Notes on Numerical Methods for Partial DifferentialEquations in Finance

Jacques Printems

Notes on Numerical Methods for Partial Differential Equations in Finance

Jacques Printems

Université de Paris 12
e-mail: printems@univ-paris12.fr
http://perso-math.univ-mlv.fr/users/printems.jacques/

Summer School of Mathematical Finance
CIMPA – IMAMIS – VIETNAM
Institute of Mathematics
18 Hoang Quoc Viet Road
10307 HANOI

23/4 – 4/5/2007

Contents

Chapter 1

Partial Differential Equations in Mathematical Finance

1.1 Black-Sholes analysis

The purpose of theses notes is to introduce some numerical tools based on P.D.E. for the pricing of one of the most popular derivative products based on equities, i.e. the options. Most of this introduction is taken from [8] (chapter 3).
The simplest financial option, a **European call option**, is a contract with the following conditions:

- At a prescribed time in the future, known as the **expiry date**, the owner of the option *may*
- purchase a prescribed asset, known as the **underlying asset** or, briefly, the **underlying**, for a
- prescribed amount, known as the **exercise price** or **strike price**.

The word *may* in this description implies that for the owner (the holder) of the option, this contract is a *right* and not an *obligation*. One of our main concerns throughout theses notes is to find:

- How much would one pay for this right ?

The option to *buy* an asset as described above is known as a **call** option. The right to *sell* an asset is known as a **put** option.

Let us derive now the original Black & Scholes partial differential equation for the price of the simplest options, so-called *vanilla options*. Let us introduce some simple notation.

- We denote by V the value of an option; V is a function of the current value of the underlying asset, S, and time, t: $V = V(S, t)$. The value of the option also depends on the following parameters:
- σ, the volatility of the underlying asset;

- K the exercise price;
- T, the expiry or maturity;
- r the interest rate.

First we consider what happens just at the moment of expiry of a call option, i.e. at time $t = T$.

If $S > K$ at expiry, it makes financial sense to exercise the call option, which cost an amount K to obtain an asset worth S. The profit os such a transaction is $S - K$. On the other hand, if $S < K$ at expiry, we should not exercise the option because we would make a loss of $E - S > 0$. In this case, the option expires valueless. Thus, the value of the call option at expiry can be written as

$$V(S, T) = \max(S - K, 0).$$

More generally, suppose that we have an option whose value $V(S, t)$ depends only on S and t (regardless whether V is a call or a put). Let us also suppose that the asset price S follows the lognormal random walk

$$\frac{\mathrm{d}S}{S} = \mu\,\mathrm{d}t + \sigma\,\mathrm{d}B, \tag{1.1.1}$$

where B is a Wiener process. Using Itô's lemma, we can write

$$\begin{aligned} \mathrm{d}V &= \frac{\partial V}{\partial t}\,\mathrm{d}t + \frac{\partial V}{\partial S}\,\mathrm{d}S + \frac{1}{2}\,\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\,\mathrm{d}t \\ &= \left(\frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{1}{2}\,\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\mathrm{d}t + \sigma S \frac{\partial V}{\partial S}\,\mathrm{d}B. \end{aligned} \tag{1.1.2}$$

This is the random walk followed by V. Note that such an equation need V to be regular enough.

Let us now consider a portfolio Π which contains the option and a ratio $-\Delta S$ of the asset S the option is based on, namely

$$\Pi = V - \Delta S. \tag{1.1.3}$$

Using (1.1.1) and (1.1.2) together gives the random walk followed by Π:

$$\begin{aligned} \mathrm{d}\Pi &= \mathrm{d}V - \Delta \mathrm{d}S, \\ &= \left(\frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{1}{2}\,\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - \mu \Delta S\right)\mathrm{d}t + \sigma S \left(\frac{\partial V}{\partial S} - \Delta\right)\mathrm{d}B, \end{aligned}$$

where we have used the self-financed feature of the portfolio, i.e. $\mathrm{d}(\Delta)S = 0$ (it means that Δ is held fixed during the time-step $\mathrm{d}t$).

We can then eliminate the random component in this random walk by choosing

$$\Delta = \frac{\partial V}{\partial S}. \tag{1.1.4}$$

This result in a portfolio whose increment is now deterministic

$$\mathrm{d}\Pi = \left(\frac{\partial V}{\partial t} + \frac{1}{2}\,\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\,\mathrm{d}t. \tag{1.1.5}$$

The concepts of arbitrage and supply and demand tell us that the right hand side of (1.1.5) must be equal to $r\Pi\,\mathrm{d}t$. Thus, we have

$$r\Pi\,\mathrm{d}t = \left(\frac{\partial V}{\partial t} + \frac{1}{2}\,\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\,\mathrm{d}t. \tag{1.1.6}$$

Then substituting (1.1.3) and (1.1.4) into (1.1.6) leads to

$$\frac{\partial V}{\partial t} + \frac{1}{2}\,\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0. \tag{1.1.7}$$

This is the **Black & Scholes partial differential equation**.

Chapter 2

Finite Difference Methods for PDE

2.1 Introduction

2.1.1 Taylor's expansion: a formal framework

The Finite Difference (FD) method is based on the properties of the Taylor's expansion and on the application of the definition of derivatives. It is the simplest method when the mesh is uniform but it needs the mesh to be regular enough. Indeed, in dimension n, the nodes of the mesh are located at the intersection of n families of curves, each of them belonging to a unique curve of each family.

Its purpose is to approximate differentials equations. Indeed, it allows to construct approximations of the derivative operator $D = \frac{\partial}{\partial x}$ acting on differentiable functions of the real variable x.

Following [5], we can present shortly the method in dimension 1 in a (very) formal framework. Let us consider a spatial discretization

$$-\infty < \cdots < x_{i-1} < x_i < x_{i+1} < \cdots < +\infty,$$

which is regular in the sense that $x_{i+1} - x_i = \Delta x$ is supposed to be constant. Let v a function regular enough on $\mathbb{R}$ (let say $v \in \mathcal{C}_b^\infty$). We set for any $i \in \mathbb{Z}$, $v_i = v(x_i)$ and we introduce the operators

$$\begin{aligned} Ev_i &= v_{i+1}, \\ \delta^+ v_i &= v_{i+1} - v_i, \\ \delta^- v_i &= v_i - v_{i-1}, \\ \delta v_i &= v_{i+1/2} - v_{i-1/2}, \\ \bar{\delta} v_i &= (v_{i+1} - v_{i-1})/2, \\ \mu v_i &= (v_{i+1/2} + v_{i-1/2})/2, \end{aligned} \tag{2.1.1}$$

where we noted $v_{i\pm 1/2} = v(x_{i\pm 1/2})$ with $x_{i+1/2} = (x_{i+1} - x_i)/2$.

Let D be the derivative operator on the functions : $Dv(x) = \frac{\partial v}{\partial x}(x)$. The FD method essentially exploits the Taylor formula in a way which can be summarized like that:

$$v_{i+1} = v_i + \Delta x Dv(x_i) + \frac{\Delta x^2}{2} D^2 v(x_i) + \frac{\Delta x^3}{3!} D^3 v(x_i) + \cdots + \frac{\Delta x^n}{n!} D^n v(x_i) + \cdots,$$

which allows to write

$$Ev_i = \exp(\Delta x\, D) v_i, \quad \text{and } E = \exp(\Delta x\, D).$$

We will prefer this formulation :

$$D = \frac{1}{\Delta x}\ln(E). \tag{2.1.2}$$

2.1.2 Forward Finite Difference for the first derivative

The formula (2.1.2) allows to express D in function of δ^+:

$$\begin{aligned} D &= \frac{1}{\Delta x}\ln(1+\delta^+) \\ &= \frac{1}{\Delta x}\sum_{k=1}^{\infty}(-1)^{k+1}\frac{(\delta^+)^k}{k}, \\ &= \frac{1}{\Delta x}\left(\delta^+ - \frac{(\delta^+)^2}{2} + \frac{(\delta^+)^3}{3} + \cdots\right) \end{aligned} \tag{2.1.3}$$

Of course, this expansion should be restricted to analytic functions. For polynomial functions, the right member of the expansion has only a finite number of terms. Since $\delta^+ = \mathcal{O}(\Delta x)$, the truncated expansion with $n+1$ first terms yields an approximation $\mathcal{O}(\Delta x^n)$ of D. This gives:

$$\begin{aligned} Dv_i &= \frac{v_{i+1}-v_i}{\Delta x} - \frac{\Delta x}{2}v_{xx}(x_i) + \cdots \\ &= \frac{v_{i+1}-v_i}{\Delta x} + \mathcal{O}(\Delta x) \end{aligned} \tag{2.1.4}$$

$$\begin{aligned} Dv_i &= \frac{-3v_i + 4v_{i+1} - v_{i+2}}{2\Delta x} + \frac{\Delta x^2}{3}v_{xxx}(x_i) + \cdots \\ &= \frac{-3v_i + 4v_{i+1} - v_{i+2}}{2\Delta x} + \mathcal{O}(\Delta x^2) \end{aligned} \tag{2.1.5}$$

(useful for the approximation of the derivative at the frontier of a domain)

2.1.3 Backward Finite Difference for the first derivative

Again (2.1.2) and (2.1.1) yield to:

$$\begin{aligned} D &= -\frac{1}{\Delta x}\ln(1-\delta^-) \\ D &= \frac{1}{\Delta x}\left(\delta^- + \frac{(\delta^-)^2}{2} + \frac{(\delta^-)^3}{3} + \cdots\right). \end{aligned} \tag{2.1.6}$$

The formulæ are of the same kind:

$$Dv_i = \frac{v_i - v_{i-1}}{\Delta x} + \mathcal{O}(\Delta x) \tag{2.1.7}$$

$$Dv_i = \frac{3v_i - 4v_{i-1} + v_{i-2}}{2\Delta x} + \mathcal{O}(\Delta x^2) \tag{2.1.8}$$

2.1.4 Centered Finite Difference for the first derivative

In a first time, we write D with respect to δ. It comes

$$\delta = 2\sinh\left(\frac{\Delta x D}{2}\right),$$

and then

$$D = \frac{2}{\Delta x}\text{argsinh}\left(\frac{\delta}{2}\right) = \frac{2}{\Delta x}\ln\left(\frac{\delta}{2} + \sqrt{1+\frac{\delta^2}{4}}\right),$$

or

$$D = \frac{2}{\Delta x}\left(\frac{\delta}{2} - \frac{1}{6}\left(\frac{\delta}{2}\right)^3 + \frac{3}{40}\left(\frac{\delta}{2}\right)^5 - \frac{5}{112}\left(\frac{\delta}{2}\right)^7 + \cdots\right). \tag{2.1.9}$$

Nevertheless, this formula gives an expression of D with respect to the value of v at the middle of the mesh and not at the nodes. In order to obtain an expression in function of $\bar{\delta}$, we use the following relation between μ and δ:

$$\mu^2 = 1 + \frac{\delta^2}{4},$$

which can be easily derived from the relations (2.1.1).

