 0.5\,\frac{2.425}{100} = 0.012125.$$

As $B(0.5, 1) = \exp\left(-\int_{0.5}^1 r(s)\,ds\right)$, and with the negative sign reversing the inequalities,

$$\exp(-0.012125) = 0.987948 \leq B(0.5, 1) \leq \exp(-0.011425) = 0.988640.$$

Intern 3 is incorrect. They have omitted to convert the interest rate data into fractions by dividing by 100.
Reality check: There is no way 0.3 will grow to 1 in 1/2 a year with annualised interest rates of just over 2%, so the bounds must be wrong!

b) Use a quadrature rule of your choice and the data in Table 2.1 to estimate the value at $t = 0.5$ of \$1 million received at $T = 1$.
Answer: To estimate $B(0.5, 1)$ use the 7 data values $r_i, i = 7, \ldots, 13$, going from $t_7 = 6$ months to $t_{13} = 12$ months, with an interval width of $h = 1/12$ years. Using the Trapezoidal rule

$$\begin{aligned}\int_{0.5}^{1} r(s)ds &\approx \frac{h}{100}\left[\frac{2.305}{2} + 2.295 + 2.285 + 2.290 + 2.330 + 2.380 + \frac{2.425}{2}\right] \\ &= 0.01162083.\end{aligned}$$

Hence

$$B(0.5, 1) = \exp(-0.01162083) = 0.9884463.$$

The the value at $t = 0.5$ of \$1 million received at $T = 1$ is

$$B(0.5, 1) \times 10^6 = \$988,446,$$

rounding to the nearest dollar.
Alternatively, using Simpson's rule

$$\begin{aligned}\int_{0.5}^{1} r(s)ds &\approx \frac{h}{300}[2.305 + 4 \times 2.295 + 2 \times 2.285 + 4 \times 2.290 + \\ &\qquad 2 \times 2.330 + 4 \times 2.380 + 2.425] \\ &= 0.01161666.\end{aligned}$$

Hence

$$B(0.5, 1) = \exp(-0.01161666) = 0.9884505$$

so the value at $t = 0.5$ of \$1 million received at $T = 1$ is

$$B(0.5, 1) \times 10^6 = \$988,451,$$

rounding to the nearest dollar.
The values for $B(0.5, 1)$ should satisfy the correct bounds from part a).
What accuracy should the answer be given to? The interest rate data in Table 2.1 is given to the nearest 0.005%, that is with absolute error of up to 0.0025% or 0.000025 as a fraction.

c) The Gauss-Legendre nodes z_j and weights w_j for $j = 1, \ldots, N$ on the interval $[-1, 1]$ are calculated using the MATLAB command

```
[z, w] = gauleg(N)
```

A) Explain how the Gauss-Legendre nodes and weights can be used to estimate $B(0.5, 1)$.
Answer: The Gauss-Legendre nodes and weights are given for the interval $[-1, 1]$. The first step is to use a linear transformation $s = \alpha + \beta z$ to map $z \in [-1,]1$ to $s \in [0.5, 1]$. The transformation is given by the corresponding end-points:

$$\begin{aligned} z = -1 &\iff s = 0.5 \implies \alpha - \beta = 0.5 \\ z = 1 &\iff s = 1 \implies \alpha + \beta = 1 \\ &\implies \alpha = \frac{3}{4}, \quad \beta = \frac{1}{4}, \end{aligned}$$

so $s = \frac{3}{4} + \frac{1}{4}z$ and $ds = \frac{1}{4}dz$. Then

$$\int_{0.5}^{1} r(s)\, ds = \int_{-1}^{1} r\left(\frac{3}{4} + \frac{1}{4}z\right)\frac{1}{4}dz \approx Q_N^{GL} := \sum_{j=1}^{N} \frac{w_j}{4}\, r\left(\frac{3}{4} + \frac{1}{4}z_j\right).$$

Finally $B(0.5, 1) = \exp\left(-Q_N^{GL}\right)$.

