Hopscotch methods for two-state financial models*

Adam Kurpiel†
L.A.R.E., Université Montesquieu-Bordeaux IV, France

Thierry Roncalli‡
FERC, City University Business School, England

January 15, 1999

Abstract

In this paper, we consider Hopscotch methods for solving two-state financial models. We first derive a solution algorithm for two-dimensional partial differential equations with mixed boundary conditions. We then consider a number of financial applications including stochastic volatility option pricing, term structure modelling with two states and elliptic irreversible investment problems.

1 Introduction

The contributions of Black and Scholes [1973] and Merton [1973] to contingent claims pricing theory are clearly some of the most significant in the development of finance theory. The consistent use of arbitrage theory leading to their well known solution for pricing options.Vasicek [1977], using his term structure model, provided another important development in the area of contingent claims deriving a solution for a bond price that has to satisfy a particular partial differential equation. To obtain the solution, Vasicek used the Feynman-Kac representation and the Girsanov theorem and showed the link between the partial differential equation and martingale approaches. This relationship has subsequently been extensively exploited to find symbolic solutions for a number of contingent claim valuation problems. Moreover, the link is fundamental for numerical solutions based on Monte Carlo methods.

Both these models however only consider one state variable whereas option and bond pricing theory has now been extended to take into account more state variables. For example, in the famous model of Longstaff and Schwartz [1992], **both** the instantaneous interest rate $r(t)$ and the volatility measure $V(t)$ are stochastic. Balduzzi, Das, Foresi and Sundaram [1996] present a term structure model with a third state variable, the mean reversion parameter and Canabarro [1995] describes why in general more than one state variable may be needed in term structure modelling. One of the main difficulties in option theory has been to capture the *smile curve* and number of authors have introduced stochastic volatility (Hull and White [1987], Wiggins [1987]). However, explicit analytic solutions are available for only a few models and Monte Carlo methods have been used extensively to find numerical solutions. These methods, however fail to provide accurate solutions for the greeks and delta/gamma hedging. Moreover, they can not be used for American option pricing, because there is no Feynman-Kac representation and a variational inequality problem has to be solved (Lamberton and Lapeyre [1997]).

Gordon [1965] and Gourlay [1970] introduced a class of, so called, *Hopscotch* algorithms to solve parabolic and elliptic partial differential equations in two or more state variables although their utility in financial applications has not yet been realised. The purpose of this paper is then to present Hopscotch methods and to demonstrate how they can be used to solve financial models with two-state variables.

*We thank Eric Bouyé and Mark Salmon for their helpful comments.
†email: kurpiel@montesquieu.u-bordeaux.fr
‡email: t.roncalli@city.ac.uk

The paper is organized as follows. In section two, we present the Hopscotch algorithm. In particular, we formulate a problem which is more general than those considered by Gourlay. Moreover, we show how to take mixed boundary conditions into account[1]. We then analyse the stability issue and propose efficient programming methods. In section three, we consider a number of financial models and solve them with the hopscotch algorithm. Section four concludes and suggests directions for further research.

2 Hopscotch methods

We consider the linear parabolic equation

$$\frac{\partial u(t,x,y)}{\partial t} + f(t,x,y)\,u(t,x,y) = \mathcal{A}_t u(t,x,y) + g(t,x,y) \tag{1}$$

where $\mathcal{A}_t$ is the general two dimensional differential operator

$$\begin{aligned}\mathcal{A}_t u(t,x,y) &= a(t,x,y)\frac{\partial^2 u(t,x,y)}{\partial x^2} + 2b(t,x,y)\frac{\partial^2 u(t,x,y)}{\partial x\partial y} + c(t,x,y)\frac{\partial^2 u(t,x,y)}{\partial y^2} + \\ &\quad d(t,x,y)\frac{\partial u(t,x,y)}{\partial x} + e(t,x,y)\frac{\partial u(t,x,y)}{\partial y}\end{aligned} \tag{2}$$

The idea is to solve (1) in a region of the (t,x,y) space given by $\mathfrak{T}\times\mathfrak{R}$ where $\mathfrak{R}$ is a closed region of the (x,y) plane with a continuous boundary $\partial\mathfrak{R}$. In particular, for convenient computation, we propose that

$$\mathfrak{R} = \left[x^-, x^+\right] \times \left[y^-, y^+\right]$$

$$\mathfrak{T} = \left[t^-, t^+\right]$$

To solve (1) numerically , we need to impose some boundary conditions. For $t = t^-$, we consider a Dirichlet condition. For the other boundary, we could choose between a Dirichlet or a Neumann condition.

$$\begin{aligned}
&u\left(t^-,x,y\right) = u_{(t^-)}(x,y) \\
&u\left(t,x^-,y\right) = u_{(x^-)}(t,y) \quad \bigvee \quad \left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^-} = u'_{(x^-)}(t,y) \\
&u\left(t,x^+,y\right) = u_{(x^+)}(t,y) \quad \bigvee \quad \left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^+} = u'_{(x^+)}(t,y) \\
&u\left(t,x,y^-\right) = u_{(y^-)}(t,x) \quad \bigvee \quad \left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^-} = u'_{(y^-)}(t,x) \\
&u\left(t,x,y^+\right) = u_{(y^+)}(t,x) \quad \bigvee \quad \left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^+} = u'_{(y^+)}(t,x)
\end{aligned} \tag{3}$$

There are several differences here compared to Gourlay [7,8,9]. First, we have changed the problem (1) in order to take account of the first derivatives in the $\mathcal{A}_t$ operator. Moreover, we have introduced a new term $f(t,x,y)\,u(t,x,y)$ in the Partial Differential Equation. These modifications are necessary to ensure that the fundamental equation of finance can be written in this form. Gourlay did not in fact consider how to introduce boundary conditions into the algorithm.

[1] This is important because mixed boundary conditions frequently arise when solving financial models. For instance, consider the Black and Scholes problem. Let $C(t,S)$ be the call option price at time t on an asset whose current price is S. Then, we have a Dirichlet boundary condition for S equal to 0

$$C(t,0) = 0$$

and a Neumann boundary condition for S equal to ∞

$$C_S(t,\infty) = 1$$

2.1 The Hopscotch algorithm

Hopscotch methods are based on a vec form of finite difference methods. In other words the idea exploits the same formulation as used when multidimensional arrays are stored in a computing language, where matrices do not exist physically, but are in fact stored in rows.

2.1.1 Notation

In order to develop a numerical solution for (1), we need to discretise the process $u(t,x,y)$ in both time and space dimensions. Let N_t, N_x and N_y be the number of discretisation points for t, x and y respectively. We denote by k, h_x and h_y the mesh spacings in time and space in the x and y directions respectively. Then, we have

$$k = \frac{t^+ - t^-}{N_t - 1}$$

$$h_x = \frac{x^+ - x^-}{N_x - 1}$$

$$h_y = \frac{y^+ - y^-}{N_y - 1}$$

We note

$$\begin{aligned} t_m &= t^- + m \cdot k \\ x_i &= x^- + i \cdot h_x \\ y_j &= y^- + j \cdot h_y \end{aligned}$$

Let $u_{i,j}^m$ be the approximate solution to (1) at the grid point (t_m, x_i, y_j) and $u(t_m, x_i, y_j)$ the exact solution of the Partial Differencial Equation at this point.

Let M be the matrix with (i,j) entry $(M_{i,j})$ and denote $\mathrm{vec}(M)$ by $\mathbf{m}$..

2.1.2 The Algorithm

The explicit form of equation (1) is

$$\frac{u_{i,j}^{m+1} - u_{i,j}^m}{k} + f_{i,j}^m u_{i,j}^m = \mathbf{A}_{i,j}^m + g_{i,j}^m \tag{4}$$

while the implicit form is

$$\frac{u_{i,j}^{m+1} - u_{i,j}^m}{k} + f_{i,j}^{m+1} u_{i,j}^{m+1} = \mathbf{A}_{i,j}^{m+1} + g_{i,j}^{m+1} \tag{5}$$

Introducing theta-schemes gives

$$\begin{aligned} \left(1 + k\theta_{i,j}^{m+1} f_{i,j}^{m+1}\right) u_{i,j}^{m+1} &- k\theta_{i,j}^{m+1}\left(\mathbf{A}_{i,j}^{m+1} + g_{i,j}^{m+1} + p_{i,j}^{m+1}\right) \\ &= \left(1 - k\theta_{i,j}^m f_{i,j}^m\right) u_{i,j}^m + k\theta_{i,j}^m \left(\mathbf{A}_{i,j}^m + g_{i,j}^m + p_{i,j}^m\right) \end{aligned} \tag{6}$$

with

$$\theta_{i,j}^{m+1} + \theta_{i,j}^m = 1 \tag{7}$$

We can show that there exists a square matrix H_m and a vector $\mathbf{p}_m$ such that

$$\mathbf{A}^m = H_m \mathbf{u}_m + \mathbf{p}_m \tag{8}$$

We call $p_{i,j}^m$ the residual absorption function. Then, we have

$$\begin{aligned} \left[I + k\Theta_{m+1}\mathbf{f}_{m+1} - k\Theta_{m+1}H_{m+1}\right]\mathbf{u}_{m+1} &= \left[I - k\Theta_m\mathbf{f}_m + k\Theta_m H_m\right]\mathbf{u}_m + \\ & k\left[\Theta_{m+1}\mathbf{g}_{m+1} + \Theta_m\mathbf{g}_m\right] + k\left[\Theta_{m+1}\mathbf{p}_{m+1} + \Theta_m\mathbf{p}_m\right] \end{aligned} \tag{9}$$

with

$$\Theta_m = \text{diag}\left(\boldsymbol{\theta}_m\right)$$

and

$$\boldsymbol{\theta}_m = \left(\theta^m_{i,j}\right)$$

The equation (9) is the general vec form of finite difference methods for two-dimensional Partial Differential Equations. We have now to specify $\mathbf{A}^m_{i,j}$ and Θ_m. Following GOURLAY and MCGUIRE [1971], we do not for the moment fix the choice of $\mathbf{A}^m_{i,j}$ and the choice of Θ_m .

2.1.3 Discretisation schemes and the choice of the filling hopscotch method

We first present the general scheme and show how to take account of boundary conditions. Then ,we propose two specific discretisation schemes in the spirit of GOURLAY and MCKEE [1977].

2.1.3.1 The general scheme. We have

$$\mathbf{A}^m_{i,j} = a^m_{i,j}\delta_{xx}u^m_{i,j} + 2b^m_{i,j}\delta_{xy}u^m_{i,j} + c^m_{i,j}\delta_{yy}u^m_{i,j} + d^m_{i,j}\delta_x u^m_{i,j} + e^m_{i,j}\delta_y u^m_{i,j} \tag{10}$$

One difficulty is the choice of the five operators δ_{xx}, δ_{xy}, δ_{yy}, δ_x and δ_y.

We consider the following three operators δ^-_x, δ^0_x and δ^+_x

$$\delta^-_x u^m_{i,j} = \frac{u^m_{i,j} - u^m_{i-1,j}}{h_x} \tag{11}$$

$$\delta^+_x u^m_{i,j} = \frac{u^m_{i+1,j} - u^m_{i,j}}{h_x} \tag{12}$$

$$\delta^0_x u^m_{i,j} = \frac{1}{2}\left(\delta^-_x u^m_{i,j} + \delta^+_x u^m_{i,j}\right) = \frac{u^m_{i+1,j} - u^m_{i-1,j}}{2h_x} \tag{13}$$

We have a choice between these three alternatives for the discretisation of the first derivatives.The most common operator in numerical analysis is δ^0_x. For the second derivatives - there are many possibilities. For example, we could choose the traditional scheme

$$\delta_{xx}u^m_{i,j} = \delta^+_x\delta^-_x u^m_{i,j} = \delta^-_x\delta^+_x u^m_{i,j} = \frac{u^m_{i+1,j} - 2u^m_{i,j} + u^m_{i-1,j}}{h^2_x} \tag{14}$$

but we could also consider other possibilities such as, $\delta^+_x\delta^+_x$, $\delta^0_x\delta^0_x$ or $\delta^-_x\delta^0_x$. For the mixed derivatives, choice between the alternatives is very important. Gourlay and McKee [1977] made the following choice :

	δ_x	δ_y	δ_{xx}	δ_{yy}	δ_{xy}
original “ordered odd-even” hopscotch	✓	✓	$\delta^+_x\delta^-_x$	$\delta^+_y\delta^-_y$	$\frac{1}{2}\left(\delta^+_x\delta^-_y + \delta^-_x\delta^+_y\right)$
original “line” hopscotch	✓	✓	$\delta^+_x\delta^-_x$	$\delta^+_y\delta^-_y$	$\delta^0_x\delta^0_y$

The general form of $\mathbf{A}^m_{i,j}$ is

$$\mathbf{A}^m_{i,j} = \sum_{\tilde{\imath}=-2}^{2}\sum_{\tilde{\jmath}=-2}^{2} \delta^m_{i,j,\tilde{\imath},\tilde{\jmath}} u^m_{i+\tilde{\imath},j+\tilde{\jmath}} \tag{15}$$

Then, we can write the matrix $\mathbf{A}^m$,with elements $(\mathbf{A}^m_{i,j})$, in the following form

$$\mathbf{A}^m = \Delta_m \mathbf{u}_m + \mathbf{q}_m \tag{16}$$

In this case, the structure of the Δ_m matrix is the following

$$\Delta_m = \begin{bmatrix}
\Delta^m_{1,0} & \Delta^m_{1,1} & \Delta^m_{1,2} & & & & & & \\
\Delta^m_{2,-1} & \Delta^m_{2,0} & \Delta^m_{2,1} & \Delta^m_{2,2} & & & & & \\
\Delta^m_{3,-2} & \Delta^m_{3,-1} & \Delta^m_{3,0} & \Delta^m_{3,1} & \Delta^m_{3,2} & & & & \\
& & & \ddots & & & & & \\
& & \Delta^m_{j,-2} & \Delta^m_{j,-1} & \Delta^m_{j,0} & \Delta^m_{j,1} & \Delta^m_{j,2} & & \\
& & & & & \ddots & & & \\
& & & & \Delta^m_{N_y-2,-2} & \Delta^m_{N_y-2,-1} & \Delta^m_{N_y-2,0} & \Delta^m_{N_y-2,1} & \Delta^m_{N_y-2,2} \\
& & & & & \Delta^m_{N_y-1,-2} & \Delta^m_{N_y-1,-1} & \Delta^m_{N_y-1,0} & \Delta^m_{N_y-1,1} \\
& & & & & & \Delta^m_{N_y,-2} & \Delta^m_{N_y,-1} & \Delta^m_{N_y,0}
\end{bmatrix} \tag{17}$$

and the residual absorption vector, $\mathbf{q}_m$ acts just like an adjustment

$$\mathbf{q}_m := \mathbf{A}^m - \Delta_m \mathbf{u}_m \tag{18}$$

In fact, $\mathbf{q}_m$ reflects the boundary conditions. When we use them, we can split the vector $\mathbf{q}_m$ and have

$$\mathbf{q}_m = \Lambda_m \mathbf{u}_m + \mathbf{p}_m \tag{19}$$

Then, it is clear that H_m in equation (9) is

$$H_m = \Delta_m + \Lambda_m \tag{20}$$

The nature of the boundary condition is important, because a Dirichlet condition will influence the $\mathbf{p}_m$ vector while a Neuman condition will affect the Λ_m matrix.