Multiplying (2.1.9) by the relation $1 = \mu(1+\delta^2/4)^{-1/2}$, we obtain

$$D = \frac{1}{\Delta x}\mu\left(\delta - \frac{1}{6}\delta^3 + \frac{1}{30}\delta^5 - \frac{1}{140}\delta^7 + \frac{1}{630}\delta^9\cdots\right). \tag{2.1.10}$$

Finally $\mu\delta = \bar{\delta}$ gives:

$$Dv_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} - \frac{\Delta x^2}{6}u_{xxx} + \cdots$$

$$Dv_i = \frac{u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2}}{12\Delta x} + \frac{\Delta x^4}{30}u_{5x} + \cdots$$

and so on.

2.1.5 FD approximations for the second derivative

We end this subsection by FD schemes for D^2. Such approximations can be derived from Eqs (2.1.3), (2.1.6) and (2.1.9) squared.

- First order forward FD :

$$D^2 v_i = \frac{v_{i+2} - 2v_{i+1} + v_i}{\Delta x^2} + \mathcal{O}(\Delta x)$$

- First order backward FD :

$$D^2 v_i = \frac{v_i - 2v_{i-1} + v_{i-2}}{\Delta x^2} + \mathcal{O}(\Delta x)$$

- Second order centered FD :

$$D^2 v_i = \frac{v_{i+1} - 2v_i + v_{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{2.1.11}$$

2.1.6 FD approximations in dimension 2

Let v be a regular enough function of the variables x and y. We focuse here on the approximation of the second deriviative of v with respect to x, y: $\dfrac{\partial^2 v}{\partial x^2}$, $\dfrac{\partial^2 v}{\partial y^2}$ and $\dfrac{\partial^2 v}{\partial x \partial y}$.

Approximations of the first two is simply the corresponding approximation of the second derivative of $v(\cdot, y)$ viewed as a function of x or $v(x, \cdot)$ viewed as a function of y.

Particular attention has to be paid for the the mixed derivative. A first try leads to the second order approximation:

$$\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{i,j} = \frac{1}{\Delta x \Delta y} \bar{\delta_x} \bar{\delta_y} u_{i,j} + \mathcal{O}(\Delta x^2, \Delta y^2), \tag{2.1.12}$$

with obvious notations. This is represented by the following figure 2.1

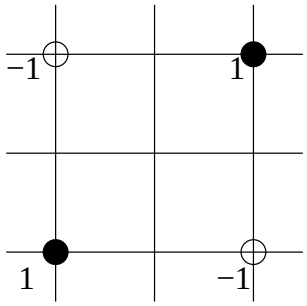


Figure 2.1: Representation of formula (2.1.12).

Other consistant examples (i.e. approximation of order greater or equal to one) are

$$\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{i,j} = \frac{1}{\Delta x \Delta y} \bar{\delta_x} \delta_y^+ u_{i,j} + \mathcal{O}(\Delta x^2, \Delta y), \tag{2.1.13}$$

$$\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{i,j} = \frac{1}{\Delta x \Delta y} \delta_x^+ \delta_y^+ u_{i,j} + \mathcal{O}(\Delta x, \Delta y), \tag{2.1.14}$$

$$\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{i,j} = \frac{1}{\Delta x \Delta y} \delta_x^- \delta_y^- u_{i,j} + \mathcal{O}(\Delta x, \Delta y), \tag{2.1.15}$$

whose representations are in Fig.2.2. The truncature errors in the last two approximations (2.1.14) and (2.1.15) are of opposed sign, it means that the arithmetic mean will be second order approximation in both x and y:

$$\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{i,j} = \frac{1}{2\Delta x \Delta y} \left(\delta_x^+ \delta_y^+ + \delta_x^- \delta_y^-\right) u_{i,j} + \mathcal{O}(\Delta x^2, \Delta y^2). \tag{2.1.16}$$

The same is also true for

$$\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{i,j} = \frac{1}{2\Delta x \Delta y} \left(\delta_x^+ \delta_y^- + \delta_x^- \delta_y^+\right) u_{i,j} + \mathcal{O}(\Delta x^2, \Delta y^2). \tag{2.1.17}$$

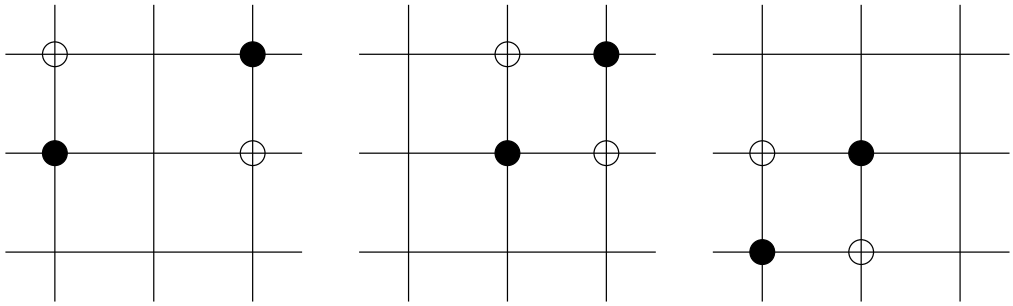


Figure 2.2: Representation of formulæ (2.1.13)–(2.1.15).

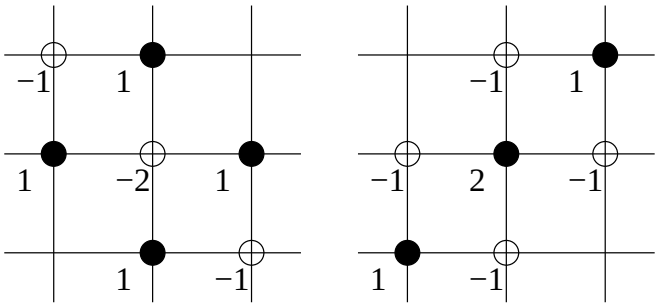


Figure 2.3: Representation of formulæ (2.1.16)–(2.1.17).

More generally, every convex linear combinaison of the last two approximations leads to a second order approximation. We will see that (2.1.17)–(2.1.16) are stable approximation of the mixed derivative whereas (2.1.12) is not.

2.2 Short study of the heat equation. Stability and precision

In order to illustrate the basic concepts of numerical analysis, such as stability and accuracy, in the framework of finite difference methods, we will consider the heat equation. Let us recall that the Black & Scholes PDE under suitable change of varaibles can be transformed into the heat equation.

Let $\alpha > 0$. We consider the following PDE:

$$\frac{\partial u}{\partial t}(x,t) = \alpha \frac{\partial^2 u}{\partial x^2}(x,t), \quad (x,t) \in]-L, L[\times]0, T[, \tag{2.2.18}$$

with the following boundary and initial conditions:

$$u(x,0) = h(x), \quad x \in]-L, L[, \tag{2.2.19}$$

and

$$u(-L,t) = l_1(t), \quad u(L,t) = l_2(t), \quad t \in]0, T[. \tag{2.2.20}$$

Computation domain : cartesian grid $x_i = i\Delta x$, $t_n = n\Delta t$ for $1 \leq i \leq M$ and $0 \leq n \leq N$ with

$$\Delta t = T/N, \quad \Delta x = 2L/(M+1),$$

for N et M two integer ≥ 1.

The operator $\dfrac{\partial^2}{\partial x^2}$ is approached by (2.1.11) at the points $x_i \in]-L, L[$ by

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_n) = \alpha \, \frac{u(x_{i+1}, t_n) - 2u(x_i, t_n) + u(x_{i-1}, t_n)}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{2.2.21}$$

As regards the time integrator, we use a so-called θ-scheme which can be found from (2.1.4) and (2.1.5) applied to $u(x_i, t)$ at time $t = t_n$ (resp. at time $t = t_{n+1}$) and a convex linear combinaison:

$$\begin{aligned} \frac{u(x_i, t_{n+1}) - u(x_i, t_n)}{\Delta t} &= \theta \frac{\partial u}{\partial t}(x_i, t_{n+1}) + (1-\theta)\frac{\partial u}{\partial t}(x_i, t_n) + \mathcal{O}(\Delta t), \\ &= \theta\alpha \frac{\partial^2 u}{\partial x^2}(x_i, t_{n+1}) + (1-\theta)\alpha \frac{\partial^2 u}{\partial x^2}(x_i, t_n) + \mathcal{O}(\Delta t), \end{aligned} \tag{2.2.22}$$

where $\theta \in [0, 1]$.

When $\theta = 0$, we recover the Explicit Euler-scheme, $\theta = 1$, the Implicit Euler Scheme and when $\theta = 1/2$, the Crank–Nicolson scheme. Let us note that in the case where $\theta = 1/2$, the truncation errors in (2.1.4) and (2.1.5) are opposite, so that the term in Δ vanishes and we recover a second order scheme in time. Any other choice of θ leads a first order scheme in time.

We replace $u(x_i,t_n)$ par u_i^n et $\mathcal{O}$ by 0 and then (2.2.21)–(2.2.22) gives the following system of difference equations:

$$(\Sigma)\begin{cases} \dfrac{u_i^{n+1}-u_i^n}{\Delta t}=\dfrac{\alpha\theta}{\Delta x^2}\left(u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}\right)+\dfrac{\alpha(1-\theta)}{\Delta x^2}\left(u_{i+1}^{n}-2u_i^{n}+u_{i-1}^{n}\right), \\ \\ 1\le i\le M, \quad 0\le n\le N. \end{cases}$$

We define now the *consistancy error* of the scheme Σ.

Definition 2.2.1 *We call* consistance error *the remaining terms when you replace u_i^n in the scheme by the exact solution $u(x_i,t_n)$, denoted by ε_i^n.*

Eqs (2.2.21)–(2.2.22) leads easily to the bound

$$|\varepsilon_i^n|\le C\left(\Delta t^\beta+\Delta x^2\right), \tag{2.2.23}$$

where C depends on the higher order derivatives of u with respect to x and t and where $\beta=1$ si $\theta\neq 1/2$ and $\beta=2$ otherwise.