B) **Intern 3** used the cubic spline and the Gauss-Legendre nodes to estimate $B(0.5, 1)$ to high accuracy, but needed a large number of nodes N. Explain why this happens.
Answer: The Gauss-Legendre rule with N nodes assumes the integrand is $2N$ times continuously differentiable, in which case the integration error reduces rapidly with N. However a cubic spline is only twice continuously differentiable, with discontinuities in the third derivatives at the knot points. Hence a large number of points were need to achieve high accuracy.

d) How can you estimate $B(0.5, 1)$ accurately using the cubic spline and Simpson's rule?
Answer: Simpson's rule has degree of precision 3, meaning that it is exact for cubic polynomials. The spline is a piecewise cubic polynomial. It is not a cubic over the whole interval $[0.5, 1]$, but just on each subinterval $[t_i, t_{i+1}]$. Simpson's rule requires an even number of intervals, so we need to add the midpoint to each sub-interval, get its value using the cubic spline and then integrate each cubic piece using Simpson's rule. This will give the exact integral of the cubic spline as

$$\int_{0.5}^{1} r(s)ds = \sum_{i=7}^{12} \int_{t_i}^{t_{i+1}} r(s)\, ds = \sum_{i=7}^{12} \frac{1}{72}\left(r_i + 4r_{i+\frac{1}{2}} + r_{i+1}\right),$$

where r_i, r_{i+1} are data values in the table and $r_{i+\frac{1}{2}}$ is the value of $r(s)$ at the midpoint $(t_i + t_{i+1})/2$, evaluated using the cubic spline, and $h = 1/24$ is the interval width after adding a midpoint to subdivide each monthly interval $[t_i, t_{i+1}]$.

ANSWER EITHER QUESTION 3 OR QUESTION 4, NOT BOTH

3. The value $V(S,t)$ of an option on an underlying asset with price S at time t satisfies the Black–Scholes PDE

$$\frac{\partial V}{\partial t} + rS\,\frac{\partial V}{\partial S} + \frac{\sigma^2}{2}S^2\,\frac{\partial^2 V}{\partial S^2} = rV \tag{3.1}$$

for $0 < S < \infty$ and $0 < t < T$. Assume the risk-free interest rate r and the volatility σ are known positive constants.

At expiry T, the payoff of an "exotic" option with strikes $0 < K_1 < K_2$ is

$$f(S) = \begin{cases} 0 & \text{if } S < K_1; \\ S - K_1 & \text{if } K_1 \le S \le K_2; \\ K_2 - K_1 & \text{if } S > K_2. \end{cases}$$

i) Carefully explain what the "initial" conditions are for this problem.

Answer: The "initial" conditions are at expiry $t = T$ where the value of the option is defined to be the payoff, so

$$V(S,T) = f(S) = \begin{cases} 0 & \text{if } S < K_1; \\ S - K_1 & \text{if } K_1 \le S \le K_2; \\ K_2 - K_1 & \text{if } S > K_2. \end{cases}$$

ii) The change of variables to log-moneyness y and time to expiry τ is

$$y = \log(S/X) \qquad \text{and} \qquad \tau = T - t,$$

where X is a positive constant. This gives $W(y,\tau) = V(S,t)$.

a) Find expressions for

$$\frac{\partial W}{\partial \tau}, \qquad \frac{\partial W}{\partial y}, \qquad \frac{\partial^2 W}{\partial y^2},$$

in terms of the partial derivatives of $V(S,t)$.