Now, let us consider the problem of boundary conditions in the 2D problem. For the Dirichlet conditions, we have

Boundary conditions	Numerical approximation
$u\left(t, x^-, y\right) = u_{(x^-)}\left(t, y\right)$	$u^m_{0,j} = u_{(x^-)}\left(t_m, y_j\right)$
$u\left(t, x^+, y\right) = u_{(x^+)}\left(t, y\right)$	$u^m_{N_x+1,j} = u_{(x^+)}\left(t_m, y_j\right)$
$u\left(t, x, y^-\right) = u_{(y^-)}\left(t, x\right)$	$u^m_{i,0} = u_{(y^-)}\left(t_m, x_i\right)$
$u\left(t, x, y^+\right) = u_{(y^+)}\left(t, x\right)$	$u^m_{i,N_y+1} = u_{(y^+)}\left(t_m, x_i\right)$

and for the Neuman conditions, we suggest the following numerical substitutions

Boundary conditions	Numerical approximaiton
$\left.\frac{\partial\, u(t,x,y)}{\partial\, x}\right\rvert_{x=x^-} = u'_{(x^-)}\left(t, y\right)$	$u^m_{0,j} = u^m_{1,j} - h_x \acute{u}^m_{0,j}$ $\acute{u}^m_{0,j} = u'_{(x^-)}\left(t_m, y_j\right)$
$\left.\frac{\partial\, u(t,x,y)}{\partial\, x}\right\rvert_{x=x^+} = u'_{(x^+)}\left(t, y\right)$	$u^m_{N_x+1,j} = u^m_{N_x,j} + h_x \acute{u}^m_{N_x+1,j}$ $\acute{u}^m_{N_x+1,j} = u'_{(x^+)}\left(t_m, y_j\right)$
$\left.\frac{\partial\, u(t,x,y)}{\partial\, y}\right\rvert_{y=y^-} = u'_{(y^-)}\left(t, x\right)$	$u^m_{i,0} = u^m_{i,1} - h_y \acute{u}^m_{i,0}$ $\acute{u}^m_{i,0} = u'_{(y^-)}\left(t_m, x_i\right)$
$\left.\frac{\partial\, u(t,x,y)}{\partial\, y}\right\rvert_{y=y^+} = u'_{(y^+)}\left(t, x\right)$	$u^m_{i,N_y+1} = u^m_{i,N_y} + h_y \acute{u}^m_{i,N_y+1}$ $\acute{u}^m_{i,N_y+1} = u'_{(y^+)}\left(t_m, x_i\right)$

We note that we face some restrictions when we define the five operators δ_{xx}, δ_{xy}, δ_{yy}, δ_x and δ_y given to the decomposition $\mathbf{q}_m = \Lambda_m \mathbf{u}_m + \mathbf{p}_m$. It is necessary that the absolute values of $\tilde{\imath}$ and $\tilde{\jmath}$ are different from 2. That implies that the $\Delta^m_{j,-2}$ and $\Delta^m_{j,2}$ matrices are null matrices and that $\Delta^m_{j,-1}$, $\Delta^m_{j,0}$ and $\Delta^m_{j,1}$ are tridiagonal matrices. In this case, we note that this implies there is only **one** possible scheme for the second order derivatives,$\delta_{xx} = \delta^+_x \delta^-_x$, and all the other schemes are excluded[2]. However there are no restrictions on the first derivatives and the mixed derivatives schemes. It is also clear that Δ_m corresponds to the following specification

$$\Delta^m_{j,\tilde{\jmath}} = \begin{bmatrix} \delta^m_{1,j,0,\tilde{\jmath}} & \delta^m_{1,j,1,\tilde{\jmath}} & & & & & & & \\ \delta^m_{2,j,-1,\tilde{\jmath}} & \delta^m_{2,j,0,\tilde{\jmath}} & \delta^m_{2,j,1,\tilde{\jmath}} & & & & & & \\ & \delta^m_{3,j,-1,\tilde{\jmath}} & \delta^m_{3,j,0,\tilde{\jmath}} & \delta^m_{3,j,1,\tilde{\jmath}} & & & & & \\ & & & \ddots & & & & & \\ & & & \delta^m_{i,j,-1,\tilde{\jmath}} & \delta^m_{i,j,0,\tilde{\jmath}} & \delta^m_{i,j,1,\tilde{\jmath}} & & & \\ & & & & & \ddots & & & \\ & & & & & \delta^m_{N_x-2,j,-1,\tilde{\jmath}} & \delta^m_{N_x-2,j,0,\tilde{\jmath}} & \delta^m_{N_x-2,j,1,\tilde{\jmath}} & \\ & & & & & & \delta^m_{N_x-1,j,-1,\tilde{\jmath}} & \delta^m_{N_x-1,j,0,\tilde{\jmath}} & \delta^m_{N_x-1,j,1,\tilde{\jmath}} \\ & & & & & & & \delta^m_{N_x,j,-1,\tilde{\jmath}} & \delta^m_{N_x,j,0,\tilde{\jmath}} \end{bmatrix} \tag{21}$$

To determine Λ_m and $\mathbf{p}_m$, we integrate the boundary conditions. These matrices could be determined by initially setting them to null matrices and updated sequentially. We however need to be careful with the 2D case unlike the one-dimensional case which is straight forward because it only concerns two points. In 2D case, $\partial\,\mathfrak{R}$ is a square, i.e. 4 segments and 4 corners[3]. So, we have to distinguish the segments case ($2 \le i \le N_x - 1$ and $2 \le j \le N_y - 1$) and the corners case ($i = 1, N_x$ and $j = 1, N_x$).

For the **segments** case, we have :

- Conditions on x^-
  - $u\left(t, x^-, y\right) = u_{(x^-)}\left(t, y\right)$

$$(\mathbf{p}_m)_{1+(j-1)N_x} \leftarrow \delta^m_{1,j,-1,-1} u^m_{0,j-1} + \delta^m_{1,j,-1,0} u^m_{0,j} + \delta^m_{1,j,-1,1} u^m_{0,j+1}$$

  - $\left.\frac{\partial\, u(t,x,y)}{\partial\, x}\right|_{x=x^-} = u'_{(x^-)}\left(t, y\right)$

$$\begin{aligned} \lambda^m_{1,j,0,-1} &\leftarrow \delta^m_{1,j,-1,-1} \\ \lambda^m_{1,j,0,0} &\leftarrow \delta^m_{1,j,-1,0} \\ \lambda^m_{1,j,0,1} &\leftarrow \delta^m_{1,j,-1,1} \\ (\mathbf{p}_m)_{1+(j-1)N_x} &\leftarrow -h_x\left(\delta^m_{1,j,-1,-1} \acute{u}^m_{0,j-1} + \delta^m_{1,j,-1,0} \acute{u}^m_{0,j} + \delta^m_{1,j,-1,1} \acute{u}^m_{0,j+1}\right) \end{aligned}$$

- Conditions on x^+
  - $u\left(t, x^+, y\right) = u_{(x^+)}\left(t, y\right)$

$$(\mathbf{p}_m)_{N_x+(j-1)N_x} \leftarrow \delta^m_{N_x,j,1,-1} u^m_{N_x+1,j-1} + \delta^m_{N_x,j,1,0} u^m_{N_x+1,j} + \delta^m_{N_x,j,1,1} u^m_{N_x+1,j+1}$$

[2]They are $\delta^-_x \delta^-_x$, $\delta^0_x \delta^0_x$, $\delta^+_x \delta^+_x$, $\delta^0_x \delta^-_x$ and $\delta^+_x \delta^0_x$.

[3]And the problem becomes very complicated in 3D case, because $\partial\,\mathfrak{R}$ is a box with 6 planes, 12 edges and 8 corners.

- $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^+} = u'_{(x^+)}(t,y)$

$$\begin{array}{rcl}
\lambda^m_{N_x,j,0,-1} & \leftarrow & \delta^m_{N_x,j,1,-1} \\
\lambda^m_{N_x,j,0,0} & \leftarrow & \delta^m_{N_x,j,1,0} \\
\lambda^m_{N_x,j,0,1} & \leftarrow & \delta^m_{N_x,j,1,1} \\
(\mathbf{p}_m)_{N_x+(j-1)N_x} & \leftarrow & h_x\left(\delta^m_{N_x,j,-1,-1}\acute{u}^m_{N_x+1,j-1} + \delta^m_{N_x,j,-1,0}\acute{u}^m_{N_x+1,j} + \delta^m_{N_x,j,-1,1}\acute{u}^m_{N_x+1,j+1}\right)
\end{array}$$

- Conditions on y^-
  - $u(t,x,y^-) = u_{(y^-)}(t,x)$

$$(\mathbf{p}_m)_i \leftarrow \delta^m_{i,1,-1,-1}u^m_{i-1,0} + \delta^m_{i,1,0,-1}u^m_{i,0} + \delta^m_{i,1,1,-1}u^m_{i+1,0}$$

  - $\left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^-} = u'_{(y^-)}(t,x)$

$$\begin{array}{rcl}
\lambda^m_{i,1,-1,0} & \leftarrow & \delta^m_{i,1,-1,-1} \\
\lambda^m_{i,1,0,0} & \leftarrow & \delta^m_{i,1,0,-1} \\
\lambda^m_{i,1,1,0} & \leftarrow & \delta^m_{i,1,1,-1} \\
(\mathbf{p}_m)_i & \leftarrow & -h_y\left(\delta^m_{i,1,-1,-1}\acute{u}^m_{i-1,0} + \delta^m_{i,1,0,-1}\acute{u}^m_{i,0} + \delta^m_{i,1,1,-1}\acute{u}^m_{i+1,0}\right)
\end{array}$$

- Conditions on y^+
  - $u(t,x,y^+) = u_{(y^+)}(t,x)$

$$(\mathbf{p}_m)_{i+N_x(N_y-1)} \leftarrow \delta^m_{i,N_y,-1,1}u^m_{i-1,N_y+1} + \delta^m_{i,N_y,0,1}u^m_{i,N_y+1} + \delta^m_{i,N_y,1,1}u^m_{i+1,N_y+1}$$

  - $\left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^+} = u'_{(y^+)}(t,x)$

$$\begin{array}{rcl}
\lambda^m_{i,N_y,-1,0} & \leftarrow & \delta^m_{i,N_y,-1,1} \\
\lambda^m_{i,N_y,0,0} & \leftarrow & \delta^m_{i,N_y,0,1} \\
\lambda^m_{i,N_y,1,0} & \leftarrow & \delta^m_{i,N_y,1,1} \\
(\mathbf{p}_m)_{i+N_x(N_y-1)} & \leftarrow & h_y\left(\delta^m_{i,N_y,-1,1}\acute{u}^m_{i-1,N_y+1} + \delta^m_{i,N_y,0,1}\acute{u}^m_{i,N_y+1} + \delta^m_{i,N_y,1,1}\acute{u}^m_{i+1,N_y+1}\right)
\end{array}$$

For the **corners** case, we have :

- Conditions on x^- and y^-
  - $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^-} = u'_{(x^-)}(t,y) \bigwedge \left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^-} = u'_{(y^-)}(t,x)$

$$\begin{array}{rcl}
\lambda^m_{1,1,0,0} & \leftarrow & \delta^m_{1,1,-1,-1} \\
(\mathbf{p}_m)_1 & \leftarrow & \delta^m_{1,1,-1,-1}\left(-h_x\acute{u}^m_{0,0} - h_y\acute{u}^m_{1,0}\right) \\
 & \bigvee & \delta^m_{1,1,-1,-1}\left(-h_x\acute{u}^m_{0,1} - h_y\acute{u}^m_{0,0}\right) \\
 & \bigvee & \frac{1}{2}\delta^m_{1,1,-1,-1}\left(-h_x\left(\acute{u}^m_{0,0} + \acute{u}^m_{0,1}\right) - h_y\left(\acute{u}^m_{0,0} + \acute{u}^m_{1,0}\right)\right)
\end{array}$$

- $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^-} = u'_{(x^-)}(t,y) \bigwedge u(t,x,y^-) = u_{(y^-)}(t,x)$

$$(\mathbf{p}_m)_1 \leftarrow \delta^m_{1,1,-1,-1}\left(u^m_{1,0} - h_x \acute{u}^m_{0,0}\right) \bigvee \delta^m_{1,1,-1,-1} u^m_{0,0}$$

- $\left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^-} = u'_{(y^-)}(t,x) \bigwedge u(t,x^-,y) = u_{(x^-)}(t,y)$

$$(\mathbf{p}_m)_1 \leftarrow \delta^m_{1,1,-1,-1}\left(u^m_{0,1} - h_y \acute{u}^m_{0,0}\right) \bigvee \delta^m_{1,1,-1,-1} u^m_{0,0}$$