Let us make now an important remark which is at the center of Numerical Analysis : the smallness of ε_i^n gives us a clue about the degree of approximation of **the equation**, not about the approximation of the **unknown** u. We will say that (Σ) is *consistant* iif ε_i^n tends to 0 when $\Delta t,\Delta x$ tend to 0. So a scheme is consistant when it approximates the equation.

Before going further, let us use the matrix notation. We set $U^n=\begin{bmatrix} u_1^n \\ \vdots \\ u_M^n \end{bmatrix}\in\mathbb{R}^M$.

Then the scheme (Σ) can be written as

$$\left(I+\alpha\theta\frac{\Delta t}{\Delta x^2}L\right)U^{n+1}=\left(I-\alpha(1-\theta)\frac{\Delta t}{\Delta x^2}L\right)U^n+\Delta t F^n, \tag{2.2.24}$$

where

$$L=\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}\in\mathbb{R}^{M\times M} \text{ and } F^n=\begin{bmatrix} \alpha\theta\frac{\Delta t}{\Delta x^2}l_1(t_{n+1})+\alpha(1-\theta)\frac{\Delta t}{\Delta x^2}l_1(t_n) \\ 0 \\ \vdots \\ 0 \\ \alpha\theta\frac{\Delta t}{\Delta x^2}l_2t_{n+1})+\alpha(1-\theta)\frac{\Delta t}{\Delta x^2}l_2(t_n) \end{bmatrix}\in\mathbb{R}^M.$$

Eq. (2.2.24) can be written under the following "integrated" form :

$$U^n=A_\theta^nU^0+\Delta t\sum_{k=0}^{n-1}A_\theta^{n-k-1}\left(I+\alpha\theta\frac{\Delta t}{\Delta x^2}L\right)^{-1}F^k, \tag{2.2.25}$$

with $U^0=\begin{bmatrix} u(x_1,0) \\ \vdots \\ u(x_M,0) \end{bmatrix}$ (e.g.), and where

$$A_\theta=\left(I+\alpha\theta\frac{\Delta t}{\Delta x^2}L\right)^{-1}\left(I-\alpha(1-\theta)\frac{\Delta t}{\Delta x^2}L\right).$$

Eq. (2.2.25) suggests that at each time step, one has to inverse a linear system (except when $\theta = 0$). We do not precise in theses notes the way we are doing that. Let us say that we can use in the context of the heat equation in dimension 1, the Gauss elimination algorithm without encounter any problems (as storage, efficiency, ...).

We are going to show now that the scheme (Σ) is convergent in some sense, i.e. that u_i^n tends to $u(x_i, t_n)$ in some sense when Δt and Δx tend to 0. We can imagine that the smallness of ε_i^n is not sufficient for that. Indeed, the formulation (2.2.25) says that the consistancy errors are accumulating in time in order to form the global error. We need one more assumption : the **stability**.

Hypothesis 1 *Let us suppose that A_θ is such that*

$$\|A_\theta\|_2 \leq 1. \tag{2.2.26}$$

We have denoted by $\|B\|_2$, for a matrix B, the operator norm based on the Euclidean norm in $\mathbb{R}^M$. It means that

$$\|B\|_2 = \sup_{x \in \mathbb{R}^M,\, \|x\|_2 = 1} \|Bx\|_2.$$

We will say that the scheme is **L^2-stable** if A_θ satisfy the assumption (2.2.26). We can now state :

Proposition 2.2.2 *Under the assumption (2.2.26), (Σ) is convergent in the sense where*

$$\lim_{N,M \longrightarrow +\infty} \max_{0 \leq n \leq N} \Delta x \sum_{i=1}^{M} |u(x_i, t_n) - u_i^n|^2 = 0.$$

Proof

We set $\widetilde{U}^n = \begin{bmatrix} u(x_1, t_n) \\ \vdots \\ u(x_M, t_n) \end{bmatrix}$, $E^n = U^n - \widetilde{U}^n$, the vector of the (global) errors et $\mathcal{E}^n = {}^t[\varepsilon_1^n, \ldots, \varepsilon_M^n]$ the consistancy error (or local errors) at time t_n.

Due to the definition 2.2.1 of the local errors, we have directly:

$$\left(I + \alpha\theta \frac{\Delta t}{\Delta x^2} L\right) E^{n+1} = \left(I - \alpha(1-\theta) \frac{\Delta t}{\Delta x^2} L\right) E^n + \Delta t\, \mathcal{E}^n. \tag{2.2.27}$$

The stability criterion (2.2.26) allows us to bound the global error at time t_n since we can control the accumulation of the local errors. Indeed, the time integrated version of (2.2.27), i.e. (2.2.25) where U^n replaced by E^n and F^n replaced by $\mathcal{E}^n$ yields

$$\|E^n\|_2 \leq \|E^0\| + \Delta t \sum_{k=0}^{n-1} \left\| \left(I + \alpha\theta \frac{\Delta t}{\Delta x^2} L\right)^{-1} \right\| \|\mathcal{E}^k\|. \tag{2.2.28}$$

We show now that the operator norm in the last inequality can be majorized by 1. Let $Y \in R^M$ and let us consider $X \in \mathbb{R}^M$ such that

$$\left(1 + \alpha\theta \frac{\Delta t}{\Delta x^2} L\right) X = Y.$$

It is sufficient to show that $\|X\|_2 \leq \|Y\|_2$. Indeed, taking the scalar product of the last equality by X, leads to

$$\|X\|_2^2 + \alpha\theta \frac{\Delta t}{\Delta x^2} (LX, X) = (Y, X).$$

We note now that L is a positive (even definite positive) operator. Indeed,

$$\begin{aligned}(LX,X) &= \sum_{i=1}^{M}(2z_i - z_{i-1} - z_{i+1})\, z_i \\ &= \sum_{i=1}^{M}(z_{i+1}-z_i)^2 + z_1^2 + z_M^2 \quad \text{(by discrete integration by part)},\end{aligned}$$

where we set $z_0 = z_{M+1} = 0$. We conclude by the Cauchy-Schwarz inequality.

Now (2.2.28) is reduced to

$$\|E^n\|_2 \leq \|E^0\| + \Delta t \sum_{k=0}^{n-1} \|\mathcal{E}^k\|. \tag{2.2.29}$$

Plugging (2.2.23) in (2.2.29) yields

$$\|E^n\|_2 \leq C\Delta t \sum_{k=0}^{n-1} \sqrt{M}(\Delta t^\beta + \Delta x^2) \leq CT\sqrt{M}(\Delta t^\beta + \Delta x^2),$$

which is the desired result since $\sqrt{M} \leq \Delta x^{-1/2}$. ∎

Remark 2.2.3 *We have shown the convergence in the functional space $L^\infty(0,T,L^2(\mathbb{R}))$. Indeed, we note the analogy between*

$$\max_{0\leq n\leq N} \Delta x \sum_{i=1}^{M} |e_i^n|^2 \quad \textit{and} \quad \sup_{t\in[0,T]} \int_0^L |e(x,t)|^2 \,\mathrm{d}x.$$

What are now the conditions on the parameters (α, θ and especially Δt and Δx) under which Σ is L^2-stable ? We can answer by the following spectral study. In fact, the eigenvectors and eigenvalues of L can be found to be (using trigonometric formulæ for example)

$$X_k = \begin{bmatrix} \sin\left(\frac{k\pi}{M+1}\right) \\ \vdots \\ \sin\left(\frac{Mk\pi}{M+1}\right) \end{bmatrix}, \quad \lambda_k = 4\sin^2\left(\frac{k\pi}{2(M+1)}\right), \quad 1 \leq k \leq M.$$

So the eigenvalue of A_θ are

$$\mu_k = \frac{1 - 4\alpha(1-\theta)\sin^2\left(\frac{k\pi}{2(M+1)}\right)}{1 + 4\alpha\theta\sin^2\left(\frac{k\pi}{2(M+1)}\right)}, \quad 1 \leq k \leq M.$$

It is easy to verify that the function $\psi : t \mapsto (1 - 4\alpha(1-\theta)\Delta t/\Delta x^2 t)/(1 + 4\alpha\theta\Delta t/\Delta x^2 t)$ is non increasing on $[0,1]$. Moreover, $\psi(1) = (1 - 4\alpha(1-\theta)\Delta t/\Delta x^2)/(1 + 4\alpha\theta\Delta t/\Delta x^2)$. Now we note that $\psi(0) = 1$. In order to have $|\psi(t)| \leq 1$ and hence $|\mu_k| \leq 1$ for any k, we have to show that $\psi(1) \geq -1$. This gives us the L^2-stability criterion:

$$\alpha(1-2\theta)\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}. \tag{2.2.30}$$

We are going now to numerically test whetever these conditions of stability (and hence of convergence) are "sharp" or not in the case $\theta = 0$ (Explicit Euler scheme). In order to do that we choose $\alpha = 1$ a piecewise constant initial condition h on $[0, 1]$:

$$h(x) = u(x, 0) = \begin{cases} 1 & \text{si } x \in [0.25, 0.75] \\ 0 & \text{sinon.} \end{cases}$$

Let us fix $\Delta x = 0.01$. The stability criterion (2.2.30) gives a theoretical limit for the time step: $\Delta t = 0.5 \times 10^{-4}$.