Answer: The new variables are

$$y = \log(S/X) \iff S = Xe^y \implies \frac{dS}{dy} = Xe^y = S,$$

$$\tau = T - t \iff t = T - \tau \implies \frac{dt}{d\tau} = -1.$$

Using these relations with $W(y,\tau) = V(S,t)$ and the chain rule,

$$\begin{aligned}
\frac{\partial W}{\partial \tau} &= \frac{\partial V}{\partial t} \times \frac{dt}{d\tau} = -\frac{\partial V}{\partial t}, \\
\frac{\partial W}{\partial y} &= \frac{\partial V}{\partial S} \times \frac{dS}{dy} = S\frac{\partial V}{\partial S}, \\
\frac{\partial^2 W}{\partial y^2} &= \frac{\partial}{\partial y}\frac{\partial W}{\partial y} = \frac{\partial}{\partial S}\left(S\frac{\partial V}{\partial S}\right) \times \frac{dS}{dy} \\
&= \left(\frac{\partial V}{\partial S} + S\frac{\partial^2 V}{\partial S^2}\right) S = S\frac{\partial V}{\partial S} + S^2\frac{\partial^2 V}{\partial S^2}.
\end{aligned}$$

b) Show that the Black and Scholes PDE (3.1) is equivalent to

$$-\frac{\partial W}{\partial \tau} + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial W}{\partial y} + \frac{\sigma^2}{2}\frac{\partial^2 W}{\partial y^2} = rW \qquad (3.2)$$

in terms of the log-moneyness y and the time to expiry τ.
Answer: The expressions in part a) for the partial derivatives give

$$\frac{\partial V}{\partial t} = -\frac{\partial W}{\partial \tau}, \qquad S\frac{\partial V}{\partial S} = \frac{\partial W}{\partial y}, \qquad S^2\frac{\partial^2 W}{\partial S^2} = \frac{\partial^2 W}{\partial y^2} - \frac{\partial W}{\partial y}.$$

Substituting in the Black-Scholes PDE (3.1) gives

$$-\frac{\partial W}{\partial \tau} + r\frac{\partial W}{\partial y} + \frac{\sigma^2}{2}\left(\frac{\partial^2 W}{\partial y^2} - \frac{\partial W}{\partial y}\right) = rW.$$

Collecting the terms in the derivatives gives

$$-\frac{\partial W}{\partial \tau} + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial W}{\partial y} + \frac{\sigma^2}{2}\frac{\partial^2 W}{\partial y^2} = rW$$

as required.

c) What are the initial conditions for the transformed problem?
Answer: The initial conditions for the transformed problem are at $\tau = 0$, (as option expiry at $t = T \iff \tau = T - t = 0$)

$$\begin{aligned} W(y,0) = f(Xe^y) &= \begin{cases} 0 & \text{if } Xe^y < K_1; \\ Xe^y - K_1 & \text{if } K_1 \le Xe^y \le K_2; \\ K_2 - K_1 & \text{if } Xe^y > K_2, \end{cases} \\ &= \begin{cases} 0 & \text{if } y < \log(K_1/X); \\ Xe^y - K_1 & \text{if } \log(K_1/X) \le y \le \log(K_2/X); \\ K_2 - K_1 & \text{if } y > \log(K_2/X). \end{cases} \end{aligned}$$

iii) Let $y_{\max} > 0$ and consider the domain $y \in [-y_{\max}, y_{\max}]$ and $\tau \in [0, T]$. Consider the grid (y_j, τ_ℓ) where

$$y_j = -y_{\max} + j\Delta y,\ j = 0,1,2,\ldots,2(n+1), \quad \tau_\ell = \ell\Delta\tau,\ \ell = 0,1,2,\ldots,m,$$

and

$$\Delta y = \frac{y_{\max}}{n+1} \qquad \text{and} \qquad \Delta\tau = \frac{T}{m}.$$

Let $W_j^\ell \approx W(y_j, \tau_\ell)$ denote the approximate solution.

a) At the time to expiry τ_ℓ and log-moneyness y_j give

A) central difference approximations of $O((\Delta y)^2)$ for the partial derivatives

$$\frac{\partial W}{\partial y} \quad \text{and} \quad \frac{\partial^2 W}{\partial y^2}.$$

Please see over . . .