- $u(t,x^-,y) = u_{(x^-)}(t,y) \bigwedge u(t,x,y^-) = u_{(y^-)}(t,x)$

$$(\mathbf{p}_m)_1 \leftarrow \delta^m_{1,1,-1,-1} u^m_{0,0}$$

- Conditions on x^- and y^+
  - $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^-} = u'_{(x^-)}(t,y) \bigwedge \left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^+} = u'_{(y^+)}(t,x)$

$$\begin{array}{rcl} \lambda^m_{1,Ny,0,0} & \leftarrow & \delta^m_{1,Ny,-1,1} \\ (\mathbf{p}_m)_{1+N_x(Ny-1)} & \leftarrow & \delta^m_{1,Ny,-1,1}\left(-h_x \acute{u}^m_{0,N_y+1} + h_y \acute{u}^m_{1,N_y+1}\right) \\ & \bigvee & \delta^m_{1,Ny,-1,1}\left(-h_x \acute{u}^m_{0,N_y} + h_y \acute{u}^m_{0,N_y+1}\right) \\ & \bigvee & \frac{1}{2}\delta^m_{1,Ny,-1,1}\left(-h_x\left(\acute{u}^m_{0,N_y} + \acute{u}^m_{0,N_y+1}\right) + h_y\left(\acute{u}^m_{0,N_y+1} + \acute{u}^m_{1,N_y+1}\right)\right) \end{array}$$

  - $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^-} = u'_{(x^-)}(t,y) \bigwedge u(t,x,y^+) = u_{(y^+)}(t,x)$

$$(\mathbf{p}_m)_{1+N_x(Ny-1)} \leftarrow \delta^m_{1,Ny,-1,1}\left(u^m_{1,N_y+1} - h_x \acute{u}^m_{0,N_y+1}\right) \bigvee \delta^m_{1,Ny,-1,1} u^m_{0,N_y+1}$$

  - $\left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^+} = u'_{(y^+)}(t,x) \bigwedge u(t,x^-,y) = u_{(x^-)}(t,y)$

$$(\mathbf{p}_m)_{1+N_x(Ny-1)} \leftarrow \delta^m_{1,Ny,-1,1}\left(u^m_{0,N_y} + h_y \acute{u}^m_{0,N_y+1}\right) \bigvee \delta^m_{1,Ny,-1,1} u^m_{0,N_y+1}$$

  - $u(t,x^-,y) = u_{(x^-)}(t,y) \bigwedge u(t,x,y^+) = u_{(y^+)}(t,x)$

$$(\mathbf{p}_m)_{1+N_x(Ny-1)} \leftarrow \delta^m_{1,Ny,-1,1} u^m_{0,N_y+1}$$

- Conditions on x^+ and y^-
  - $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^+} = u'_{(x^+)}(t,y) \bigwedge \left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^-} = u'_{(y^-)}(t,x)$

$$\begin{array}{rcl} \lambda^m_{N_x,1,0,0} & \leftarrow & \delta^m_{N_x,1,1,-1} \\ (\mathbf{p}_m)_{N_x} & \leftarrow & \delta^m_{N_x,1,1,-1}\left(h_x \acute{u}^m_{N_x+1,0} - h_y \acute{u}^m_{N_x,0}\right) \\ & \bigvee & \delta^m_{N_x,1,1,-1}\left(h_x \acute{u}^m_{N_x+1,1} - h_y \acute{u}^m_{N_x+1,0}\right) \\ & \bigvee & \frac{1}{2}\delta^m_{N_x,1,1,-1}\left(h_x\left(\acute{u}^m_{N_x+1,0} + \acute{u}^m_{N_x+1,1}\right) - h_y\left(\acute{u}^m_{N_x,0} + \acute{u}^m_{N_x+1,0}\right)\right) \end{array}$$

– $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^+} = u'_{(x^+)}(t,y) \bigwedge u(t,x,y^-) = u_{(y^-)}(t,x)$

$$(\mathbf{p}_m)_{N_x} \leftarrow \delta^m_{N_x,1,1,-1}\left(u^m_{N_x,0} + h_x \acute{u}^m_{N_x+1,0}\right) \bigvee \delta^m_{N_x,1,1,-1} u^m_{N_x+1,0}$$

– $\left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^-} = u'_{(y^-)}(t,x) \bigwedge u(t,x^+,y) = u_{(x^+)}(t,y)$

$$(\mathbf{p}_m)_{N_x} \leftarrow \delta^m_{N_x,1,1,-1}\left(u^m_{N_x+1,1} - h_y \acute{u}^m_{N_x+1,0}\right) \bigvee \delta^m_{N_x,1,1,-1} u^m_{N_x+1,0}$$

– $u(t,x^+,y) = u_{(x^+)}(t,y) \bigwedge u(t,x,y^-) = u_{(y^-)}(t,x)$

$$(\mathbf{p}_m)_{N_x} \leftarrow \delta^m_{N_x,1,1,-1} u^m_{N_x+1,0}$$

- Conditions on x^+ and y^+

– $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^+} = u'_{(x^+)}(t,y) \bigwedge \left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^+} = u'_{(y^+)}(t,x)$

$$\begin{array}{rcl}
\lambda^m_{N_x,N_y,0,0} & \leftarrow & \delta^m_{N_x,N_y,1,1} \\
(\mathbf{p}_m)_{N_xN_y} & \leftarrow & \delta^m_{N_x,N_y,1,1}\left(h_x \acute{u}^m_{N_x+1,N_y+1} + h_y \acute{u}^m_{N_x,N_y+1}\right) \\
 & \bigvee & \delta^m_{N_x,N_y,1,1}\left(h_x \acute{u}^m_{N_x+1,N_y} + h_y \acute{u}^m_{N_x+1,N_y+1}\right) \\
 & \bigvee & \frac{1}{2}\delta^m_{N_x,N_y,1,1}\left(h_x\left(\acute{u}^m_{N_x+1,N_y} + \acute{u}^m_{N_x+1,N_y+1}\right) + h_y\left(\acute{u}^m_{N_x,N_y+1} + \acute{u}^m_{N_x+1,N_y+1}\right)\right)
\end{array}$$

– $\left.\frac{\partial u(t,x,y)}{\partial x}\right|_{x=x^+} = u'_{(x^+)}(t,y) \bigwedge u(t,x,y^+) = u_{(y^+)}(t,x)$

$$(\mathbf{p}_m)_{N_xN_y} \leftarrow \delta^m_{N_x,N_y,1,1}\left(u^m_{N_x,N_y+1} + h_x \acute{u}^m_{N_x+1,N_y+1}\right) \bigvee \delta^m_{N_x,N_y,1,1} u^m_{N_x+1,N_y+1}$$

– $\left.\frac{\partial u(t,x,y)}{\partial y}\right|_{y=y^+} = u'_{(y^+)}(t,x) \bigwedge u(t,x^+,y) = u_{(x^+)}(t,y)$

$$(\mathbf{p}_m)_{N_xN_y} \leftarrow \delta^m_{N_x,N_y,1,1}\left(u^m_{N_x+1,N_y} + h_x \acute{u}^m_{N_x+1,N_y+1}\right) \bigvee \delta^m_{N_x,N_y,1,1} u^m_{N_x+1,N_y+1}$$

– $u(t,x^+,y) = u_{(x^+)}(t,y) \bigwedge u(t,x,y^+) = u_{(y^+)}(t,x)$

$$(\mathbf{p}_m)_{N_xN_y} \leftarrow \delta^m_{N_x,N_y,1,1} u^m_{N_x+1,N_y+1}$$

2.1.3.2 Two specific discretisation schemes. Gourlay and McKee [1977] proposed two specific hopscotch algorithms. For each case, there is a correspondence between the choice of the Θ_m matrix and the definition of the $\delta^m_{i,j,\cdot,\cdot}$ values. In fact, the link is not necessary[4]. We could split this choice into two separate decisions. Then, we have a θ hopscotch method for some choice of the Θ_m matrix and a *filling* hopscotch method following from the $\delta^m_{i,j,\cdot,\cdot}$ definition. In the spirit of the Gourlay's work, we define two specific *filling* methods[5,6] :

[4]But it could be justified in term of algorithm performance.

[5]For all schemes, we have $\delta_x = \delta^0_x$, $\delta_y = \delta^0_y$, $\delta_{xx} = \delta^+_x\delta^-_x$ and $\delta_{yy} = \delta^+_y\delta^-_y$.

[6]We will see later on that the exact choice of the method is important because of computational considerations.

- The left-right method. Suppose that $\delta_{xy} = \frac{1}{2}\left(\delta_x^+ \delta_y^- + \delta_x^- \delta_y^+\right)$, then we may verify that

$$\begin{aligned}
\delta_{i,j,-1,-1}^m &= 0 \\
\delta_{i,j,0,-1}^m &= \frac{b_{i,j}^m}{h_x h_y} + \frac{c_{i,j}^m}{h_y^2} - \frac{e_{i,j}^m}{2h_y} \\
\delta_{i,j,1,-1}^m &= -\frac{b_{i,j}^m}{h_x h_y} \\
\delta_{i,j,-1,0}^m &= \frac{a_{i,j}^m}{h_x^2} + \frac{b_{i,j}^m}{h_x h_y} - \frac{d_{i,j}^m}{2h_x} \\
\delta_{i,j,0,0}^m &= -2\left(\frac{a_{i,j}^m}{h_x^2} + \frac{b_{i,j}^m}{h_x h_y} + \frac{c_{i,j}^m}{h_y^2}\right) \\
\delta_{i,j,1,0}^m &= \frac{a_{i,j}^m}{h_x^2} + \frac{b_{i,j}^m}{h_x h_y} + \frac{d_{i,j}^m}{2h_x} \\
\delta_{i,j,-1,1}^m &= -\frac{b_{i,j}^m}{h_x h_y} \\
\delta_{i,j,0,1}^m &= \frac{b_{i,j}^m}{h_x h_y} + \frac{c_{i,j}^m}{h_y^2} + \frac{e_{i,j}^m}{2h_y} \\
\delta_{i,j,1,1}^m &= 0
\end{aligned}$$

- The center method. If we choose $\delta_{xy} = \delta_x^0 \delta_y^0$, then we have

$$\begin{aligned}
\delta_{i,j,-1,-1}^m &= \frac{b_{i,j}^m}{2h_x h_y} \\
\delta_{i,j,0,-1}^m &= \frac{c_{i,j}^m}{h_y^2} - \frac{e_{i,j}^m}{2h_y} \\
\delta_{i,j,1,-1}^m &= -\frac{b_{i,j}^m}{2h_x h_y} \\
\delta_{i,j,-1,0}^m &= \frac{a_{i,j}^m}{h_x^2} - \frac{d_{i,j}^m}{2h_x} \\
\delta_{i,j,0,0}^m &= -2\left(\frac{a_{i,j}^m}{h_x^2} + \frac{c_{i,j}^m}{h_y^2}\right) \\
\delta_{i,j,1,0}^m &= \frac{a_{i,j}^m}{h_x^2} + \frac{d_{i,j}^m}{2h_x} \\
\delta_{i,j,-1,1}^m &= -\frac{b_{i,j}^m}{2h_x h_y} \\
\delta_{i,j,0,1}^m &= \frac{c_{i,j}^m}{h_y^2} + \frac{e_{i,j}^m}{2h_y} \\
\delta_{i,j,1,1}^m &= \frac{b_{i,j}^m}{2h_x h_y}
\end{aligned}$$

2.1.4 The choice of the θ hopscotch method

Gourlay and McKee [1977] defined two hopscotch methods:

1. the "ordered odd-even" hopscotch scheme

$$\theta_{i,j}^m = \begin{cases} 1 & \text{if} \quad m+i+j \text{ is odd} \\ 0 & \text{if} \quad m+i+j \text{ is even} \end{cases}$$

2. the "line" hopscotch scheme

$$\theta_{i,j}^m = \begin{cases} 1 & \text{if} \quad m+j \text{ is odd} \\ 0 & \text{if} \quad m+j \text{ is even} \end{cases}$$

The ordered odd-even hopscotch was used by Gourlay and MacKee with the left-right method and the line method was associated to the center method. We note that these schemes introduce more sparcity in the system (9). This is not the case when we employ the Crank-Nicholson method ($\theta_{i,j}^m = \frac{1}{2}$) or the usual θ-scheme

$$\theta_{i,j}^m = 1 - \theta \text{ and } \theta_{i,j}^{m+1} = \theta$$

2.2 Stability

Because Hopscotch methods are a special case of general θ-schemes, these methods satisfy the following proposition:

Proposition 1 *The stability assumption*[7] *is verified if*

$$k \to 0 \bigwedge h \to 0 \bigwedge \frac{k}{h^2} \to 0 \tag{22}$$

for h equal to h_x and h_y.

It is difficult to demonstrate this proposition for the general problem but Gourlay [1971] shows that the Hopscotch algorithm may be regarded as an A.D.I. method. Using this observation, he demonstrated that the algorithm is stable and converges under weak conditions on H_m matrix and on the mesh ratios $\frac{k}{h^2}$.

We can illustrate the stability issue with an example. Consider the linear parabolic PDE system defined by

$$\begin{aligned}
a(t,x,y) &= \frac{1}{2}x^2 + y^2 \\
b(t,x,y) &= -\frac{1}{2}\left(x^2 + y^2\right) \\
c(t,x,y) &= x^2 + \frac{1}{2}y^2 \\
d(t,x,y) &= x \\
e(t,x,y) &= -y \\
f(t,x,y) &= 1 \\
g(t,x,y) &= 2xy^2 e^{-t}
\end{aligned}$$

$\mathfrak{R}$ is set to $[0,1] \times [0,1]$ and we have

$$\begin{aligned}
u(t,0,y) &= 0 \\
u(t,1,y) &= \left(y + y^2\right) e^{-t} + 1 \\
u(t,x,0) &= x \\
u(t,x,1) &= \left(x + x^2\right) e^{-t} + x
\end{aligned}$$

The solution of the cauchy problem with $u(0,x,y) = x^2 y + xy^2 + x$ is

$$u(t,x,y) = \left(x^2 y + xy^2\right) e^{-t} + x$$

We have solved this for t^+ equal to 5 by considering a left-right filling method and an ordered odd-even method. We consider three different cases:

[7] see for example Thomée [1990].