The following SCILAB script simulate the scheme (Σ) with a time discretization just below and just beyond the theoretical limit. We can see in Fig 2.4 that the condition (2.2.30) have to be satisfied and is a necessary condition when $\theta = 0$.

```
// Discretization parameters
M=99;
N=22000; // Stable case
//N=18000; // Instable case
nmax=15;

// Parameters
alpha=1;
theta=0;

// Do not edit below
L=2*diag(ones(M,1))-diag(ones(M-1,1),-1)-diag(ones(M-1,1),1);
I=eye(M,M);

Deltat=1/N;
Deltax=1/(M+1);

A=inv(I + alpha*theta*Deltat/Deltax^2*L)*(I - (1-theta)*alpha*Deltat/Deltax^2*L);

// Initialisation of U^n
x=linspace(0,1,M+2);
y=1*((x>=0.25)&(x<=0.75));
U=y(2:100)';

// time iteration
for i=1:nmax,
  U=A*U;
end
xbasc(); plot(x(2:100),U)
```

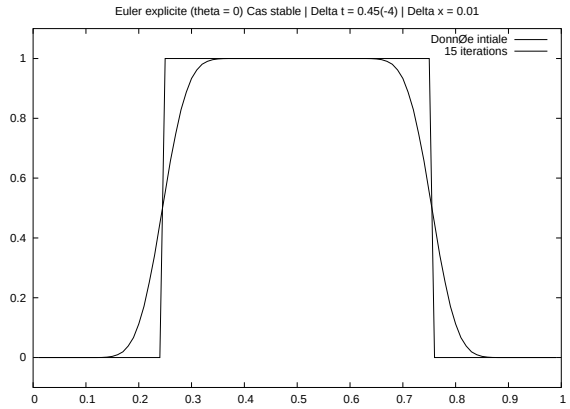
Before the end of this paragraph, let us mention another important notion of stability : the L^∞-stability. We will say that the scheme (Σ) is L^∞**-stable** iff

$$\|A_\theta\|_\infty \leq 1, \tag{2.2.31}$$

where we have used here the notation $\|B\|_\infty$, for a matrix B, for the operator norm based on the supremum norm in $\mathbb{R}^M$. It means that

$$\|B\|_\infty = \sup_{x \in \mathbb{R}^M,\, \|x\|_\infty = 1} \|Bx\|_\infty, \quad \text{where } \|x\|_\infty = \max_{1 \leq i \leq M} |x_i|.$$

a)



b)

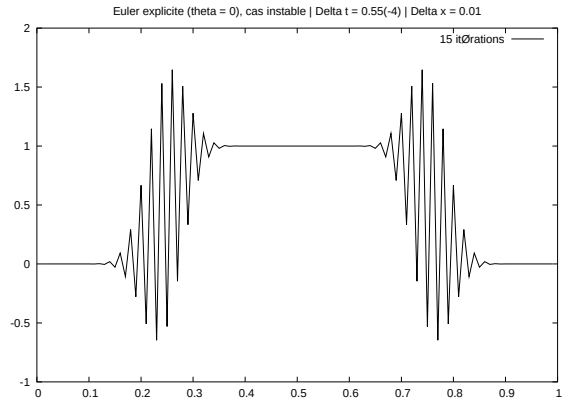


Figure 2.4: Excplit Euler scheme ($\theta = 0$) after 15 iterations in time for $\alpha = 1$, $M = 99$, $N = 22000$ a), $N = 18000$ b).

What are the conditions on the parameters under which the scheme is L^∞-stable ? And does it converge in this case ?

Proposition 2.2.4 *Let $\Delta t > 0$ and $\Delta x > 0$ such that*

$$2\alpha(1-\theta)\frac{\Delta t}{\Delta x^2} \leq 1. \tag{2.2.32}$$

Then the scheme (Σ) is L^∞-stable and moreover it converges in the following sense:

$$\lim_{N,M \longrightarrow +\infty} \max_{0 \leq n \leq N} \max_{1 \leq i \leq M} |u(x_i, t_n) - u_i^n| = 0.$$

Proof

First let us prove stability. Let $X \in \mathbb{R}^M$, $\|X\|_\infty = 1$ and $Y = A_\theta X$. We have

$$\left(I - \alpha(1-\theta)\frac{\Delta t}{\Delta x^2} L\right) X = \left(I + \alpha\theta\frac{\Delta t}{\Delta x^2} L\right) Y,$$

or, for any $1 \leq i \leq M$,

$$\left(1 - 2\alpha(1-\theta)\frac{\Delta t}{\Delta x^2}\right) z_i + \alpha(1-\theta)\frac{\Delta t}{\Delta x^2}(z_{i+1} + z_{i-1}) = \left(1 + 2\alpha\theta\frac{\Delta t}{\Delta x^2}\right) y_i - \alpha\theta\frac{\Delta t}{\Delta x^2}(y_{i+1} + y_{i-1}), \tag{2.2.33}$$

where we have set $z_0 = z_{M+1} = y_0 = y_{M+1} = 0$.

Let i_0, j_0 be the indices such that $y_{i_0} = \max_{1 \leq i \leq M} y_i$ and $y_{j_0} = \min_{1 \leq j \leq M} y_j$. The system of equations (2.2.33) imply

$$\begin{aligned} y_{i_0} &\leq z_{i_0} + 2\alpha(1-\theta)\frac{\Delta t}{\Delta x^2}(1 - z_{i_0}), \\ y_{j_0} &\geq z_{j_0} + 2\alpha(1-\theta)\frac{\Delta t}{\Delta x^2}(-1 - z_{j_0}). \end{aligned}$$

Thus a sufficient condition of stability is (2.2.32). Morevover, if we consider Ineq. (2.2.28) where $\|\cdot\|_2$ is replaced by $\|\cdot\|_\infty$, then we will have convergence of the scheme in the same techniques than in Proposition 2.2.2 provided we show that the norm operator in the sum of (2.2.28) is bounded. Indeed, we already have

done this. In fact, let us consider again X and Y with $\|X\|_\infty = 1$ and $X = (I + \alpha\theta\Delta t/\Delta x^2 L)Y$. The same ideas as above can by applied here and we get

$$y_{i_0} \leq z_{i_0} \leq 1, \quad \text{and } y_{j_0} \geq z_{j_0} \geq -1,$$

with the same notations. Hence $\|Y\|_\infty \leq 1$. ∎

We sum up in the table 2.1 the sufficient conditions of stability of θ–schemes applied to the heat equation.

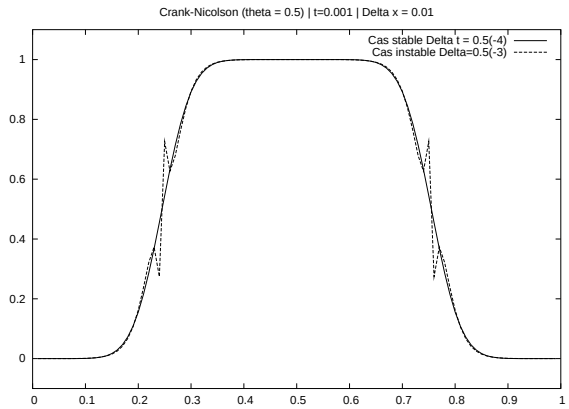
Table 2.1: Sufficient conditions of L^2 and L^∞-stability of θ–schemes.

θ–schemes	L^∞	L^2
$\theta = 0$	$2\alpha\Delta t \leq \Delta x^2$	$2\alpha\Delta t \leq \Delta x^2$
$\theta = 0.5$	$\alpha\Delta t \leq \Delta x^2$	no conditions
$\theta = 1$	no conditions	no conditions
$\theta \in [0, 1]$	$2\alpha(1-\theta)\Delta t \leq \Delta x^2$	$2(1-2\theta)\Delta t \leq \Delta x^2$

We will now test the stability conditions in the case of the Crank–Nicolson scheme ($\theta = 1/2$). We set $\Delta x = 0.01$ ($M = 99$ points). The critical value of Δt is 10^{-4}. Figure 2.5 (a) shows the result obtained at time $t = 0.001$ in the two cases (stable and instable) and (b) at time $t = 0.01$. We can see that the differences vanishes with time.

We have used the previous SCILAB script.

a)



b)

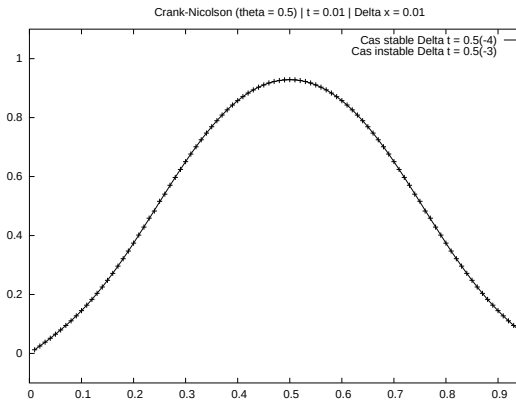


Figure 2.5: Crank–Nicolson scheme ($\theta = 0.5$) for $\alpha = 1$ and $N = 99$. a) $t = 0.001$: the dash line represent the instable case ($\Delta t = 0.5 \times 10^{-3}$) and the solid line, the stable case ($\Delta t = 0.5 \times 10^{-4}$). Same notations for b) at time $t = 0.01$.

2.3 General parabolic case

One can easliy see that results of stability of the previous section can be generalized to the case where the discrete Laplacian $-L$ is replaced by a matrix B such that the following holds:

$$\left\{\begin{array}{l} b_{i,i} \leq 0, \\ b_{i,j} \geq 0, \quad i \neq j, \\ \sum_{j=1}^{M} b_{i,j} = 0. \end{array}\right. \tag{2.3.34}$$

The scheme (Σ) can be written in this case as

$$\left(I - \alpha\theta \frac{\Delta t}{\Delta x^2} B\right) U^{n+1} = \left(I + \alpha(1-\theta)\frac{\Delta t}{\Delta x^2} B\right) U^n + \Delta t F^n.$$

What are the new L^∞-stability criterion ? With the same notation than in the previous section, we have

$$y_{i_0} - \alpha\theta\frac{\Delta t}{\Delta x^2} b_{i_0,i_0} y_{i_0} - \alpha\theta\frac{\Delta t}{\Delta x^2}\sum_{j\neq i_0} b_{i_0,j} y_j = x_{i_0} + \alpha(1-\theta)\frac{\Delta t}{\Delta x^2} b_{i_0,i_0} x_{i_0} + \alpha(1-\theta)\frac{\Delta t}{\Delta x^2}\sum_{j\neq i_0} b_{i_0,j} x_j.$$

The conditions $[M]$ implies that the left member of the previous equation is bounded by below by y_{i_0}. Same techniques apply and give as stability's criterion:

$$\alpha \max_{1\leq i\leq M} |b_{i,i}|(1-\theta)\frac{\Delta t}{\Delta x^2} \leq 1. \tag{2.3.35}$$

More generally, if $X = {}^t[z_1, \ldots, z_M] \in \mathbb{R}^M$ and $Y = A_\theta X$ then the same computations as above give

$$y_{i_0} \leq x_{i_0} + \alpha(1-\theta)\frac{\Delta t}{\Delta x^2}|b_{i_0,i_0}|\left(\max_{1\leq i\leq M} x_i - x_{i_0}\right).$$