B) forward difference approximation of $O(\Delta\tau)$ for the partial derivative

$$\frac{\partial W}{\partial \tau}.$$

Answer: At time to expiry τ_ℓ and log-moneyness y_j, finite difference approximations of the derivatives are

$$\begin{aligned}
\left.\frac{\partial W}{\partial y}\right|_{\tau_\ell, y_j} &= \frac{W_{j+1}^\ell - W_{j-1}^\ell}{2\Delta y} + O((\Delta y)^2), \\
\left.\frac{\partial^2 W}{\partial y^2}\right|_{\tau_\ell, y_j} &= \frac{W_{j+1}^\ell - 2W_j^\ell + W_{j-1}^\ell}{(\Delta y)^2} + O((\Delta y)^2), \\
\left.\frac{\partial W}{\partial \tau}\right|_{\tau_\ell, y_j} &= \frac{W_j^{\ell+1} - W_j^\ell}{\Delta \tau} + O(\Delta\tau).
\end{aligned}$$

b) Show that the finite difference equation for the PDE (3.2) has the form

$$W_j^{\ell+1} = \alpha W_{j-1}^\ell + \beta W_j^\ell + \gamma W_{j+1}^\ell, \tag{3.3}$$

and find the coefficients α, β and γ.

Answer: Ignoring the $O((\Delta y)^2)$ and $O(\Delta\tau)$ terms in the finite difference approximations and substituting in the transformed PDE (3.2)

$$-\left(\frac{W_j^{\ell+1} - W_j^\ell}{\Delta\tau}\right) + \left(r - \frac{\sigma^2}{2}\right)\left(\frac{W_{j+1}^\ell - W_{j-1}^\ell}{2\Delta y}\right) + \frac{\sigma^2}{2}\left(\frac{W_{j+1}^\ell - 2W_j^\ell + W_{j-1}^\ell}{(\Delta y)^2}\right) = rW_j^\ell.$$

Multiplying through by $-\Delta\tau$ and collecting terms gives

$$\begin{aligned}
W_j^{\ell+1} &= W_{j-1}^\ell \Delta\tau \left(-\frac{\left(r - \frac{\sigma^2}{2}\right)}{2\Delta y} + \frac{\sigma^2}{2(\Delta y)^2}\right) + \\
&\quad W_j^\ell \left(1 - r\Delta\tau - \frac{\sigma^2 \Delta\tau}{(\Delta y)^2}\right) + \\
&\quad W_{j+1}^\ell \Delta\tau \left(\frac{\left(r - \frac{\sigma^2}{2}\right)}{2\Delta y} + \frac{\sigma^2}{2(\Delta y)^2}\right).
\end{aligned}$$

Hence

$$\begin{aligned}
\alpha &= \frac{\sigma^2 \Delta\tau}{2(\Delta y)^2} - \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{2\Delta y}, \\
\beta &= 1 - r\Delta\tau - \frac{\sigma^2\Delta\tau}{(\Delta y)^2}, \\
\gamma &= \frac{\sigma^2 \Delta\tau}{2(\Delta y)^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{2\Delta y}.
\end{aligned}$$

Please see over ...

c) A student claims that as the initial conditions are at expiry and the time stepping is from expiry at $t_m = T$ back to $t_0 = 0$, equation (3.3) represents an implicit method so there are no restrictions on the values for $\Delta\tau$ and Δy that can be used. Is this correct? Justify your answer.
Answer: After the transformation, the initial conditions are at $\tau = 0$, so the method time steps forward from $\tau_0 = 0$ to $\tau_m = T$. Thus the values at time step ℓ are known and (3.3) is an explicit formula for the unknown values at time step $\ell + 1$. An explicit method does require a stability restriction of the form

$$\frac{\Delta\tau}{(\Delta y)^2} < \rho$$

to ensure that rounding errors in the initial conditions do not get amplified until they completely destroy the numerical solution. The student is incorrect, having forgotten to take account of the change of variables to time to expiry τ.

iv) The current asset price at $t = 0$ is $S_0 = Xe^{y_{j_0}}$.