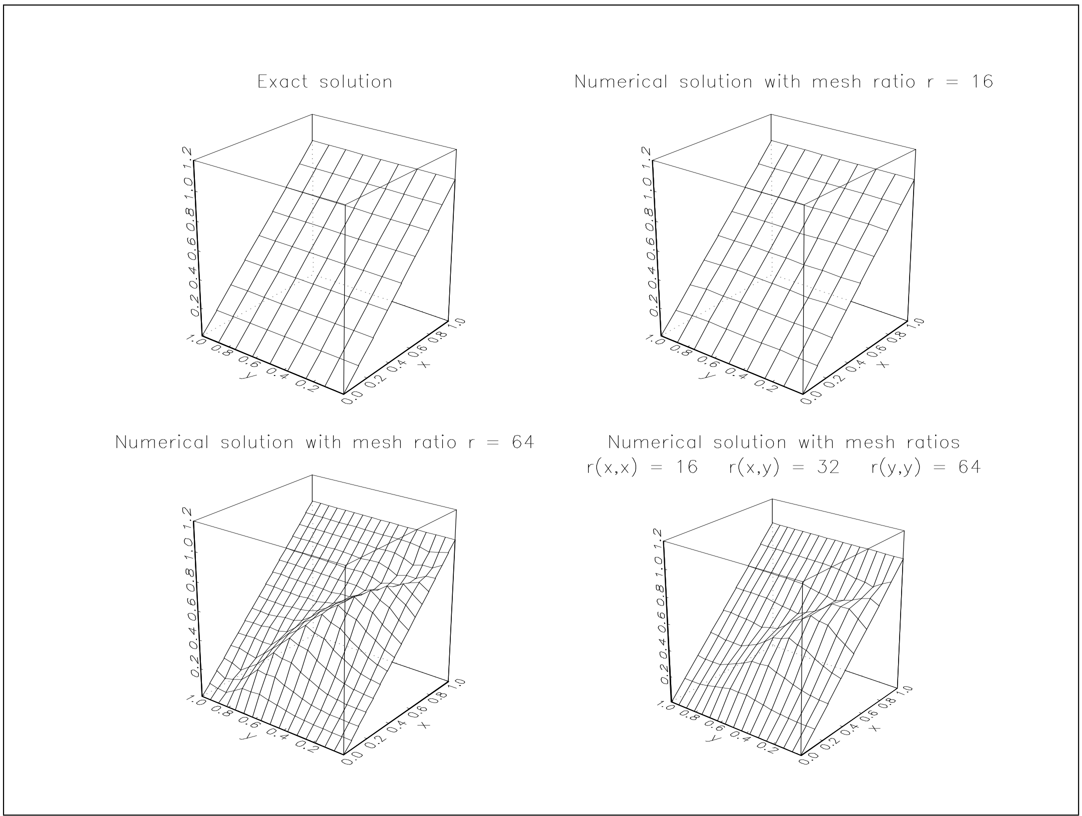

Figure 1: Illustration of the stability property

	k	h_x	h_y	$r_{x,x}$	$r_{x,y}$	$r_{y,y}$
(a)	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	16	16	16
(b)	$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{16}$	64	64	64
(c)	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	16	32	64

We report the numerical solutions $u(t^+, x, y)$ in the figure 1. It is important to note that stability depends on the values of the **three** mesh ratios, not only on the value of the central mesh ratio $r_{x,y}$. For (b) and (c), the algorithm is instable and produces bad solutions. This issue is clearly important, because if we increase the mesh **spacings** in x and y space, we also have to increase the mesh **spacing** in time space. In this situation, it is important to work with constant mesh **ratios**.

Let the mesh ratios be constant. If the algorithm is stable, then convergence is obtained if $k \to 0$. Experience shows in fact that $\exists \left(\bar{h}_x, \bar{h}_y\right) \in \mathbb{R}^2_+$ such that for $h_x \leqslant \bar{h}_x$ and $h_y \leqslant \bar{h}_y$, it is not possible to decrease the numerical error. That is why the most important parameter is k.

If we consider the numerical error for the central node $x = \frac{1}{2}$ and $y = \frac{1}{2}$ for the previous problem with $h_x = h_y = \frac{1}{10}$ we can see clearly that the numerical error decreases[8] with the number of steps N_t (figure 2).

[8]We note however that the decreases is not necessary monotone.

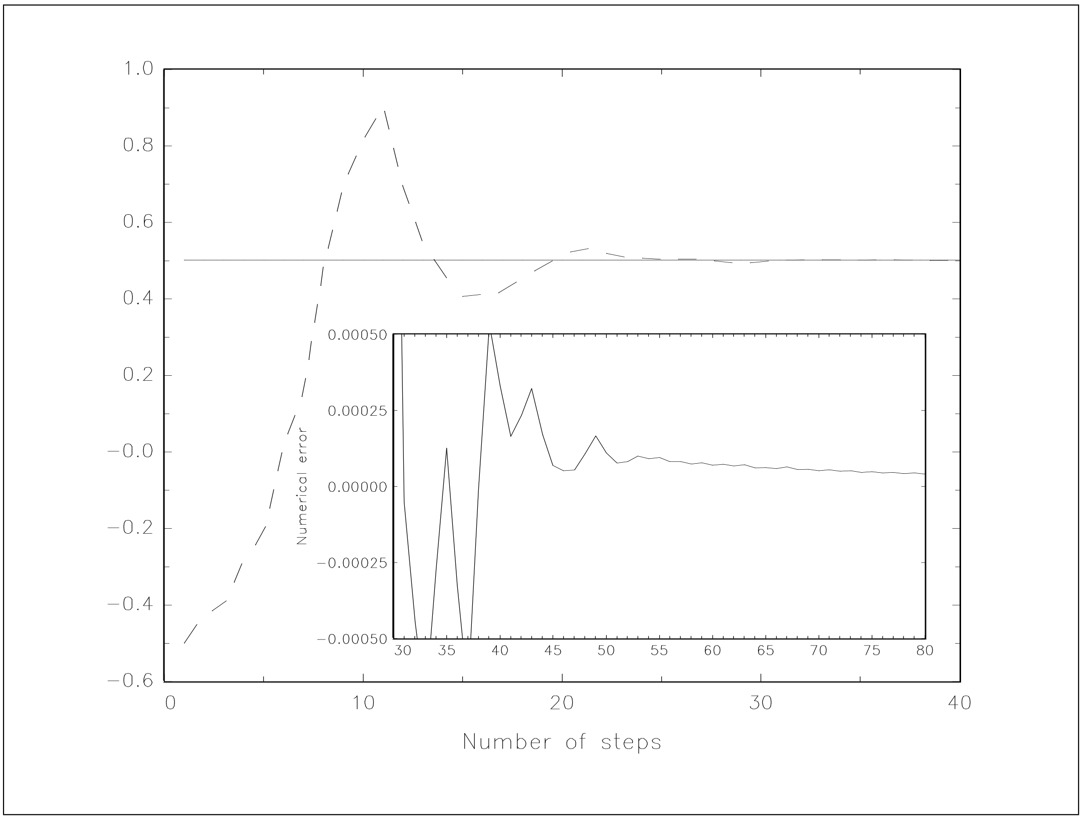

Figure 2: Illustration of the accuracy problem

2.3 Computational considerations

We note that equation (9) is of the form

$$\Psi_{m+1} u_{m+1} = \boldsymbol{\phi}_{m+1} \tag{23}$$

where Ψ is **both** a band and sparse matrix. The method of solving (9) exploits both these properties. First, it is more efficient to work with the band form for matrix operations and then transform the band system into a sparse system.

2.3.1 Efficient computation

We adopt a different version of Golub and Van Loan [1989,page 21] for the band storage of the matrices (see appendix A). Let band be the process which transform the band matrix Ψ into a band storage matrix. We have

$$\Psi.band = \text{band}\,(\Psi)$$

The table below shows the importance of the band storage in term of memory management.

$N_x = N_y$	10	20	50	100	1000
rows (Ψ)	100	400	2500	10000	1000000
cols (Ψ)	100	400	2500	10000	1000000
cells (Ψ)	10^4	1.6×10^5	6.25×10^6	10^8	10^{12}
mem (Ψ)	80 Kbytes	1.28 Mbytes	50 Mbytes	800 Mbytes	8000 Gbytes
rows $(\Psi.band)$	100	400	2500	10000	1000000
cols $(\Psi.band)$	9	9	9	9	9
cells $(\Psi.band)$	900	3600	2.25×10^4	9×10^4	9×10^6
mem $(\Psi.band)$	7.2 Kbytes	28.8 Kbytes	180 Kbytes	720 Kbytes	72 Mbytes

For differents values of N_x and N_y, we have reported the number of rows, columns, cells of the matrix Ψ for the dense and band forms. We also report the memory required to store these two matrices. For example, for $N_x = N_y = 100$, we need 800 Mbytes to store the dense form matrix and it requires 720 Kbytes for the band form matrix.

The band form is not only useful from the memory management point of view but it also facilitates the computation of the matrix Ψ_{m+1} and the vector $\boldsymbol{\phi}_{m+1}$ because we can use the following algorithms that are more efficient[9] than the corresponding matrix operations:

1. Notice that we can replace matrix-vector multiplication with the Hadamard product. For instance we have
$$V_1 \mathbf{v}_2 := \operatorname{diag}(\mathbf{v}_1)\,\mathbf{v}_2 = \mathbf{v}_1 \odot \mathbf{v}_2$$

2. We can replace the matrix addition $\Psi = \Psi_1 + \Psi_2$ by

   for $i = 1 : N_x N_y$
       for $j = 1 : 9$
           $\Psi.band\,(i,j) = \Psi_1.band\,(i,j) + \Psi_2.band\,(i,j)$
       end
   end

3. We can replace the matrix addition $\Upsilon = \Psi + \operatorname{diag}(\mathbf{v})$ by

   for $i = 1 : N_x N_y$
       for $j = 1 : 9$
           if $j = 5$
               $\Upsilon.band\,(i,j) = \Psi.band\,(i,j) + \mathbf{v}\,(i)$
           else
               $\Upsilon.band\,(i,j) = \Psi.band\,(i,j)$
           end
       end
   end

4. We could replace the scalar-matrix multiplication $\Upsilon = \alpha\Psi$ by

   for $i = 1 : N_x N_y$
       for $j = 1 : 9$
           $\Upsilon.band\,(i,j) = \alpha\Psi.band\,(i,j)$
       end
   end

[9] In dense form, matrix operation rules are N^2 process. With these algorithm, they becomes $9N$ process.

5. We could replace the matrix multiplication $\Upsilon = \operatorname{diag}(\mathbf{v})\,\Psi$ by

$$\begin{array}{l}
\textbf{for } i = 1 : N_x N_y \\
\quad \textbf{for } j = 1 : 9 \\
\quad\quad \Upsilon.band\,(i,j) = \mathbf{v}\,(i)\,\Psi.band\,(i,j) \\
\quad \textbf{end} \\
\textbf{end}
\end{array}$$

6. and finally we can replace the matrix-vector multiplication $\mathbf{v}_2 = \Psi \mathbf{v}_1$ by

$$\begin{array}{l}
\textbf{for } i = 1 : N_x N_y \\
\quad \mathbf{v}_2\,(i) = \Psi.band\,(i,1)\,\mathbf{v}_1\,(i - N_x - 1) + \Psi.band\,(i,2)\,\mathbf{v}_1\,(i - N_x) + \\
\quad\quad\quad \Psi.band\,(i,3)\,\mathbf{v}_1\,(i - N_x + 1) + \Psi.band\,(i,4)\,\mathbf{v}_1\,(i - 1) + \\
\quad\quad\quad \Psi.band\,(i,5)\,\mathbf{v}_1\,(i) + \Psi.band\,(i,6)\,\mathbf{v}_1\,(i + 1) + \\
\quad\quad\quad \Psi.band\,(i,7)\,\mathbf{v}_1\,(i + N_x - 1) + \Psi.band\,(i,8)\,\mathbf{v}_1\,(i + N_x) + \\
\quad\quad\quad \Psi.band\,(i,9)\,\mathbf{v}_1\,(i + N_x + 1) \\
\quad \textbf{with} \\
\quad\quad \mathbf{v}_1\,(k) = 0 \qquad \textbf{if } k < 1 \textbf{or } k > N_x N_y \\
\textbf{end}
\end{array}$$

2.3.2 Methods for solving sparse systems

The system (23) could of course be solved by an exact non-symmetric band algorithm. But this method is not computationally efficient. It is better to use sparse methods. In figure 3, we draw the sparse representation of the Ψ matrices. We notice that Hopscotch schemes introduce more sparcity into the system (9). So, the most efficient way to solve this problem is certainly to use iterative methods (for example Richardson or Conjugate Gradient methods). These iterative algorithms are not exact, but converges very quickly in practice. Moreover, we can use the vector u_m for the initial estimate of the solution. In this case, we replace the problem (23) by the following

$$\Psi_{m+1} v_{m+1} = \Upsilon_{m+1} \tag{24}$$

with $\Upsilon_{m+1} := \boldsymbol{\phi}_{m+1} - \Psi_{m+1} u_m$. The solution is also given by $u_{m+1} = u_m + v_{m+1}$.

3 Application to two-state financial models

In this section, we apply the Hopscotch methods described above to two-state variable financial models. First, we present the fundamental equation in finance and show that it corresponds to the problem set up in section two. Then, we consider particular cases: option pricing, term structure modelling and financial elliptic problems. Note that for all these problems we use the ordered odd-even method with a left-right center discretisation scheme. In most cases, it is less accurate, but it is faster.

3.1 General framework for contingent claims valuation

We make the following assumptions :

1. The market permits continuous and frictionless trading. Moreover, the market is complete and no arbitrage opportunities exist.