Condition (2.3.35) leads to

$$\min_{1\leq i\leq M} x_i \leq y_i \leq \max_{1\leq i\leq M} x_i.$$

It is the discrete maximum principle. If we replace (2.3.34) by

$$\left\{\begin{array}{l} b_{i,i} \leq 0, \\ b_{i,j} \geq 0, \quad i \neq j, \\ \sum_{j=1}^{M} b_{i,j} \leq -\lambda < 0, \end{array}\right. \tag{2.3.36}$$

where $\lambda > 0$, then B satisfies the *strong discrete maximum principle*:

$$BX \geq 0 \Rightarrow X \leq 0.$$

Remark 2.3.1 *There are other properties satisfied by the matrices such that (2.3.36) holds: if B satisfies (2.3.36) then B is inversible and B^{-1} has positive coefficients.*

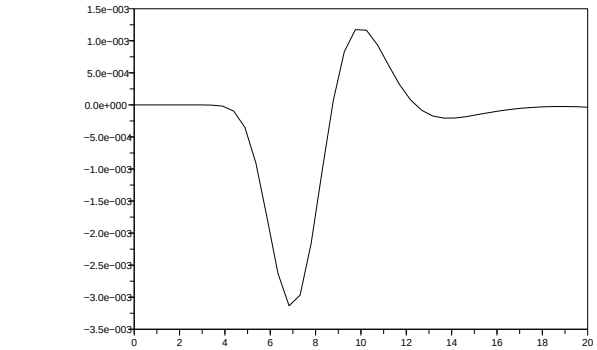
We can sum up this section by saying that (2.3.34) (or (2.3.36)) and (2.3.35) are the two stability criterion in the case of L^∞-stability. The first is spatial and the second one is temporal. From now on, we will try to be in this case when we will discretize Black & Scholes type PDE.

2.4 Numerical computation of options in dimension 1

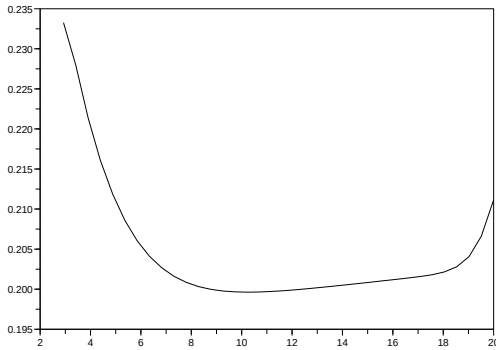
2.4.1 European case

We test our FD scheme on the pricing of a European Put (cf. appendix A.1.) with $r = 0.1$, $\sigma = 0.2$, $K = 1$ and $T = 1$. The inversion of the linear system are done by means of standart direct methods which can be found in SCILAB software (e.g.) (see [4] for a complete reference of the subject of linear system inversion).

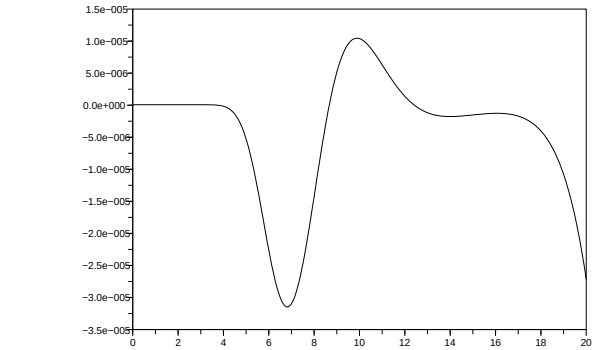
We can see below different effect of discretization on the error with respect to the real price (computed owing to Black & Scholes formula) and on the implied volatilities computed thanks to a Newton algorithm (see Appendix A for a Scilab script).



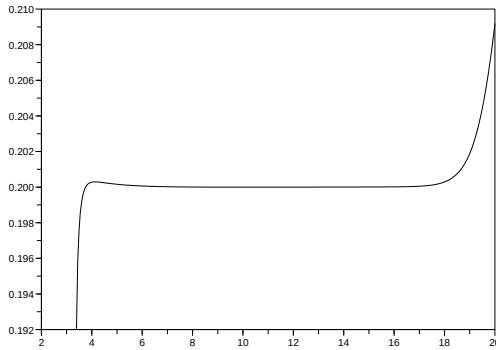
(a)



(c)



(b)



(d)

Figure 2.6: Different spatial discretization ($\Delta t = 1e-2$). $(a), (c)$ $\Delta x = 0.5$. $(b), (d)$ $\Delta x = 0.05$. Curves (c) and (d) described in both cases the implicit volatility as a function of the spot.

2.5 Numerical computation in dimension 2

We will consider for the sake of simplicity the following two dimensional model of assets:

$$\frac{\mathrm{d}S_i}{S_i} = \mu_i \,\mathrm{d}t + \sigma_i \,\mathrm{d}B_i, \quad i = 1, 2,$$

where again the parameter μ_i and σ_i are supposed to be constant and where B_1 and B_2 are two Brownian motion whose correlation is denoted by $\rho \in]-1, 1[$. It means that there exists $\widetilde{B_2}$ independant from B_1 such that

$$B_2 = \rho B_1 + \sqrt{1-\rho^2}\widetilde{B_2}.$$

Suppose that we are given an option whose value $V(S_1, S_2, t)$ depends only on t and S_1 and S_2. Using the same technique than in section 1.1 (the same delta-hedging strategy), it is easy to show that V is solution whose PDE can be written

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma_1^2 x^2 \frac{\partial^2 V}{\partial x^2} + \frac{1}{2}\sigma_2^2 y^2 \frac{\partial^2 V}{\partial y^2} + \rho xy\sigma_1\sigma_2 \frac{\partial^2 V}{\partial x \partial y} + r(x\frac{\partial V}{\partial x} + y\frac{\partial V}{\partial y}) - rV = 0,$$

where $r > 0$ is the interest rate and where $(x, y) \in \mathbb{R}^+$ and $t \in (0, T)$. We will consider the time changed version of this equation ($t \to T - t$) and will discretize it using finite differences. Le us set

$$\Delta x = \frac{L_1}{M_1 + 1}, \quad \Delta y = \frac{L_2}{M_2 + 1}, \quad \Delta t = \frac{T}{N}.$$

Here M_1 and M_2 are such that we have exactly $M_1 \times M_2$ points strictly inside the computation domain (cf. figure 2.7).

$$(x_i, y_j) = (i\Delta x, j\Delta y), \quad (i, j) \in \{2, \ldots, M_1 + 1\} \times \{2, \ldots, M_2 + 1\}.$$

$$u(x_i, y_j, t_n) \longrightarrow u_{i,j}^n$$

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j, t_n) \longrightarrow \frac{1}{\Delta x^2}(u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n)$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j, t_n) \longrightarrow \frac{1}{\Delta y^2}(u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n)$$

$$\frac{\partial^2 u}{\partial x \partial y}(x_i, y_j, t_n) \longrightarrow \frac{1}{\Delta x \Delta y}\left(2u_{i,j}^n + u_{i-1,j+1}^n + u_{i+1,j-1}^n - u_{i,j+1}^n - u_{i,j-1}^n - u_{i+1,j}^n - u_{i-1,j}^n\right))$$

Choice for the first order operator:

$$(C_x) \quad \frac{\partial u}{\partial x}(x_i, y_j, t_n) = \frac{1}{2\Delta x}(u_{i+1,j}^n - u_{i-1,j}^n)$$

$$(D_x^+) \quad \frac{\partial u}{\partial x}(x_i, y_j, t_n) = \frac{1}{\Delta x}(u_{i+1,j}^n - u_{i,j}^n)$$

$$(D_x^-) \quad \frac{\partial u}{\partial x}(x_i, y_j, t_n) = \frac{1}{\Delta x}(u_{i,j}^n - u_{i-1,j}^n)$$

With the same notations for the derivatives with respect to y.

Numerotation of the nodes:

We choose here to integrate the outer nodes (nodes that are on the frontier) in the global numerotation of the unknowns.

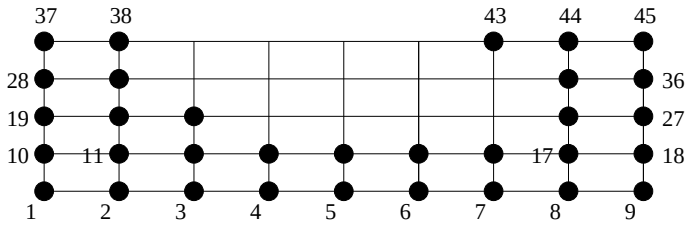


Figure 2.7: $M_1 = 7$, $M_2 = 3$

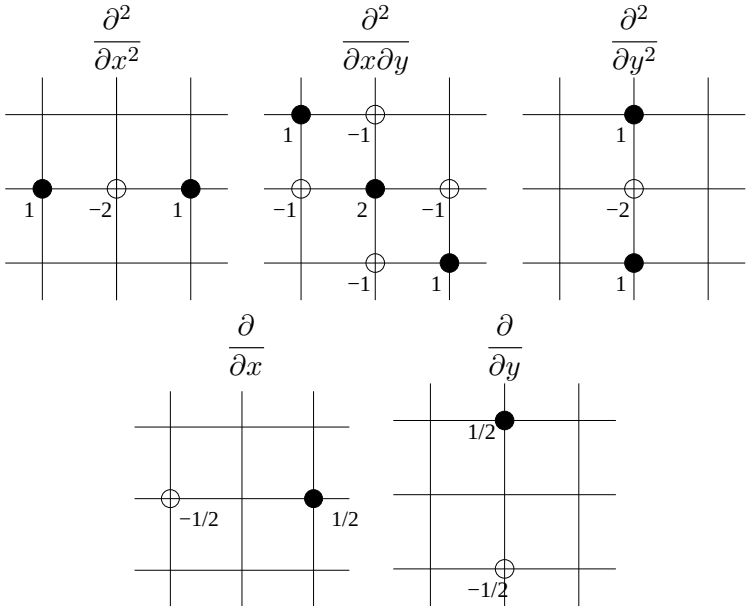
More precisely the unknowns $\{u^n_{i,j}\}_{1\leq i\leq M_1+2, 1\leq j\leq M_2+2}$ are represented by a vector $U^n = {}^t[U^n_1, \ldots, U^n_{(M_1+2)(M_2+2)}]$ where

$$U^n_k = u^n_{i,j} \Longleftrightarrow k = (M_1+2)(j-1)+i. \tag{2.5.37}$$

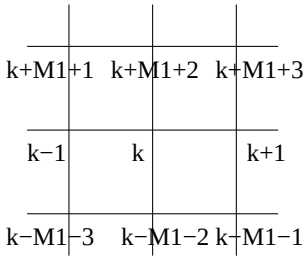
Time scheme : θ-scheme, $\theta \in [0,1]$.

$$\frac{U^{n+1}-U^n}{\Delta t} = (1-\theta)BU^n + \theta BU^{n+1}.$$

The matrix B has a "block structure" (which is typical). Its computation and storage will be done in two independant steps: a matrix B_{int} for the inner nodes (which represent the PDE) and a matrix B_{ext} for the outer nodes (which take into account the boundary conditions). Let us start with B_{int}. The storage of B_{int} are done following the *local* description of the finite differences used:



The step from *local* numerotation to *global* one are done using:



We are doing a loop (i,j) over the inner nodes of the mesh. We compute the correspond number k thanks to (2.5.37). This number corresponds to the number of the line of the matrix we compute. Non vanishing term of the matrix are hence placed and computed thanks to the coefficients of the local grids. Sparse storage is done using special functions of SCILAB[1]. Hence, we have computed `Bint`.