a) Give an approximation for the option Delta Δ.
Answer: The option Delta Δ is the first derivative of the option value with respect to the asset, evaluated at the current time $t = 0$ and asset value $S = S_0$. This corresponds to log-moneyness $y_0 = \log(S_0/X)$ and time to expiry $\tau = T$. In terms of the values W_j^ℓ produced by the numerical method,

$$\Delta = \left.\frac{\partial V}{\partial S}\right|_{S=S_0,t=0} = \left.\frac{1}{S}\frac{\partial W}{\partial y}\right|_{y=y_0,\tau=T} \approx \frac{1}{Xe^{y_{j_0}}}\frac{W_{j_0+1}^m - W_{j_0-1}^m}{2\Delta y}.$$

b) Give an approximation for the option Gamma Γ.
Answer: The option Gamma Γ is the second derivative of the option value with respect to the underlying asset, evaluated at the current time $t = 0$ and asset value $S = S_0$. In terms of the transformed quantities

$$\begin{aligned}
\Gamma &= \left.\frac{\partial^2 V}{\partial S^2}\right|_{S=S_0,t=0} \\
&= \left.\frac{1}{S^2}\left(\frac{\partial^2 W}{\partial y^2} - \frac{\partial W}{\partial y}\right)\right|_{y=y_0,\tau=T} \\
&\approx \frac{1}{X^2e^{2y_{j_0}}}\left(\frac{W_{j_0+1}^m - 2W_{j_0}^m + W_{j_0-1}^m}{(\Delta y)^2} - \frac{W_{j_0+1}^m - W_{j_0-1}^m}{2\Delta y}\right).
\end{aligned}$$

ANSWER EITHER QUESTION 3 OR QUESTION 4, NOT BOTH

4. i) Consider the expected value

$$\mathbb{E}[h] = \int_0^\infty h(x)p(x)dx,$$

where $p : \mathbb{R} \to \mathbb{R}$ is a probability density function (pdf) with $p(x) = 0$ for all $x < 0$ and $p(x) > 0$ for all $x > 0$.

a) Define the corresponding cumulative distribution function (cdf) P.
Answer: The cumulative distribution function (cdf) P is

$$P(x) = \text{Prob}(X \le x) = \begin{cases} 0 & x \le 0; \\ \displaystyle\int_0^x p(u)\,du & x > 0. \end{cases}$$

Make sure the limit of integration and the dummy variable of integration are different ($\int_0^x p(x)\,dx$ is not correct), and the behaviour for $x < 0$ is defined.

b) What properties does P possess?
Answer: Two key properties are

- $P(0) = 0$ and $\lim_{x\to\infty} P(x) = 1$.
- As $p(x) > 0$ for all $x > 0$, the cdf $P(x)$ is strictly increasing on $[0, \infty)$. Hence the function $P : [0, \infty) \to [0, 1]$ has an inverse.

Note that P is not strictly increasing on the whole of $\mathbb{R}$, and does not have an inverse as a function $P : \mathbb{R} \to [0, 1]$.

c) Use the cdf P to transform the expected value $\mathbb{E}[h]$ into an integral over $[0, 1]$.
Answer: Let $z = P(x)$ for $x \ge 0$, so the inverse P^{-1} exists and $x = P^{-1}(z)$. Then

- For $z \in [0, 1]$, $x = P^{-1}(z) \in [0, \infty)$.
- The derivative $\frac{dz}{dx} = P'(x) = p(x)$, so $dz = p(x)\,dx$.
- The limits of integration are: $x = 0 \iff z = P(0) = 0$ and $x = \infty \iff z = P(\infty) = 1$.