2. The price of the financial asset $P(t)$ is completely determined by the vector $X(t)$ of the M state variables. We have

$$P(t) = P(t, X(t)) \tag{25}$$

3. The M-dimensional state vector $X(t)$ is a diffusion process defined by the following Stochastic Differential Equation

$$\left\{ \begin{array}{rcl} dX(t) & = & \mu(t, X(t))\, dt + \Sigma(t, X(t))\, dW(t) \\ X(t_0) & = & X_0 \end{array} \right. \tag{26}$$

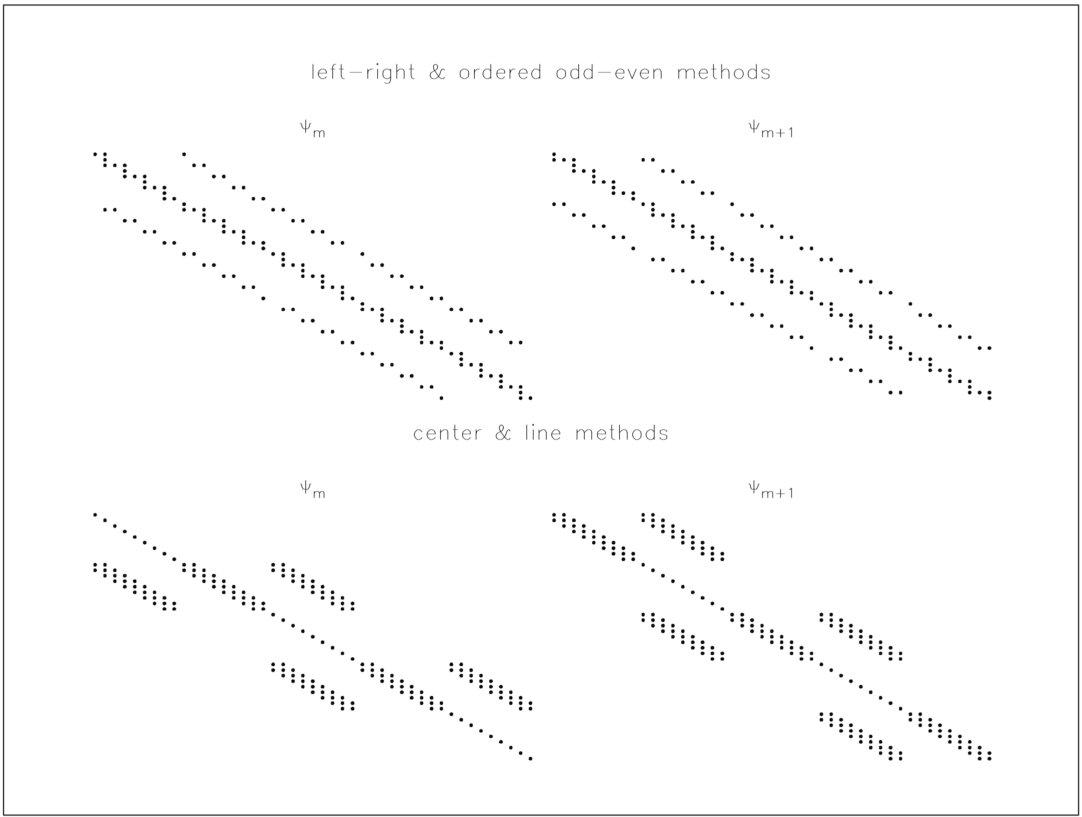

Figure 3: Sparse representation of the Ψ matrices

where $W(t)$ is a N-dimensional Wiener process defined on the fundamental probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with the covariance matrix

$$E\left[W(t)\, W(t)^{\top}\right] = \rho t \tag{27}$$

4. There is a risk-free asset whose return r depends on the state variables $X(t)$. So we have

$$r = r(t, X(t)) \tag{28}$$

5. The maturity date of the asset is T. The delivery value B depends on the values taken by the state variables at the maturity date

$$B = P(T) = B(T, X(T)) \tag{29}$$

and the asset pays a continuous dividend b which is a function of the state vector

$$b = b(t, X(t)) \tag{30}$$

Theorem 2 *In the M-factor arbitrage model which satisfies the previous assumptions, the price of the financial asset $P(t)$ satisfies the following Partial Differential Equation*

$$\left\{ \begin{array}{l} \frac{1}{2}\operatorname{trace}\left(\Sigma(t,X)^{\top} P_{XX}(t,X)\,\Sigma(t,X)\,\rho\right) \\ +\left[\mu(t,X)^{\top} - \lambda(t,X)^{\top}\,\Sigma(t,X)^{\top}\right] P_X(t,X) \\ +P_t(t,X) - r(t,X)\,P(t,X) + b(t,X) = 0 \\ P(T) = B(T, X(T)) \end{array} \right. \tag{31}$$

Most of the two-state variable models impose $N = 2$. In this case, equation (26) becomes

$$\begin{bmatrix} dX_1(t) \\ dX_2(t) \end{bmatrix} = \begin{bmatrix} \mu_1(t, X_1, X_2) \\ \mu_2(t, X_1, X_2) \end{bmatrix} dt + \begin{bmatrix} \sigma_{1,1}(t, X_1, X_2) & \sigma_{1,2}(t, X_1, X_2) \\ \sigma_{2,1}(t, X_1, X_2) & \sigma_{2,2}(t, X_1, X_2) \end{bmatrix} \begin{bmatrix} dW_1(t) \\ dW_2(t) \end{bmatrix} \tag{32}$$

with

$$\rho = \begin{bmatrix} 1 & \rho_{1,2} \\ & 1 \end{bmatrix} \tag{33}$$

Then, the fundamental equation takes the following form

$$\begin{aligned} \left[\frac{1}{2}\sigma_{1,1}^2 + \rho_{1,2}\sigma_{1,1}\sigma_{1,2} + \frac{1}{2}\sigma_{1,2}^2\right] P_{X_1,X_1} + \left[\frac{1}{2}\sigma_{2,1}^2 + \rho_{1,2}\sigma_{2,1}\sigma_{2,2} + \frac{1}{2}\sigma_{2,2}^2\right] P_{X_2,X_2} & \\ + \left[\sigma_{1,1}\sigma_{2,1} + \rho_{1,2}\sigma_{1,1}\sigma_{2,2} + \sigma_{1,2}\sigma_{2,2} + \rho_{1,2}\sigma_{1,2}\sigma_{2,1}\right] P_{X_1,X_2} & \\ + \left[\mu_1 - \lambda_1\sigma_{1,1} - \lambda_2\sigma_{1,2}\right] P_{X_1} + \left[\mu_2 - \lambda_1\sigma_{2,1} - \lambda_2\sigma_{2,2}\right] P_{X2} & \\ + P_t - rP + b &= 0 \end{aligned} \tag{34}$$

Let $\tau = T - t$ be the time to maturity of the asset. We see that equation (34) could be put in the form (1). In this case, τ takes the role of the variable t and X_1 and X_2 correspond to the x and y variables. We have

$$\begin{aligned} a(\tau, X_1, X_2) &= \frac{1}{2}\sigma_{1,1}^2 + \rho_{1,2}\sigma_{1,1}\sigma_{1,2} + \frac{1}{2}\sigma_{1,2}^2 \\ b(\tau, X_1, X_2) &= \frac{1}{2}\left(\sigma_{1,1}\sigma_{2,1} + \rho_{1,2}\sigma_{1,1}\sigma_{2,2} + \sigma_{1,2}\sigma_{2,2} + \rho_{1,2}\sigma_{1,2}\sigma_{2,1}\right) \\ c(\tau, X_1, X_2) &= \frac{1}{2}\sigma_{2,1}^2 + \rho_{1,2}\sigma_{2,1}\sigma_{2,2} + \frac{1}{2}\sigma_{2,2}^2 \\ d(\tau, X_1, X_2) &= \mu_1 - \lambda_1\sigma_{1,1} - \lambda_2\sigma_{1,2} \\ e(\tau, X_1, X_2) &= \mu_2 - \lambda_1\sigma_{2,1} - \lambda_2\sigma_{2,2} \\ f(\tau, X_1, X_2) &= r(T - \tau, X_1, X_2) \\ g(\tau, X_1, X_2) &= b(T - \tau, X_1, X_2) \\ u(0, X_1, X_2) &= B(T, X_1, X_2) \end{aligned}$$

3.2 Option pricing

This research into numerical methods was in fact driven by a desire to solve Stochastic Volatility option problems. In particular, we wanted to analyse the impact of Stochastic Volatility on American options (see Kurpiel and Roncalli [1998a]).

3.2.1 Black-Scholes models

Let K and τ be the exercise price and the time to maturity of an European option on the underlying asset price $S(t)$. In the Black-Scholes framework, the call option price $C(\tau, S)$ satisfies the following equation

$$\begin{cases} \frac{1}{2}\sigma^2 S^2 C_{SS} + bC_S = C_\tau + rC \\ C(0, S) = (S - K)_+ \end{cases} \tag{35}$$

The parameter b is the cost-of-carry rate[10]. To solve this problem numerically using Hopscotch methods, we have to add two boundary conditions for the extreme values S^- and S^+ taken by the S variable. For S equal to S^-, we chose the following condition

$$u\left(t, S^-, y\right) = 0 \tag{36}$$

because the option price tends to be zero when the underlying asset price decreases (out-of-the-money call options). For S equal to S^+, we choose between three boundary conditions:

[10]For a currency option, b is equal to the differential interest rate $r - r^\star$ (Garman and Kohlhagen [1993]), for an option on futures, b is set to 0 (Black [1976]), and for an option on a dividend paying stock, b corresponds to the difference between the instantaneous interest rate and the annual dividend yield d.

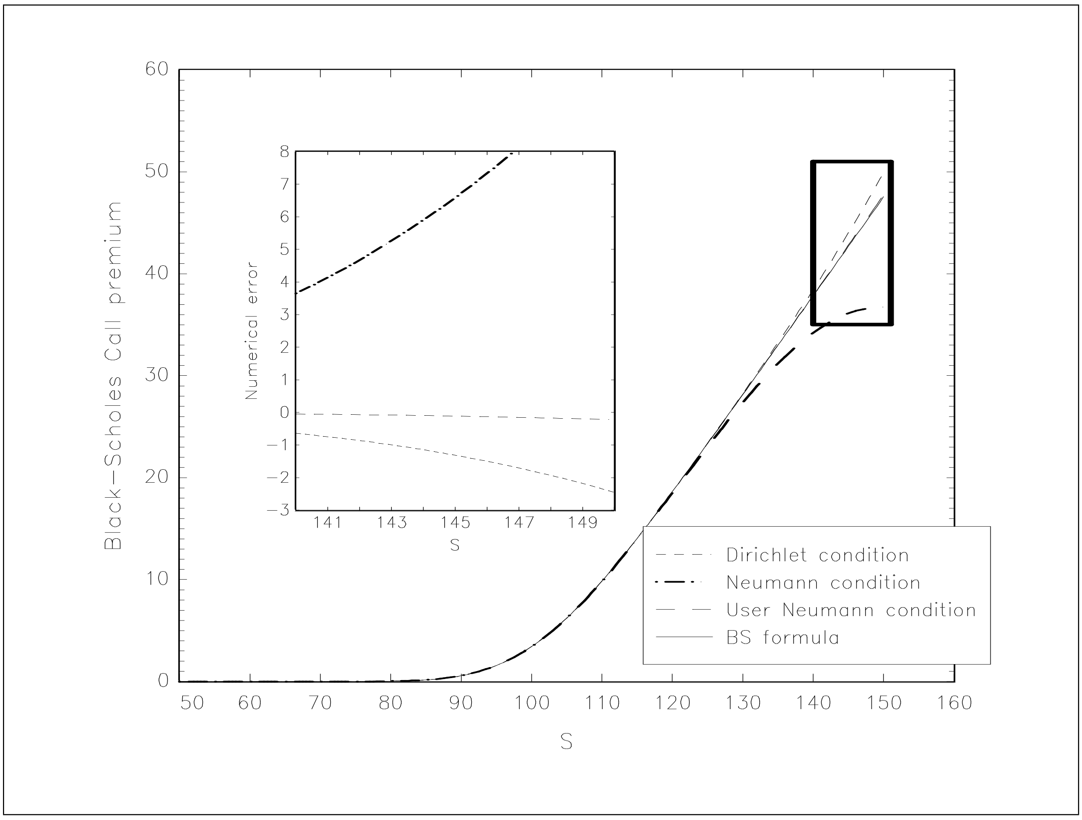

Figure 4: Influence of boundary conditions

1. We impose a Dirichlet condition
$$u\left(t,S^{+},y\right)=S^{+}-K \tag{37}$$
We can use this boundary condition because of the nature of in-the-money options. When the underlying asset price increases, the time value of the option decreases and the intrinsic value increases and the time value tends to 0 when S tends to $+\infty$.

2. We consider the usual Neumann condition
$$u_S\left(t,S^{+},y\right)=0 \tag{38}$$
This boundary condition is often used in numerical analysis.

3. We could also choose the following user-defined Neumann condition
$$u_S\left(t,S^{+},y\right)=1 \tag{39}$$
The argument is pratically the same as for the first choice.

Consider an option with the following parameters $K=100$, $\tau=0.25$, $\sigma=0.20$, $r=0.08$ and $b=-0.04$. We take $S^{-}=50$ and $S^{+}=150$. We set the mesh spacing in S equal to 0.5 and the mesh spacing in τ equal to $\frac{1}{1825}$, that is approximately $\frac{1}{5}$ day. The figure 4 illustrates the solution for the different boundary conditions. We can clearly see how the choice influences the solution and we note that a bad choice of the boundary condition clearly produces poor results. However it is important to notice that the main errors are to be found near the boundary region, not in the central part of the domain for S. For example, we obtain the following values for the centered nodes

S	$u(t, S^+, y) = S^+ - K$	$u_S(t, S^+, y) = 0$	$u_S(t, S^+, y) = 1$	"True value"
95	1.5701527	1.5701457	1.5701523	1.569
100	3.4227318	3.4226425	3.4227264	3.421
105	6.2218521	6.2210601	6.2217987	6.220

In this example, the choice of boundary condition has little effect on the centered nodes. This is very important for financial modelling since in many cases, we do not know four boundary conditions. Sometimes, a simple guess is used as a prior for a boundary condition. This example shows that we may however use "incorrect" boundary conditions and still consider numerical solutions in the central region of $\mathfrak{R}$. Of course, we must be careful and we have to verify the behaviour of the numerical solution when we change the boundary function.