Now, we will add line in the matrix in order to take into account the boundary conditions. Lines will be different depending on what boundary conditions we use. Let k be a number corresponding to an outer points (ex: 9,18,27,36 and 45 on Fig. 2.7).

- *Dirichlet boundary condition* : Suppose that the value of u is set at the node k, then the line k contains 1 on the diagonal and zero everywhere else.
- *Neumann condition* : Suppose that the value of $\dfrac{\partial u}{\partial n}$ has to be set at this node. For example, $\dfrac{\partial u}{\partial y} = 0$. Then the line k contains $1/\Delta y$ and $-1/\Delta y$ at the appropriate nodes.

Hence, we have computed a matrix `Bext`.

The reader can refer to Appendix A.2 where a scilab script compute the two matrices.

We give an example of the matrix which discretize the operator $\frac{\partial^2}{\partial x \partial y}$ for $M_1 = M_2 = 2$:

```
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.  - 1.    1.    0.  - 1.    2.  - 1.    0.    1.  - 1.    0.    0.    0.    0.    0.    0. !
!  0.    0.  - 1.    1.    0.  - 1.    2.  - 1.    0.    1.  - 1.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.  - 1.    1.    0.  - 1.    2.  - 1.    0.    1.  - 1.    0.    0. !
!  0.    0.    0.    0.    0.    0.  - 1.    1.    0.  - 1.    2.  - 1.    0.    1.  - 1.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
```

[1] Voir le lien `http://scilabsoft.inria.fr/` for the official web page.

Only 4 lines has non nul element : it corresponds to the inner points of the domain. Example of matrix `Bext` with the same data:

```
-->full(Bext)
 ans  =

!   1.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.  - 1.    0.    0.    0.    1.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.  - 1.    0.    0.    0.    1.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.  - 1.    1.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.    0.    0.    1.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.  - 1.    1.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    1.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.  - 1.    1.    0.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    1.    0.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.  - 1.    0.    0.    0.    1.    0.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.  - 1.    0.    0.    0.    1.    0. !
!   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.  - 1.    1.    0. !
```

The linear system can be written in a first step:

$$(I_{int} - \theta \Delta t B_{int}) U^{n+1} = F_{int},$$

with

$$F_{int} = (I_{int} + (1-\theta)\Delta t B_{int}) U^{n}.$$

But written like that, there are more unknowns (number of columns = total number of points) than equations (number of line = number of inner points where is written the equation). We complete the system in

$$(I_{int} - \theta \Delta t B_{int} + B_{ext}) U^{n+1} = F_{int} + F_{ext},$$

where F_{ext} contains the values of u or its derivatives that are to be set for the outer nodes.

2.5.1 European case

We are now going to compute the price of an European Put over the minimum of two assets. Here the payoff function $h(S_1, S_2) = (K - \min(S_1, S_2))_+$. We use the scilab script in Appendix A.2. We show in Fig. 2.8 the solution V at time $T = 1$ as a function of S_1 and S_2. Parameters of the models has been set to $r = 0.1$, $\sigma_1 = \sigma_2 = 0.2$, $K = 10 = S_1(0) = S_2(0)$, $rho = 0.5$, $T = 1$.

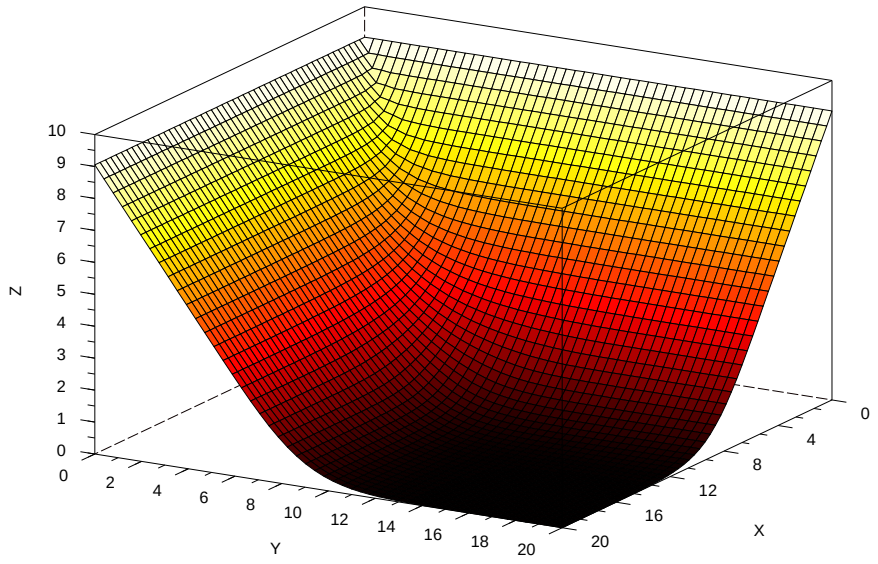


Figure 2.8: Value of the Put over the minimum of two assets in a 2d Black & Scholes model. Here, parameters are $r = 0.1$, $\sigma_i = 0.2$, $\rho = 0.5$, $K = 10 = S_1(0) = S_2(0)$ and $T = 1$.

Chapter 3

Finite Elements Methods for PDE

References about Finite Element Methods: [1], [7] for the fundamentals. A good introduction: [3].

3.1 Toy model 1: Dirichlet problem

Let be f be a continuous function on $(-1,1)$. We are faced to three problems: to find the solution u of each of the three following cases.

$$(\mathcal{D}) \begin{cases} u \text{ twice differentiable such that} \\ -u''(x) = f(x), \quad \text{for any } x \in (-1,1), \\ \text{with the boundary conditions } u(-1) = u(1) = 0. \end{cases}$$

$$(\mathcal{V}) \begin{cases} u \in \mathcal{E} \\ \\ (u', v') = (f, v), \quad \text{for any } v \in \mathcal{E}, \end{cases}$$

$$(\mathcal{M}) \begin{cases} u \in \mathcal{E} \\ \\ \inf_{v \in \mathcal{E}} F(v) = F(u), \end{cases}$$

where we have set

$$\mathcal{E} = \{v : (-1,1) \to \mathbb{R} \mid v(-1) = v(1) = 0,\ v, v' \text{ continue and piecewise continue}\},$$

and where (f,g) denotes the scalar product in $L^2(-1,1)$ the space of squared integrable functions on $(-1,1)$, i.e.

$$(f,g) = \int_{-1}^{1} f(x)g(x)\,\mathrm{d}x,$$

and where $F(v) = \displaystyle\int_{-1}^{1} |v(x)|^2\,\mathrm{d}x - \int_{-1}^{1} f(x)v(x)\,\mathrm{d}x = \frac{1}{2}(v', v') - (f, v)$ for any $v \in \mathcal{E}$.

Theorem 3.1.1 *Under suitable regularity assumptions on the solution u, the problems $(\mathcal{D})$, $(\mathcal{V})$ and $(\mathcal{M})$ are equivalent.*

Proof

We can first show that $(\mathcal{D})$ and $(\mathcal{V})$ are equivalent. Indeed starting from $(\mathcal{D})$, multiplying the differential equation by $v \in \mathcal{E}$ and integrating by part leads to $(\mathcal{V})$. Conversely, starting from $(\mathcal{V})$, if we know moreover that u is twice differentiable (in fact we can demonstrate this : it is a regularizing property) then we can do the IPP in the other sense.

Let us now show that $(\mathcal{M})$ implies $(\mathcal{V})$. Let u be such that $(\mathcal{M})$ holds. By definition, the function $g(\varepsilon) = F(u + \varepsilon v)$ has a minimum in $\varepsilon = 0$. By calculus, g is differentiable and $g'(0) = (u', v') - (f, v)$. Hence $g'(0) = 0$. Conversely, let u be such that $(\mathcal{V})$ hold and let $v \in \mathcal{E}$. We set $w = v - u$ or $v = u + w$. Expanding the term $F(v) = F(u + w)$ leads to

$$\begin{aligned} F(v) &= \frac{1}{2}(u', u') + (u', w') + \frac{1}{2}(w', w') - (f, u) - (f, w) \\ &= F(u) + (u', w') - (f, w) + \frac{1}{2}(w', w') \\ &= F(u) + \frac{1}{2}(w', w') \end{aligned}$$

where we have used that $\mathcal{E}$ is a linear space, i.e. $w = v - u \in \mathcal{E}$ and hence satisfies $(\mathcal{V})$. Finally, since $(w', w') \geq 0$, we can conclude. ∎

We are making now some important remarks. The problem $(\mathcal{V})$ is called the **variational formulation** of the differential problem $(\mathcal{D})$. Getting from $(\mathcal{V})$ to $(\mathcal{D})$ required regularity results about the solution and also the fact that

$$\int_{-1}^{1} (u'' + f)\, v \, \mathrm{d}x = 0 \quad \forall v \in \mathcal{E} \quad \text{implies } u'' + f = 0.$$

Saying it differently, it means that the space $\mathcal{E}$ must be "larger enough".

Giving a variational formulation relies here on the construction of the space $\mathcal{E}$. We are no more looking after approximated values of the solution at some nodes (as we do in Finite Difference Methods) but we search a function as an element (i.e. a point) belonging to some functional space. It is the main feature of the Finite Element Method that we will briefly describe hereafter.

In order to go further, we will change a bit the functional framework and give a more general setting. Indeed, when f is no more enough regular (let say continuous), the space $\mathcal{E}$ is not large enough with respect to the topology induced by the L^2 norm on $(-1, 1)$. Thus we have to deal with its completion: an Hilbert space.