Thus

$$\mathbb{E}[h] = \int_0^\infty h(x)p(x)\,dx = \int_0^1 h(P^{-1}(z))\,dz.$$

ii) Consider estimating the d–dimensional variance

$$\text{Var}[f] = \mathbb{E}[(f - \mu)^2] = \int_{[0,1]^d} (f(\boldsymbol{x}) - \mu)^2 p(\boldsymbol{x})d\boldsymbol{x}, \tag{4.1}$$

where $p : [0, 1]^d \to \mathbb{R}$ is a probability density function (pdf) and $\mu = \mathbb{E}[f]$ is known. A "low-discrepancy" point set $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ has

$$D^*(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N) \le c_d \, \frac{(\log N)^{d-1}}{N}, \tag{4.2}$$

where c_d is a positive constant which does not depend on N. The Koksma-Hlawka inequality is

$$\left| \int_{[0,1]^d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - \frac{1}{N} \sum_{k=1}^{N} f(\boldsymbol{x}_k) \right| \leq V(f) \; D^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N). \tag{4.3}$$

a) Explain what the star discrepancy $D^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ is.
Answer: The star discrepancy of s set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of points in $[0,1]^d$ is

$$D^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \sup_{\boldsymbol{y} \in [0,1]^d} \left| \frac{\# \boldsymbol{x}_j \in [\mathbf{0}, \boldsymbol{y}]}{N} - \frac{\prod_{j=1}^d y_j}{1} \right|.$$

This is the largest different between the proportion of the points lying in the rectangular box with vertices at the origin and the point $\boldsymbol{y} \in [0,1]^d$ with sides parallel to the axes and the volume of the box (as a proportion of the unit cube which has volume 1). The supremum (least upper bound, or maximum if it is achieved) is taken over all points $\boldsymbol{y} \in [0,1]^d$, so over all such rectangular boxes in the unit cube. The star discrepancy measures the uniformity of the distribution of the points $\boldsymbol{x}_j$, $j = 1, \ldots, N$ in the unit cube.

b) Outline the key steps in a Monte-Carlo method to estimate (4.1).
Answer: The Monte-Carlo method generates points $\boldsymbol{x}_j$, $j = 1, \ldots, N$ uniformly distributed over the unit cube $[0,1]^d$ and then uses the sample mean to estimate the expected value (4.1), so

$$\int_{[0,1]^d} (f(\boldsymbol{x}) - \mu)^2 p(\boldsymbol{x}) d\boldsymbol{x} \approx \frac{1}{N} \sum_{j=1}^{N} (f(\boldsymbol{x}_j) - \mu)^2 p(\boldsymbol{x}_j).$$

Alternatively you can generate the points $\boldsymbol{x}_j$, $j = 1, \ldots, N$ uniformly distributed **according to the distribution defined by the pdf** $p(\boldsymbol{x})$, then

$$\int_{[0,1]^d} (f(\boldsymbol{x}) - \mu)^2 p(\boldsymbol{x}) d\boldsymbol{x} \approx \frac{1}{N} \sum_{j=1}^{N} (f(\boldsymbol{x}_j) - \mu)^2.$$

c) Compare the integration errors when using a Monte-Carlo method with using a "low-discrepancy" point set.
Answer: The Monte-Carlo method has root mean square (RMS) error of the form

$$\frac{\sigma(f)}{N^{\frac{1}{2}}}.$$

The key advantage is that this does not depend directly on the dimension d (the variance $\sigma^2(f)$ of the function does depend on the dimension d). The disadvantage is that the convergence rate of $O(N^{-\frac{1}{2}})$ is slow.

For "low-discrepancy" point sets, the Koksma-Hllwaka inequality shows that the error for numerical integration is bounded by

$$V(f)\, c_d\, \frac{(\log N)^{d-1}}{N}.$$

For small dimensions d and large N, this behaves like $O\left(\frac{1}{N}\right)$ which should be better than the Monte-Carlo method. However for modest/higher values of d, the $\log(N)^{d-1}$ term cannot be treated as a constant, unless N is very very large (so large that computation is unrealistic).

d) Consider a problem with $d = 20$ and $N = 10^{20}$.

A) Evaluate $(\log N)^{d-1}$ and $N^{\frac{1}{2}}$.

B) What are the implications for using a "low-discrepancy" point set satisfying (4.2)?