The American case is interesting, because we do not know of any other example of numerical American option pricing with stochastic volatility. Before we apply the algorithm to this problem, we show how to modify the Hopscotch algorithm in order to take into account the special nature of the American option. For an American option, we have to verify that

$$C(\tau, S) \geqslant (S - K)_+ \tag{40}$$

for each value of τ. In this case, the problem has no Feynman-Kac representation, but becomes a variational inequalities problem. Lamberton and Lapeyre [1997] show that it could be solved by finite difference methods. At each iteration m, the solution $u_{i,j}^{m+1}$ given by the equation (9) is replaced by the following value

$$\max\left(u_{i,j}^{m+1}, (S_i - K)_+\right) \tag{41}$$

The intuition is that the intrinsic option value must be the payoff of the option, because the seller of the option could exercise at any moment.

The table below reproduces the results of the table I (call options) and the table II (put options) of Barone-Adesi and Whaley [1987]. We report values for the European (eu) and American (am) cases. In this last case, we used the Barone-Adesi-Whaley quadratic approximation (baw), the Hopscotch method (h) and the implicit Finite differences methods (fd) to compute the option prices[11]. For the Hopscotch method, we use the same parameters as above. The values of the put options for $b = -0.04$ and the call options for $b = 0.04$ are not reported because these two cases are not very interesting (we could show that the American price is the same as the European price). The Hopscotch American option prices are very close to the American options prices computed by the Barone-Adesi and Whaley (FD method).

[11] The values for the fd method are those calculated by Barone-Adesi and Whaley [1987].

		Call options ($b=-0.04$)				Put options ($b=0.04$)			
Options parameters	S_0	C_{EU}	$C_{\mathrm{AM}}^{\mathrm{BAW}}$	$C_{\mathrm{AM}}^{\mathrm{H}}$	$C_{\mathrm{AM}}^{\mathrm{FD}}$	P_{EU}	$P_{\mathrm{AM}}^{\mathrm{BAW}}$	$P_{\mathrm{AM}}^{\mathrm{H}}$	$P_{\mathrm{AM}}^{\mathrm{FD}}$
	80	0.029	0.032	0.029	0.03	18.868	20	20	20
$r=0.08$	90	0.57	0.59	0.579	0.58	9.765	10.183	10.223	10.22
$\sigma=0.2$	100	3.421	3.525	3.524	3.52	3.455	3.544	3.547	3.55
$\tau=0.25$	110	9.847	10.315	10.356	10.35	0.777	0.798	0.788	0.79
	120	18.618	20	20	20	0.112	0.118	0.113	0.11
	80	0.029	0.032	0.029	0.03	18.680	20	20	20
$r=0.12$	90	0.564	0.587	0.574	0.58	9.667	10.161	10.197	10.20
$\sigma=0.2$	100	3.387	3.506	3.501	3.5	3.421	3.525	3.523	3.52
$\tau=0.25$	110	9.749	10.288	10.326	10.32	0.769	0.794	0.782	0.78
	120	18.433	20	20	20	0.111	0.118	0.112	0.11
	80	1.046	1.067	1.052	1.06	20.105	20.528	20.586	20.59
$r=0.08$	90	3.232	3.284	3.269	3.27	12.738	12.297	12.957	12.95
$\sigma=0.4$	100	7.291	7.411	7.41	7.4	7.364	7.456	7.458	7.46
$\tau=0.25$	110	13.248	13.502	13.53	13.52	3.910	3.958	3.944	3.95
	120	20.728	21.233	21.298	21.29	1.927	1.954	1.931	1.94
	80	0.21	0.229	0.214	0.21	18.077	20	20	20
$r=0.08$	90	1.312	1.387	1.359	1.36	10.041	10.706	10.756	10.75
$\sigma=0.2$	100	4.465	4.724	4.709	4.71	4.555	4.772	4.767	4.77
$\tau=0.50$	110	10.163	10.955	10.998	11.00	1.681	1.760	1.736	1.74
	120	17.851	20	20	20	0.514	0.546	0.526	0.53

3.2.2 Stochastic volatility option models

A general stochastic volatility option model is defined by the following diffusion equations for the state variables

$$\left\{\begin{array}{rcl} dS(t) & = & \mu_S(t,S(t))\,dt+\Sigma_S(t,S(t),V(t))\,dW_1(t) \\ dV(t) & = & \mu_V(t,V(t))\,dt+\Sigma_V(t,S(t),V(t))\,dW_2(t) \end{array}\right. \tag{42}$$

with

$$E\left[W_1(t)\,W_2(t)\right]=\rho t \tag{43}$$

For example, Hull and White [1987] assume that

$$\left[\begin{array}{c} dS(t) \\ dV(t) \end{array}\right]=\left[\begin{array}{c} \mu_S S(t) \\ \mu_V V(t) \end{array}\right]dt+\left[\begin{array}{cc} \sqrt{V(t)}S(t) & 0 \\ 0 & \sigma_V V(t) \end{array}\right]\left[\begin{array}{c} dW_1(t) \\ dW_2(t) \end{array}\right] \tag{44}$$

Wiggins [1987] uses a similar model where the trend function of the $V(t)$ process is not $\mu_V V(t)$ but a general function $f(V(t))$ of the second state variable. The model used by Heston [1973] is defined by the following EDS

$$\left[\begin{array}{c} dS(t) \\ dV(t) \end{array}\right]=\left[\begin{array}{c} \mu S(t) \\ \kappa(\theta-V(t)) \end{array}\right]dt+\left[\begin{array}{cc} \sqrt{V(t)}S(t) & 0 \\ 0 & \sigma_V\sqrt{V(t)} \end{array}\right]\left[\begin{array}{c} dW_1(t) \\ dW_2(t) \end{array}\right] \tag{45}$$

The dynamic of the underlying asset is very close to the geometric brownian motion used by Black and Scholes [1973], except that the volatility is not constant but stochastic. Heston chooses the "square root process" introduced by Cox, Ingersoll and Ross [1985b]. This process is very close to the Ornstein-Uhlenbeck process, but the diffusion function is not constant and equals $\sigma_V^2 V(t)$. For solving the European option case, Heston uses characteristic function techniques with the following market prices

$$\lambda_1(t,S,V)=\frac{\mu-r}{\sqrt{V}} \tag{46}$$

and

$$\lambda_2(t,S,V)=\frac{\lambda}{\sigma_V}\sqrt{V} \tag{47}$$

The market price of the first risk (that is the first Wiener process) is the same as considered by Black and Scholes. It could be easily found using an asset duplication argument. For the second market price, Heston follows Cox, Ingersoll and Ross [1995] and assumes the form (47).

$\boldsymbol{\rho = -0.5}$					
S_0	C_{EU}	$C_{\mathrm{EU,SV}}^{\mathrm{He}}$	$C_{\mathrm{EU,SV}}^{\mathrm{H}}$	$C_{\mathrm{AM}}^{\mathrm{BAW}}$	$C_{\mathrm{AM,SV}}^{\mathrm{H}}$
80	0.0291	0.0169	0.0168	0.0322	0.0170
85	0.1545	0.1170	0.1166	0.1624	0.1185
90	0.5700	0.5040	0.5034	0.5896	0.5133
95	1.5689	1.5030	1.5026	1.6151	1.5401
100	3.4211	3.3961	3.3958	3.5249	3.5038
105	6.2204	6.2496	6.2492	6.4448	6.5016
110	9.8470	9.9094	9.9087	10.3146	10.4123
115	14.0596	14.1251	14.1242	15	15.0156
120	18.6180	18.6682	18.6675	20	20

$\boldsymbol{\rho = 0.5}$					
S_0	C_{EU}	$C_{\mathrm{EU,SV}}^{\mathrm{He}}$	$C_{\mathrm{EU,SV}}^{\mathrm{H}}$	$C_{\mathrm{AM}}^{\mathrm{BAW}}$	$C_{\mathrm{AM,SV}}^{\mathrm{H}}$
80	0.0291	0.0484	0.0482	0.0322	0.0487
85	0.1545	0.1992	0.1989	0.1624	0.2017
90	0.5700	0.6354	0.6354	0.5896	0.6465
95	1.5689	1.6240	1.6241	1.6151	1.6606
100	3.4211	3.4298	3.4298	3.5249	3.5319
105	6.2204	6.1768	6.1763	6.4448	6.4223
110	9.8470	9.7773	9.7764	10.3146	10.3006
115	14.0596	13.9958	13.9950	15	15
120	18.6180	18.5756	18.5756	20	20

The tables above have been obtained with the following parameters:

b	r	κ	θ	σ_V	λ
-0.04	0.08	0.9	0.04	0.10	0.0

We report in the tables the values of an option defined by $K = 100$ and $\tau = 0.25$. We assume that V_0 is equal to 0.04. For the Hopscotch method, we use the following parameters $S^- = 50$, $S^+ = 150$, $V^- = 0.002$ and $V^+ = 0.122$. For the mesh spacing in S, V and τ, we take the following values 0.5, 0.002 and $\frac{1}{1835}$. $C_{\mathrm{EU,SV}}^{\mathrm{He}}$ corresponds to the value given by the Heston formula, $C_{\mathrm{EU,SV}}^{\mathrm{H}}$ and $C_{\mathrm{AM,SV}}^{\mathrm{H}}$ are the values for European and American options obtained with the Hopscotch method. Note that the boundary conditions for V are[12]

$$C_V\left(\tau, S, V^-\right) = C_V\left(\tau, S, V^+\right) = 0 \tag{48}$$

We note that the closed formula of Heston gives very accurate results. In general, the difference between the Heston solution and the Hopscotch method occurs after the second digit. We also stress the difference between the negative ρ case and the positive ρ case in our example, because of the impact of this parameter on in-the-money and out-of-the-money options. Of course, we could have used other specifications for the stochastic volatility process and solved the problem with Hopscotch methods when we are unable to compute the **analytic** solution.

Hull and White suggest the use of Monte Carlo methods to solve this type of problem. A first difficulty is that they can't be used for the American option case. The second problem is more important and is that option models are not just used for pricing. In practice, they are used for computing the greeks and for derivatives hedging (Kurpiel and Roncalli [1998b]). Monte Carlo mehtods are not stable enough for these

[12] For a discussion of the choice of boundary conditions, see Kurpiel and Roncalli [1998a].

computations as the results depend critically on the simulation paths. This problem does not arise with Hopscotch methods. For example, we can approximate, with good degree of accuracy, the delta, gamma, theta and vega coefficients using the following formulas

$$\begin{aligned}
\Delta\left(S=S_i,V=V_j,\tau=\tau_m\right) &= \frac{u_{i+1,j}^m-u_{i-1,j}^m}{2h_S} && (49)\\
\Gamma\left(S=S_i,V=V_j,\tau=\tau_m\right) &= \frac{u_{i+1,j}^m-2u_{i,j}^m+u_{i-1,j}^m}{h_S^2} && (50)\\
\Theta\left(S=S_i,V=V_j,\tau=\tau_m\right) &= \frac{u_{i,j}^{m+1}-u_{i,j}^{m-1}}{2k} && (51)\\
\vartheta\left(S=S_i,V=V_j,\tau=\tau_m\right) &= \frac{u_{i,j+1}^m-u_{i,j-1}^m}{2h_V} && (52)
\end{aligned}$$

The third problem with Monte Carlo methods concerns the implied volatility. With MC methods, we have to employ a Newton-Raphson procedure with a numerical gradient, but we have found again that this doesn't produce accurate results. With Hopscotch methods, on the other hand, you just have to search after solving the Hopscotch problem with a sort algorithm. In this case, we don't have to invert the pricing formula[13].

3.3 Term structure modelling

2D term structure models are generally based on the following model

$$\begin{bmatrix} dr(t) \\ d\chi(t) \end{bmatrix} = \begin{bmatrix} \mu_r\left(t,r(t),\chi(t)\right) \\ \mu_\chi\left(t,r(t),\chi(t)\right) \end{bmatrix} dt + \begin{bmatrix} \sigma_r\left(t,r(t),\chi(t)\right) & 0 \\ 0 & \sigma_\chi\left(t,r(t),\chi(t)\right) \end{bmatrix} \begin{bmatrix} dW_1(t) \\ dW_2(t) \end{bmatrix} \tag{53}$$

with

$$E\left[W_1(t)\,W_2(t)\right]=\rho t \tag{54}$$

For example, Longstaff and Schwartz [1992] use the instantaneous volatility in the second state. Other models are based on the model of vasicek [1977]. He assumed that the instantaneous interest rate is an Ornstein-Uhlenbeck process

$$dr(t)=a\left(b-r(t)\right)dt+\sigma\,dW(t) \tag{55}$$

This Vasicek model has stimulated a number of extensions. For example, we could introduce a stochastic mean reversion

$$\left\{\begin{aligned} dr(t) &= a\left(b(t)-r(t)\right)dt+\sigma_r\left(r(t)\right)dW_1(t)\\ db(t) &= \mu_b\left(r(t),b(t)\right)dt+\sigma_b\left(b(t)\right)dW_2(t)\end{aligned}\right. \tag{56}$$

Brennan and Schwartz suggest to use a long rate $l(t)$ as the second state variable. For example, we could choose to model the term structure with the following EDS

$$\left\{\begin{aligned} dr(t) &= \left[a_r\left(b_r-r(t)\right)+\alpha\left(b_l-l(t)\right)\right]dt+\sigma_r\,dW_1(t)\\ dl(t) &= a_l\left(b_l-l(t)\right)dt+\sigma_l\,dW_2(t)\end{aligned}\right. \tag{57}$$

Figure 5 shows the impact of introducing this second state variable in the Vasicek model. We have used the following values

a_r	b_r	σ_r	λ_r
1.3	0.15	0.25	−0.3
a_l	b_l	σ_l	λ_l
0.8	0.10	0.15	0

ρ was set equal to $\frac{1}{2}$. In the figure, we show the solution for different values of α, a_l and b_l where r and l are equal to 0.15. In order to solve the problem with Hopscotch methods, we use the following parameters

$r^-=l^-$	$r^+=l^+$	$h_r=h_l$	k
0.000	0.300	0.01	$\frac{1}{1825}$

[13] Readers will find examples and results on smile curve with SV options in Kurpiel and Roncalli [1998a,1998b].