3.2 Abstract framework and discretization

Let V be a real Hilbert space whose scalar product is denoted by $(\cdot, \cdot)$ and its norm by $\|\cdot\|_V$. Let $a : V \times V \to \mathbb{R}$ be a blilinear symmetric continue V-elliptic form on V. It means that the following holds:

(*i*) $a(u, v) = a(v, u)$ for any u and v in V;

(*ii*) $a(\lambda u + \mu v, w) = \lambda a(u, w) + \mu a(v, w)$ for any $\lambda, \mu \in \mathbb{R}$ and $(u, v, w) \in V^3$;

(*iii*) There exists $M > 0$ such that $|a(u,v)| \leq M\|u\|_V\|v\|_V$ for any $(u,v) \in V^2$;

(*iv*) There exists $\alpha > 0$ such that $a(u,u) \geq \alpha\|u\|_V^2$, for any $u \in V$.

We call **V-ellipticity** the last assumption. Let $L : V \to \mathbb{R}$ be a continuous linear form on V. It means that

(*i*) $|L(v)| \leq \Lambda\|v\|$ for any u in V;

(*ii*) $L(\lambda u + \mu v) = \lambda L(u) + \mu L(v)$ for any $\lambda, \mu \in \mathbb{R}$ and $(u,v) \in V^2$.

Following the beginning of the previous section, we set two problems

$$(\mathcal{V}) \begin{cases} u \in V \\ \\ a(u,v) = L(v), \quad \text{for any } v \in V, \end{cases}$$

$$(\mathcal{M}) \begin{cases} u \in \mathcal{V} \\ \\ \inf_{v \in V} F(v) = F(u), \end{cases}$$

where $F(v) = \dfrac{1}{2}a(v,v) - L(v)$ for any $v \in V$.

It is not difficult to see that we recover the previous section's framework by setting $V = H_0^1(-1,1) = \{v, v' \in L^2(-1,1),\ v(-1) = v(1) = 0\}$ and $a(u,v) = \int_{-1}^{1} u'(x)v'(x)\,\mathrm{d}x$. In this case the V-ellipticity on a relies on the Poincaré Lemma.

We have the following result whose proof relies deeply on the Hilbert structure of the functional space V (Lax-Milgram Theorem):

Theorem 3.2.1 *Problems $(\mathcal{V})$ and $(\mathcal{M})$ are equivalent and there exists an unique solution u for these two problems. Moreover, we have*

$$\|u\|_V \leq \frac{\Lambda}{\alpha}.$$

Let now discretize the problem $(\mathcal{V})$. Let $h > 0$ and let V_h be **a finite dimensional subspace** of V (we say that the approximation is *V-conform*) such that $\dim(V_h)$ tends to infinity when h tends to 0. To be specific, we set

$$V_h = \mathrm{span}(\varphi_1, \ldots, \varphi_M).$$

The problem $(\mathcal{V})$ is replaced by

$$(\mathcal{V}_h) \begin{cases} u_h = \sum_{i=1}^{M} u_i\varphi_i \\ \\ a(u_h, \varphi_j) = L(\varphi_j), \quad \text{for any } j \in \{1, \ldots, M\}. \end{cases}$$

It gives a linear system

$$AU = b,$$

where $A = [a(\varphi_i, \varphi_j)]$, $U = {}^t[u_1, \ldots, u_M]$ and $b = {}^t[L(\varphi_1), \ldots, L(\varphi_M)]$. Then we have easily

Theorem 3.2.2 *A is positive definite. There exists an unique solution $u_h \in V_h$ and*

$$\|u_h\|_V \leq \frac{\Lambda}{\alpha}.$$

This is a stability result. Hence, a Finite Element Method is stable in this case as soon as soon the continuous problem is elliptic. Note that it is not necessarly the case for Finite Different Methods.

We study now the error between u and u_h. Indeed, we have

Theorem 3.2.3 *For any $v_h \in V_h$, we have*

$$\|u - u_h\|_V \leq \frac{M}{\alpha}\|u - v_h\|_V.$$

Proof

First, since a is in the two formulation $(\mathcal{V})$ and $(\mathcal{V}_h)$, we have by substraction:

$$a(u - u_h, v_h) = 0 \quad \text{for any } v_h \in V_h.$$

We can do that because the approximation is V-conform ($v_h \in V_h \subset V$). We see that $u - u_h$ is **orthogonal to** V_h with respect to the scalar product a.

Now, the result relies on the relation $a(u - u_h, u - u_h) = a(u - u_h, u - v_h + v_h - u_h) = a(u - u_h, u - v_h)$ since $v_h - w_h \in V_h$. We conclude by continuity of a (assertion (iii)) and V-ellipticity (assertion (iv)). ∎

This last result is fundamental. It says that the error $\|u - u_h\|_V$ can be controlled by any other approximation v_h of u in V_h, its interpolant for example. Something which **do not depend** on the original problem but only on the space V_h and the way it "approaches" V.

3.3 Toy model 2: Obstacle problem

As we have seen in the introduction, the problem of the valuation of an American option is very similar with a free boundary problem for the associated Black & Scholes PDE. Indeed, the domain $\mathbb{R}^+ \times (0, T)$ can be separated in two region, the first where one should exercise the option and the case where one should hold the option. Nevertheless, any frontier cannot be chosen. In fact, we can show that at each time t there exists an unique asset price, called the *optimal exercise price* which divide $\mathbb{R}^+$ (i.e. the S-axis) into these two regions. Moreover, $V(S, t)$ must be greater than the payoff function otherwise arbitrage possibilities can occur.

In the case of the American put option, one can show that at each time t, we must divide the S axis into two regions: one where one should exercise and

$$V = K - S, \quad \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV < 0,$$

(since $r > 0$) and the other where $V(S, t)$ satisfies locally the Black & Scholes PDE

$$V > K - S, \quad \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0.$$

We will discuss in this section about a canonical version of this problem: the obstacle problem. The main simplifications are that the problem will be time independant and the PDE will be reduced to the second derivative in space. We will see that this can be written in another form: **a variational inequality**. We will see that this new formulation is particularly well suited for the Finite Element method. The reason for this is that such a formulation eliminates the explicit dependance of the solution with respect to the free boundary during the solution process.

Let f be a regular enough function defined on $(-1,1)$ such that $f'' < 0$. We are looking for the solution of the following problem: find u, regular enough, defined on the same interval such that $u(-1) = u(1) = 0$, $u \geq f$ on $(-1,1)$ and such that $u > f$ implies $u'' = 0$. One can write this problem as a so-called linear complementary problem,

$$u''(u-f) = 0, \quad -u'' \geq 0, \quad u \geq f, \tag{3.3.1}$$

with the condition

$$u(-1) = u(1) = 0. \tag{3.3.2}$$

We now give a variational formulation of (3.3.1)–(3.3.2). Let $\mathcal{F}$ be the set

$$\mathcal{F} = \{v : (-1,1) \to \mathbb{R} \mid v(-1) = v(1) = 0,\ v \geq f, v, v' \text{ continue and piecewise continue}\}.$$

Then for any $v \in \mathcal{F}$, we multiply $-u''$ by $v - f$ and integrate it over the interval $(-1,1)$, we obtain $\int_{-1}^{1} -u''(v-f)\,\mathrm{d}x \geq 0$. Moreover, one has $\int_{-1}^{1} -u''(u-f)\,\mathrm{d}x = 0$. Hence, by addition and integration by part, we get

$$\int_{-1}^{1} -u''(v-u)\,\mathrm{d}x = \int_{-1}^{1} u'(v'-u')\,\mathrm{d}x \geq 0.$$

The linear complementary problem implies the following variational problem (in fact, one can show that there are equivalents): find u such that

$$(\mathcal{IV}) \begin{cases} u \in \mathcal{F} \\ \\ (u', v'-u') \geq 0, \quad \text{for any } v \in \mathcal{F}. \end{cases}$$

Again, as in the previous section we will write $(\mathcal{IV})$ in a more general setting:

$$(\mathcal{IV}) \begin{cases} u \in K \\ \\ a(u, v-u) \geq L(v-u), \quad \text{for any } v \in K, \end{cases}$$

where a and L are the same as in Section 3.2 and where K is a nonempty closed convex subset of V.

Then we can show

Theorem 3.3.1 (Stampacchia) *Under suitable assumptions on a and L (see Section 3.2), we can show that there exists an unique u solution of (IV). Moreover, u solves the following constrained optimization problem:*

$$\inf_{v \in K} F(v) = F(u), \tag{3.3.3}$$

where $F(v) = \frac{1}{2}a(u,u) - L(v)$.

Proof

We give the proof since it will give some clues about the choice of the numerical algorithms used to approximate a solution of $(\mathcal{IV})$.

Since L is linear continuous on V, by the Riesz theorem, we know that there exists $f \in V$ such that

$$L(v) = (f, v), \quad \forall v \in V.$$

Moreover, for the same reasons, one can easily construct a linear operator A from V into V' such that

$$a(u, v) = (Au, v) \quad \forall u, v \in V.$$

We are looking for $u \in K$ such that

$$(Au, v - u) \geq (f, v - u) \forall v \in K,$$

or identically such that

$$(\rho(f - Au) + u - u, v - u) \leq 0 \quad \forall v \in K,$$

where $\rho > 0$ will be chosen later. The last inequality is precisely the caracterization of u as the orthogonal projection of $\rho(f - Au) + u$ on K, it means with obvious notation

$$u = P_K(\rho(f - Au) + u).$$

The solution u seems to be a fixed point of $\phi(v) = P_K(\rho(f - Av) + v)$. We show now that we can choose a parameter $\rho > 0$ such that the application $\phi : V \to V$ is contractante. Indeed, we can write for any $v, w \in V$:

$$\begin{aligned}
\|\phi(v) - \phi(w)\|_V^2 &= \|P_K(\rho(f - Av) + v) - P_K(\rho(f - Aw) + w)\|_V^2 \\
&\leq \|\rho A(w - v) + v - w)\|_V^2 \\
&= \rho^2(A(v - w), v - w) - 2\rho(A(v - w, v - w) + \|v - w\|_V^2 \\
&\leq M\rho^2\|v - w\|_V^2 - 2\rho\alpha\|v - w\|_V^2 + \|v - w\|_V^2 \\
&= \|v - w\|_V^2 \left(M\rho^2 - 2\alpha\rho + 1\right),
\end{aligned}$$

where we have used the symmetry of a in the third line and the assumption of continuity and V-ellipticity in the fourth line. Now it is possible to find $\rho = \alpha/M > 0$ e.g. such that ϕ is V-contractante. ■

The last result can justify the use of Projected Gradient Method applied to the optimization (3.3.3) (See subsection 3.4.2).