Answer:

A) For $d = 20$ and $N = 10^{20}$

$$(\log N)^{d-1} = (\log 10^{20})^{19} = 4 \times 10^{31}, \qquad N^{\frac{1}{2}} = 10^{10}.$$

B) Ignoring constants and the variation of the integrand, the bound on the numerical integration error for a "low-discrepancy" point set is better than the RMS error for the Monte-Carlo method when

$$\frac{(\log N)^{d-1}}{N} < \frac{1}{N^{\frac{1}{2}}} \iff (\log N)^{d-1} < N^{\frac{1}{2}}.$$

For $d = 20$ and $N = 10^{20}$ this is no where near being true, indicating the for these values the Monte-Carlo method would be preferable. For large enough N the low-discrepancy bound would be smaller, but the values of N required would make it totally unrealistic for use with today's computers.

e) Outline the key ideas in using a function g which approximates f^2 in a variance reduction technique to estimate $\mathrm{Var}[f]$.

Answer: The expected vale $\mu = \mathbb{E}[f]$ is known, so the variance of f is calculated by

$$\mathrm{Var}[f] = \mathbb{E}[(f-\mu)^2] = \mathbb{E}[f^2] - \mu^2.$$

Let g be a function which approximates f^2 such that $\mathbb{E}[g]$ is known analytically. Then, using the linearity of the expectation,

$$\mathrm{Var}[f] = \mathbb{E}[f^2 - g + g] - \mu^2 = \mathbb{E}[f^2 - g] + \mathbb{E}[g] - \mu^2.$$

If g is a good approximation of f^2, so that $\sigma(f^2 - g) \ll \sigma(f^2)$ (variance reduction) then using a numerical method to approximate $\mathbb{E}[f^2 - g]$

then adding $\mathbb{E}[g] - \mu^2$ should produce a more accurate estimate of $\mathrm{Var}[f]$ that numerically calculating $\mathbb{E}[f^2]$ directly.

For a Monte-Carlo method using points uniformly distributed in $[0,1]^d$,

$$\mathbb{E}[f^2 - g] \approx \frac{1}{N}\sum_{j=1}^{N} \left(f(\boldsymbol{x}_j)^2 - g(\boldsymbol{x}_j)\right) p(\boldsymbol{x}_j).$$

iii) In the CEV model a process S_t follows the SDE

$$\frac{dS_t}{S_t} = \mu dt + \sigma S_t^{\gamma-1} dW_t, \qquad t \in (0,T], \tag{4.4}$$

where W_t is a Wiener process, μ, σ, γ are positive constants, and S_0 is given.

a) Derive an expression for S_{t+dt} from (4.4).
Answer: From the CEV model

$$dS_t = S_t\left(\mu dt + \sigma S_t^{\gamma-1} dW_t\right).$$

Hence

$$S_{t+dt} \approx S_t + dS_t = S_t\left(1 + \mu dt + \sigma S_t^{\gamma-1} dW_t\right) = S_t + \mu S_t dt + \sigma S_t^{\gamma} dW_t.$$

b) What are the key properties of a Wiener process W_t?
Answer: A Wiener process W_t satisfies

- $W_0 = 0$ with probability 1,
- W_t is continuous (almost surely),
- W_t has increments $W_t - W_s \sim N(0, t-s)$, so they are normally distributed with mean 0 and variance $t - s$.
- The increments $W_t - W_s$ and $W_u - W_v$ are independent for $0 \le s < t \le u < v$.

c) Let the time interval $[0,T]$ be discretized by

$$t_i = i\Delta t, \quad i = 0, ..., m, \qquad \Delta t = T/m.$$

Write down the formula for a discrete approximation $\bar{S}_i \approx S_{t_i}, i = 1, \ldots, m$ of the CEV process (4.4).
Answer: The discrete approximation of the CEV process satisfies

$$\bar{S}_{i+1} = \bar{S}_i\left(1 + \mu\Delta t + \sigma S_i^{\gamma-1}\sqrt{\Delta t}\, Z_i\right)$$

where Z_i are independent standard normal $N(0,1)$ random variables. Starting from a given S_0 this can be used to simulate the CEV process.