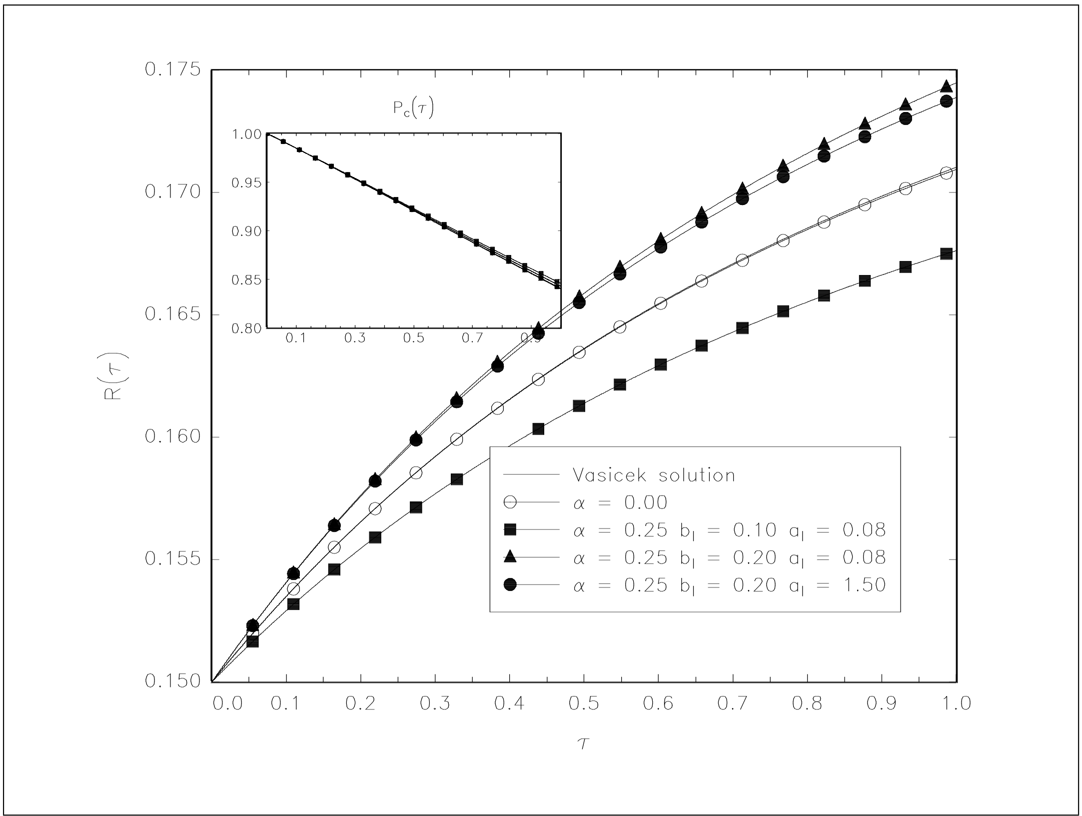

Figure 5: A Brennan and Schwartz example

Of course, for $\alpha = 0$, we obtain the Vasicek solution. In the figure, the Vasicek formula and the numerical solution could not be distinguished. We verify also that the yield rate corresponds to the instantaneous interest rate for a null maturity. We note that small differences in the term structure of a zero coupon $P^c(\tau)$ produce big differences in the term structure of interest rates $R(\tau) = -\frac{\ln P^c(\tau)}{\tau}$. This is another argument for preferring the accuracy provided by Hopscotch methods to the speed of Monte Carlo methods[14].

3.4 Financial elliptic problems

We can also apply the algorithm described in the second section to elliptic problems[15]. In finance, elliptic problems generally arise in the pricing of a perpetual option. This option is like an American option but without any specified maturity. In this case, equation (31) becomes

$$\frac{1}{2}\operatorname{trace}\left(\Sigma(X)^\top P_{XX}(X)\Sigma(X)\rho\right) + \left[\mu(X)^\top - \lambda(X)^\top \Sigma(X)^\top\right] P_X(X) - r(X)P(X) + b(X) = 0 \quad (58)$$

and the option price is not a function of time t. The first derivative P_t and the payoff boundary condition disappear. This latter condition is replaced by other conditions based on the state variables which depend on the particular problem at hand.

For example, Nickell, Perraudin and Varotto [1998] use this analysis for an equity-based credit risk model. They consider two states variables, $V(t)$ the underlying asset value and $D(t)$ the firm's liabilities. The solution is of the form (58). If we consider a transformation to the variable $k = \frac{V}{D}$, the problem becomes

[14]We could also use this argument for computing forward rates $F(\tau, m) = -\frac{1}{m}\ln\frac{P^c(\tau+m)}{P^c(\tau)}$ and $f(\tau) = F(\tau, m) = -\frac{\partial \ln P^c(\tau+m)}{\partial \tau}$.

[15]see the example 3 page 204 of Gourlay and McKee [1977].

a one-dimensionnal PDE problem and a solution can be found. They were able to use this transformation technique, because they assumed that the state variables followed two geometric brownian motions. However with other stochastic processes (mean-reversion for example), it is not obvious that this approach will work. Once again, in this case, Hopscotch methods can easily be applied to solve the problem numerically.

However, we must be **careful** when solving financial elliptic problems numerically, because they are in general more difficult than pure elliptic problems. To illustrate this difficulty, we will apply the Hopscotch method to the model of McDonald and Siegel [1985]. This model considers the problem of irreversible investment. Pindyck [1988] explains the problem as follows:

> "When investment is irreversible and future demand or cost conditions are uncertainty, an investment expenditure involves the exercising, or "killing" of an option — the option to productivity invest at any time in the future. One gives up the possibility of waiting for new information that might affect the desirability or timing of the expenditure; one cannot disinvest should market conditions change adversely. This lost option value must be included as part of the cost of the investment."

In this case, the irreversible investment problem could be view as a perpetuel option problem. Let V be the Net Present Value. The authors suppose that V follows a geometric brownian motion.

$$\left\{ \begin{array}{rcl} dV(t) & = & \alpha V(t)\, dt + \sigma V(t)\, dW(t) \\ V(t_0) & = & V_0 \end{array} \right. \tag{59}$$

Let $C(V)$ be the value of the firm's option to invest. We can show that it satisfies the following set of conditions

$$\left\{ \begin{array}{l} \frac{1}{2}\sigma^2 V^2 C_{VV} + (r-\delta) V C_V - rC = 0 \\ C(0) = 0 \\ C(V^\star) = V^\star - I \\ C_V(V^\star) = 1 \end{array} \right. \tag{60}$$

This EDP equation is just the same as in the perpetual option case given earlier with one state variable. The boundary condition reflects the investment rule. We invest if $V^\star \geqslant V \geqslant I$ with I the initial cost of the project, and, for $V = V^\star$, we exercise the option. The option value is then equal to the payoff of the option $(V^\star - I)_+$. The authors show also that the non-arbitrage condition imposes another boundary condition, well-known as the smooth-pasting condition $C_V(V^\star) = 1$.

Let see how we can solve this problem. Suppose that we know the value $V^\star$. Then, the elliptic problem (60) is equivalent to this following parabolic problem:

$$\left\{ \begin{array}{l} \frac{1}{2}\sigma^2 V^2 C^{(t)}_{VV}(t,V) + (r-\delta) V C^{(t)}_V(t,V) = C^{(t)}_t(t,V) + rC^{(t)}(t,V) \\ C^{(t)}(t,0) = 0 \\ C^{(t)}(t,V^\star) = V^\star - I \\ C^{(t)}_t(t,V) = 0 \end{array} \right. \tag{61}$$

We can apply Hopscotch methods to solve this problem (61) by intializing $C^{(t)}(0,V)$ with initial estimates and by stopping the algorithm when the condition $C^{(t)}_t(t,V) = 0$ is satisfied. Let us consider the example of Dixit and Pyndick [1994], page 153. The parameter values are

r	δ	σ
0.04	0.04	0.20

I is set to 1 and $V^\star$ is equal to 2. The figure 6 shows the convergence of the numerical solution to the exact solution. Note that we have taken uniform random numbers for the initial estimate of the solution values.

This problem (60), even if it is an elliptic problem, presents difficulties however because of the specific boundary conditions. In fact, we don't know the optimal rule $V^\star$. In what follows, we suggest a method to

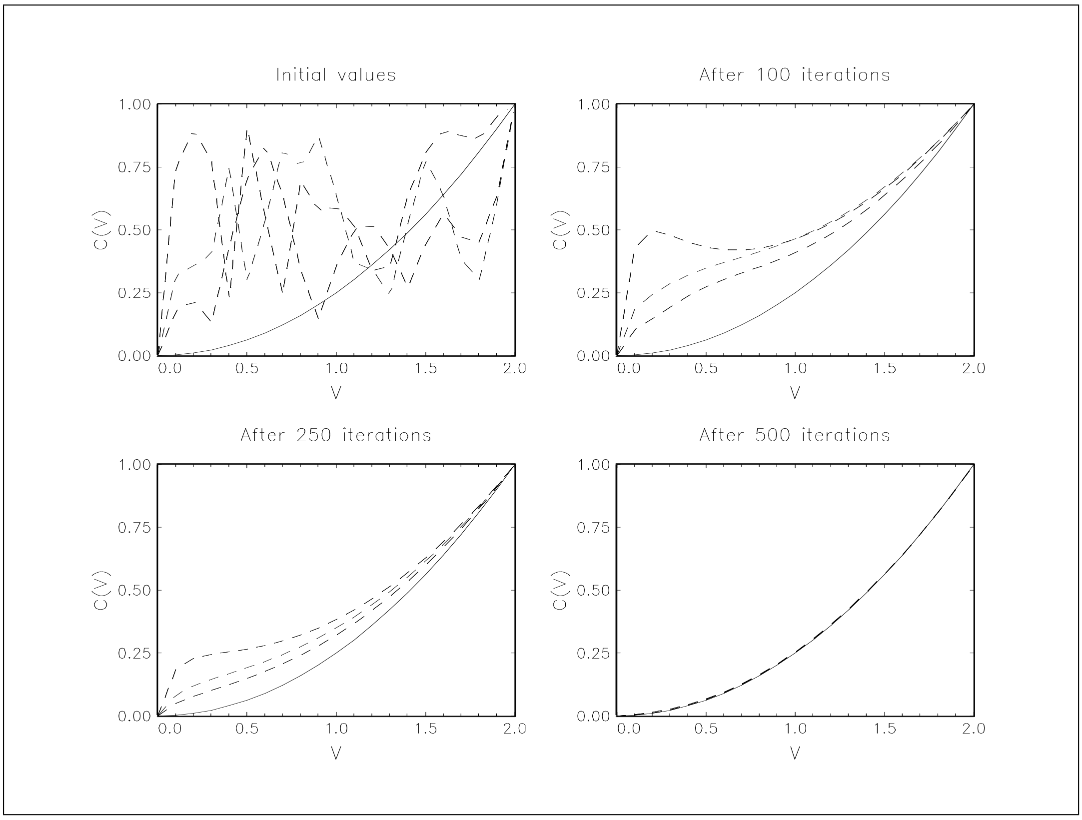

Figure 6: Convergence of the numerical solution

solve this problem. Consider a slighty different version of the previous problem

$$\begin{cases} \frac{1}{2}\sigma^2 V^2 C_{VV} + (r-\delta)\, V C_V - rC = 0 \\ C\,(0) = 0 \\ C\,(V^+) = V^+ - I \end{cases} \tag{62}$$

Now, we may discover $V^\star$ using a grid search. We know that $V^\star \geqslant I$. So, we could solve the problem (62) successively for different values of V^+ and find the value of V^+ that verifies the smooth-pasting condition $C_V\,(V^+) = 1$. We use 21 discretisation points for V and $k = 0.1$ and obtain the following results for $V^+ = \{1 : 0.1 : 3\}$

V^+	$\left.\frac{\partial C}{\partial V}\right\|_{V=V^+}$
1	-0.005
$\vdots$	
1.5	0.6476
1.6	0.7256
1.7	0.7994
1.8	0.8695
1.9	0.9214
2.0	0.9733
2.1	1.0172
2.2	1.0612
2.3	1.0981
2.4	1.1356
2.5	1.1671
$\vdots$	
3.0	1.2977

We can now guess that $V^\star \in [2.0, 2.1]$. For $V^+ = \{2 : 0.01 : 2.1\}$, we have

V^+	$\left.\frac{\partial C}{\partial V}\right\|_{V=V^+}$
2.01	0.9763
2.02	0.9815
2.03	0.9858
2.04	0.9907
2.05	0.9971
2.06	0.9991
2.07	1.0051
2.08	1.0085
2.09	1.0150

If we stop the grid seach now, we obtain the solution $V^\star = 2.06$. This numerical solution is in fact very close to the exact solution. To obtain more accuracy, we have to increase the number of discretisation points for V. Of course, we could also apply the grid search method using the following problem

$$\begin{cases} \frac{1}{2}\sigma^2 V^2 C_{VV} + (r-\delta) V C_V - rC = 0 \\ C(0) = 0 \\ C_V(V^+) = 1 \end{cases} \tag{63}$$

and find the value of V^+ such that $C(V^+) = V^+ - I$. In this case, we find that the optimal value is $V^\star = 2.07$.

V^+	$V^+ - I$	$C(V^+)$
2.01	1.01	1.038
2.02	1.02	1.042
2.03	1.03	1.048
2.04	1.04	1.053
2.05	1.05	1.055
2.06	1.06	1.065
2.07	1.07	1.067
2.08	1.08	1.075
2.09	1.09	1.081

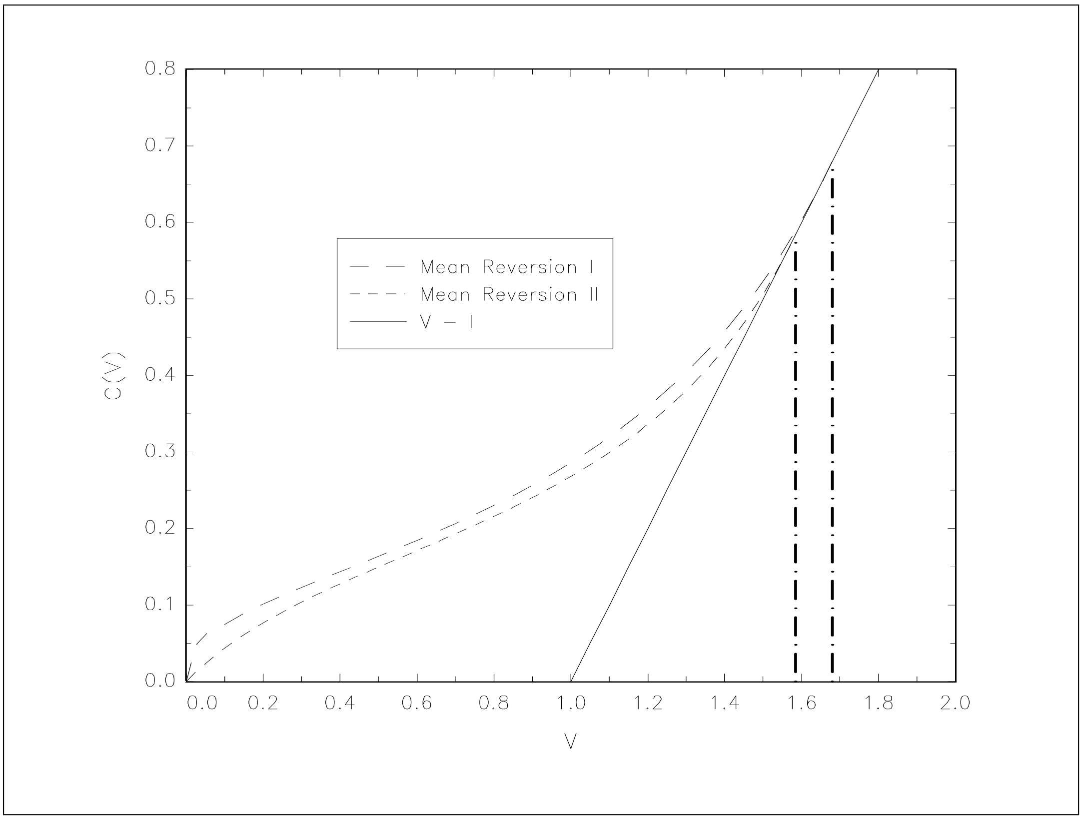

Figure 7: A Dixit and Pindyck example

Hopscotch methods could also be used to find numerical solutions for these type of models with alternative stochastic processes. For example, Dixit and Pindyck suppose that V follows the mean-reverting process

$$dV(t) = \eta\left(\bar{V} - V(t)\right)V(t)\ dt + \sigma V(t)\ dW(t) \tag{64}$$

In this case, the solution is very complicated. It is given by a confluent hypergeometric function and we need to determine some parameters numerically (see Dixit and Pindyck [1994], page 163). We have solved this problem with Hopscotch methods for $I = 1$ with

r	μ	η	$\bar{V}$	σ
0.04	0.04	0.1	1.5	0.20

We have also considered another mean-reverting process for which we believe we can not find a symbolic solution

$$dV(t) = \eta\left(\bar{V} - V(t)\right)V(t)\ dt + \sigma\sqrt{V(t)}\,dW(t) \tag{65}$$

We find the following critical values for $V^\star$: 1.68 for the first process and 1.585 for the second process. The solution of $C(V)$ is reported on the figure 7 which can be compared with figure 5.12 of Dixit and Pindyck.

We are also tempted to develop two-state variable models for the irreversibility problem. Suppose that we introduce another state variable $Y(t)$ in the model. In this case, we could suppose that the critical value $V^\star$ will depend on the state of Y. So, for each value of Y, the value of $V^\star$ will change and this is the reason why we can not use the algorithms presented here to solve the irreversibility problem with two state variables, because the boundary conditions are defined for fixed values, and can not support different values. These considerations show clearly that we have to be careful when we employ Hopscotch methods to solve elliptic problems in finance.

4 Conclusion

In this paper, we have put forward the use of Hopscotch methods in order to solve a large class of Partial Differential Equation problems in finance. We have extended the work of Gourlay in two directions. First, we have considered a more general problem that can be viewed as a Feynman-Kac representation problem. Secondly, we have shown how to take boundary conditions into account, and especially how to mix Dirichlet and Neumann conditions.

We have also demonstrated the algorithm in several important appplications in finance. We have considered option pricing with stochastic volatility, term structure modelling with two state variables and elliptic problems. In fact, the Hopscotch method could be used to solve any general two-state variable financial model.

It would be interesting to improve the algorithm for the case where we have no knowledge about boundary conditions. A possible approach would be to develop a prediction-correction method. The idea is the following. We could use the numerical solution $\{\mathbf{u}_m, m \leqslant M-1\}$ to predict (for example by interpolation) the boundary functions for $m = M$. Then, we could find the solution $\mathbf{u}_M$ from equation (9). We could then use these values $\mathbf{u}_M$ for the region $\mathring{\mathfrak{R}}$ to improve the boundary functions. Finally, solve equation (9) with these new values. We leave this development for a later paper.

References

[1] Balduzzi, P., S.J. Das, S. Foresi and R. Sundaram [1996], A simple approach to three-factor affine term structure models, *Journal of Fixed Income*, **6**, 45-53, december

[2] Black, F. [1976], The pricing of commodity contracts, *Journal of Financial Economics*, **3**, 167-179

[3] Black, F. and M. Scholes [1973], The pricing of options and corporate liabilities, *Journal of Political Economy*, **81**, 637-659

[4] Canabarro, E. [1995], Where do one-factor interest rate models fail?, *Journal of Fixed Income*, **5**, 31-52, september

[5] Dixit, A and R.S. Pindyck [1994], Investment under uncertainty, Princeton University Press

[6] Gane, C.R. and A.R. Gourlay [1977], Block hopscotch procedures for second order parabolic differential equations, *Journal of the Institute of Mathematics and Applications*, **19**, 205-216

[7] Garman, M.B. and S.W. Kohlhagen [1983], Foreign currency option values, *Journal of International Money and Finance*, **2**, 231-237

[8] Golub, G.H. and C.F. Van Loan [1989], Matrix Computations, second edition, Johns Hopkins University Press

[9] Gourlay, A.R. [1970], Hopscotch: a fast second-order partial differential equation solver, *Journal of the Institute of Mathematics and Applications*, **6**, 375-390

[10] Gourlay, A.R. and S. McGuire [1971], General hopscotch algorithm for the numerical solution of partial differential equations, *Journal of the Institute of Mathematics and Applications*, **7**, 216-227

[11] Gourlay, A.R. and S. McKee [1977], The construction of hopscotch methods for parabolic and elliptic equations in two space dimensions with a mixed derivative, *Journal of Computational and Applied Mathematics*, **3**, 201-205

[12] Heston, S.L. [1993], A closed-form solution for options with stochastic volatility with applications to bond and currency options, *Review of Financial Studies*, **6**, 327-343

[13] Hull, J.C. and A. White [1987], The pricing of options on assets with stochastic volatilities, *Journal of Finance*, **42**, 281-300

[14] Kurpiel, A. and T. Roncalli [1998a], Stochastic volatility and contingent claims pricing — the American option case, FERC Working Paper, City University Business School

[15] Kurpiel, A. and T. Roncalli [1998a], Option hedging with stochastic volatility, FERC Working Paper, City University Business School

[16] Lamberton, D. and B. Lapeyre [1997], Introduction au calcul stochastique appliqué à la finance, second edition, Ellipse, Paris

[17] Longstaff, F.A. and E.S. Schwartz [1992], Interest rate volatility and the term structure: A two-factor general equilibrium model, *Journal of Finance*, **47**, 1259-1282

[18] McDonald, R.L. and D.R. Siegel [1985], Investment and the valuation of firms when there is an option to shut down, *International Economic Review*, **26**,

[19] Merton, R.C. [1973], Theory of rational option pricing, *Bell Journal of Economics and Management Science*, **4**, 141-183

[20] Nickell, P., W. Perraudin and S. Varotto [1998], Ratings- versus equity-based credit risk modelling: An empirical analysis, presented in *Advances in Risk Management*, A CEPR/ESRC/IFR Finance Network Workshop, London, 8 october 1998

[21] Press, W.H., S.A. Teukolsky, W.T. Vetterling and B.P. Flannery [1992], Numerical Recipes in Fortran, second edition, Cambridge University Press

[22] Pindyck, R.S. [1988], Irreversibility, capacity choice and the value of the firm, *American Economic Review*, **78**, 969-985

[23] Roncalli, T. and A. Kurpiel [1998], A GAUSS implementation of Hopscotch methods — the PDE2D library, FERC, City University Business School

[24] Thomée, V. [1990], Finite difference methods for linear parabolic equations, in P.G. Ciarlet and J.L. Lions (eds), *Handbook of Numerical Analysis — Volume I*, North-Holland, 5-196

[25] Vasicek, O.A. [1977], An equilibrium characterization of the term structure, *Journal of Financial Economics*, **5**, 177-188

[26] Vermer, J.G. and B.P. Sommeijer [1997], Stability analysis of an odd-even-line hopscotch method for three-dimensional advection-diffusion problems, *SIAM Journal of Numerical Analysis*, **34**, 376-388

[27] Wiggins, J.B. [1987], Option values under stochastic volatility, *Journal of Financial Economics*, **19**, 351-372

A Form of the $\Psi.band$ matrix

$\Psi.band$ takes the following form

$$\Psi.band = \left[\begin{array}{ccccc} (\Psi.band)_- & \vdots & (\Psi.band)_0 & \vdots & (\Psi.band)_+ \end{array}\right]$$

with

$$(\Psi.band)_- = \left[\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
 & \vdots & \\
0 & 0 & 0 \\
0 & \Psi_{N_x+1,1} & \Psi_{N_x+1,2} \\
\Psi_{N_x+2,1} & \Psi_{N_x+2,2} & \Psi_{N_x+2,3} \\
 & \vdots & \\
\Psi_{2N_x,N_x-1} & \Psi_{2N_x,N_x} & 0 \\
 & \vdots & \\
\Psi_{l,l-N_x-1} & \Psi_{l,l-N_x} & \Psi_{l,l-N_x+1} \\
 & \vdots & \\
\Psi_{N_x(N_y-1)+1,N_x(N_y-1)} & \Psi_{N_x(N_y-1)+1,N_x(N_y-1)+1} & \Psi_{N_x(N_y-1)+1,N_x(N_y-1)+2} \\
 & \vdots & \\
\Psi_{N_xN_y-1,N_x(N_y-1)-2} & \Psi_{N_xN_y-1,N_x(N_y-1)-1} & \Psi_{N_xN_y-1,N_x(N_y-1)} \\
\Psi_{N_xN_y,N_x(N_y-1)-1} & \Psi_{N_xN_y,N_x(N_y-1)} & 0
\end{array}\right]$$

$$(\Psi.band)_0 = \left[\begin{array}{ccc}
0 & \Psi_{1,1} & \Psi_{1,2} \\
\Psi_{2,1} & \Psi_{2,2} & \Psi_{2,3} \\
\Psi_{3,2} & \Psi_{3,3} & \Psi_{3,4} \\
 & \vdots & \\
\Psi_{N_x,N_x-1} & \Psi_{N_x,N_x} & 0 \\
0 & \Psi_{N_x+1,N_x+1} & \Psi_{N_x+1,N_x+2} \\
\Psi_{N_x+2,N_x+1} & \Psi_{N_x+2,N_x+2} & \Psi_{N_x+2,N_x+3} \\
 & \vdots & \\
\Psi_{2N_x,2N_x-1} & \Psi_{2N_x,2N_x} & 0 \\
 & \vdots & \\
\Psi_{l,l-1} & \Psi_{l,l} & \Psi_{l,l+1} \\
 & \vdots & \\
0 & \Psi_{N_xN_y+1,N_xN_y+1} & \Psi_{N_xN_y+1,N_xN_y+2} \\
 & \vdots & \\
\Psi_{N_xN_y-1,N_xN_y-2} & \Psi_{N_xN_y-1,N_xN_y-1} & \Psi_{N_xN_y-1,N_xN_y} \\
\Psi_{N_xN_y,N_xN_y-1} & \Psi_{N_xN_y,N_xN_y} & 0
\end{array}\right]$$

$$
(\Psi.band)_+ = \begin{bmatrix}
0 & \Psi_{1,N_x+1} & \Psi_{1,N_x+2} \\
\Psi_{2,N_x+1} & \Psi_{2,N_x+2} & \Psi_{2,N_x+3} \\
\Psi_{3,N_x+2} & \Psi_{3,N_x+3} & \Psi_{3,N_x+4} \\
 & \vdots & \\
\Psi_{N_x,2N_x-1} & \Psi_{N_x,2N_x} & 0 \\
0 & \Psi_{N_x+1,2N_x+1} & \Psi_{N_x+1,2N_x+2} \\
\Psi_{N_x+2,2N_x+1} & \Psi_{N_x+2,2N_x+2} & \Psi_{N_x+2,2N_x+2} \\
 & \vdots & \\
\Psi_{2N_x,3N_x-1} & \Psi_{2N_x,3N_x} & 0 \\
 & \vdots & \\
\Psi_{l,l+N_x-1} & \Psi_{l,l+N_x} & \Psi_{l,l+N_x+1} \\
 & \vdots & \\
0 & 0 & 0 \\
 & \vdots & \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
$$

B Gauss implementation

PDE2D is a Gauss implementation of Hopscotch methods described in this paper. The library and its manual (Roncalli and Kurpiel [1998]) could be downloaded at the following url:

http://www.thierry-roncalli.com/#gauss_l8