3.4 Numerical implementation of options in dimension 1

Two numerical experiments about options are introduced in this section. There are written as problems. We can find in Appendix C, SCILAB scripts which compute options prices in dimension 1 using some examples of finite element methods.

3.4.1 The European case

We recall that the price $u(x,t)$ of an European Put is obtained by resolving the following PDE:

$$\left(\frac{\partial u}{\partial t} - \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 u}{\partial x^2} - r(t)x\frac{\partial u}{\partial x} + r(t)u\right)(t,x) = 0, \quad (t,x) \in]0,T] \times \mathbb{R}^+, \tag{3.4.4}$$

with the initial condition

$$u(0,x) = \max(K - x, 0), \quad x \in \mathbb{R}^+. \tag{3.4.5}$$

Let $L > 0$. We want to solve numerically (3.4.4)–(3.4.5) on the interval $[0,L]$ by means of a finite element method using piecewise second order polynomial functions, suitable Dirichlet boundary conditions and an Euler scheme in time.

For any $t > 0$, we denote by a_t the following bilinear symmetric form

$$a_t(\varphi,\psi) = \frac{1}{2}\int_0^L \sigma^2(t,x)x^2 \frac{\partial\varphi}{\partial x}\frac{\partial\psi}{\partial x}\,\mathrm{d}x + \int_0^L r\varphi\psi\,\mathrm{d}x,$$

and by b_t the bilinear form

$$b_t(\varphi,\psi) = \int_0^L \left(\sigma(t,x)\frac{\partial\sigma}{\partial x}(t,x)x^2 + \sigma^2(t,x)x - rx\right)\frac{\partial\varphi}{\partial x}\psi\,\mathrm{d}x.$$

Let M an integer. We set $M+1$ points on $[0,L]$: $0 = x_0 < x_1 < \ldots < x_M = L$. We set $h_i = x_{i+1} - x_i$, $h = \max h_i$, and

$$V_h = \{v \text{ continuous on } [0,L],\ v(L) = 0,\ \forall i,\ v|_{[x_i,x_{i+1}]} \text{ piecewise parabolic}\}.$$

Let also N an integer, we set $\Delta t = T/(N+1)$ et $t_k = k\Delta t$.

1. a) Show that if $v \in V_h$, $v|_{[x_i,x_{i+1}]}$ can be found by its values at the points $x_i, x_{i+\frac{1}{2}}, x_{i+1}$ where $x_{i+\frac{1}{2}} = (x_i + x_{i+1})/2$.

1. b) Give the expressions of the second order polynomial functions $\{f_i,\ i = 1,2,3\}$ on the interval $(0,1)$ such that $f_i(\alpha_j) = \delta_{i,j}$ where $\alpha_j = 0, 1/2, 1$, i.e. $f_i(\alpha_j) = 0$ if $i \neq j$ et 1 otherwise.

1. c) Show that V_h is a finite dimensional linear space whose dimension is $2M$. Show a basis $\{\varphi_i\}_{i=1,\ldots,2M}$ for V_h.

2. a) We want to approximate u at time t_k by a function $u_h^k \in V_h$. We denote by $\{u_i^k\}_{i=1,\ldots,2M}$ the composants of u_h^k in the basis $\{\varphi_i\}$ and U^k the column-wise associated vector.

Write an implicit Euler scheme for (3.4.4) and its variational formulation in V_h.

2. b) Write a fully discretization of (3.4.4) by means of finite elements such as

$$A_{k+1}U^{k+1} = MU^k, \quad 0 \le k \le N,$$

where A_{k+1}i a matrix and where M is the mass matrix (i.e. associated to the bilinear form $(\varphi,\psi) \mapsto \int_0^L \varphi\psi\,\mathrm{d}x$).

2. c) How discretize (3.4.5) by means of finite element ?

3.4.2 The American case : projected Gradient method

We consider the solution $(t,x) \mapsto u(t,x)$ of the problem

$$\begin{cases} \dfrac{\partial u}{\partial t} - \dfrac{\sigma^2(t,x)}{2} x^2 \dfrac{\partial^2 u}{\partial x^2} - r(t) x \dfrac{\partial u}{\partial x} + r(t) u \geq 0, \\ u(t,x) \geq (K-x)_+, \\ \left(\dfrac{\partial u}{\partial t} - \dfrac{\sigma^2(t,x)}{2} x^2 \dfrac{\partial^2 u}{\partial x^2} - r(t) x \dfrac{\partial u}{\partial x} + r(t) u \right) (u - (K-x)_+) = 0, \end{cases} \tag{3.4.6}$$

for any $(t,x) \in]0,T] \times]0,L[$ where $L > K > 0$ et $T > 0$.

It is well known that $u(T,S_0)$ is the price of an American put whose maturity is $T > 0$, the strike price K and the spot S_0.

We add the following boundary conditions:

$$u(0,x) = (K-x)_+, \quad x \in]0,L[,$$

and

$$u(t,L) = 0, \quad t \in]0,T].$$

We are given the points $x_0 = 0 < x_1 < \cdots < x_M = L$ and we set $h = \max(x_{i+1} - x_i)$ and

$$V_h = \{v \text{ continuous on } [0,L],\ v(L) = 0 \text{ et } \forall i,\ v\mid_{[x_i,x_{i+1}]} \text{ linear.}\}.$$

Finally, we set $\Delta t = T/(N+1)$ et $t_k = k\Delta t$.

1. Show that V_h is a linear space whose dimension is M and describe a nodal basis $\{\varphi_i\}_{1 \leq i \leq M}$ of V_h, i.e. the functions of V_h whose value at node i is 1 and 0 everywhere else.

2. We seek for an approximation of u at time t_k by a function u_h^k of V_h. Let us denote by $\{U_i^k\}_{1 \leq i \leq M}$ the composants of u_h^k in the previous basis of V_h et $U^k = {}^t[U_1^k, \ldots, U_M^k]$ the associated $\mathbb{R}^M$ vector. We can write

$$u_h^k(x) = \sum_{i=1}^{M} U_i^k \varphi_i(x).$$

Write an Euler scheme for the discretization of (3.4.6) by means of finite elements which can be written under the matrix form

$$\begin{cases} A_{k+1} U^{k+1} + B_k U^k \geq 0, \quad \text{in } \mathbb{R}^M, \\ U^{k+1} \geq H, \quad \text{in } \mathbb{R}^M, \\ (A_{k+1} U^{k+1} + B_k U^k, U^{k+1} - H) = 0, \\ U^0 = H. \end{cases} \tag{3.4.7}$$

where $U \geq V$ means $U_i \geq V_i$ for every choice of i and where $(\cdot,\cdot)$ denotes the euclidean scalar product scalaire in $\mathbb{R}^M$. Be careful to choose a scheme in time such that A_{k+1} be a **symmetric** matrix (positive definite). Describe the vector H.

3. We set $\mathcal{K} = \{X \in \mathbb{R}^M \mid X \geq H\}$. Show that solving (3.4.7) is equivalent to find U^{k+1} in $\mathcal{K}$ such that

$$(-A_{k+1}U^{k+1} - B_kU^k, V - U^{k+1}) \leq 0, \quad \text{for any } V \in \mathcal{K}. \tag{3.4.8}$$

4. (a) Let W^{k+1} be the vector of $\mathbb{R}^M$ solution of the linear system $A_{k+1}W^{k+1} = -B_kU^k$. Proove that owing to (3.4.8), U^{k+1} can be interpreted as the orthogonal projection of W^{k+1} on $\mathcal{K}$ for the scalar product associated to the matrix A_{k+1}. It means the product $(X, Y)_A = (A_{k+1}X, Y)$.

(b) Explain why such a projection exists.

5. Let $J : \mathbb{R}^M \to \mathbb{R}$ defined by $J(X) = \frac{1}{2}\|X - W^{k+1}\|_A^2$, where we set $\|X\|_A = (A_{k+1}X, X)^{1/2}$. Show that $J'(X) = A_{k+1}(X - W^{k+1})$ and that (3.4.7) is equivalent to

$$\begin{cases} U^{k+1} \in \mathcal{K}, \\ J(V) \geq J(U^{k+1}), \quad \forall V \in \mathcal{K}. \end{cases} \tag{3.4.9}$$

6. Show that there exists an unique solution to the problem (3.4.9).

7. At each time step, we want to approximate the solution of (3.4.9) by the following scheme: a vector U_0 de $\mathbb{R}^M$ to be chosen being given together with a positive real $\alpha > 0$ to be chosen later, we set, for every $n \geq 0$, $U_{n+1} = \Pi_{\mathcal{K}}(U_n - \alpha J'(U_n))$, where $\Pi_{\mathcal{K}}$ is the orthogonal projetion in $\mathbb{R}^M$ on $\mathcal{K}$ for the canonical scalar product $(\cdot, \cdot)$.

Show that $Y = \Pi_{\mathcal{K}}(X)$ is such that $y_i = \max(x_i - h_i, 0) + h_i$ for any i.

8. We denote by $0 < \lambda_1 < \cdots < \lambda_M$ the eigenvalue of A_{k+1} in the non decreasing order.

(a) Show that if $\alpha < \lambda_1/\lambda_N^2$ then the application $S : X \to X - \alpha J'(X)$ is strictly contractante. Show that this result remains true under the weaker hypothesis $\alpha < 1/\lambda_N$.

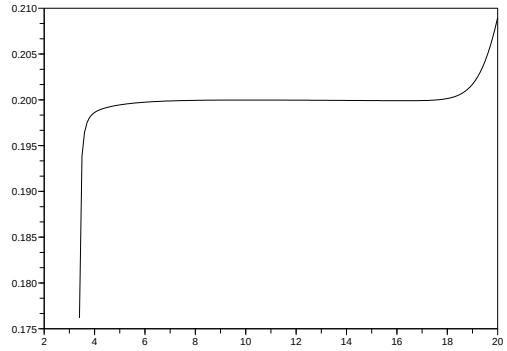
(b) Show that under this last hypothesis, the series $\{U_n\}$ converges toward U^{k+1}.

9. We find U^{k+1} at each time step the problem (3.4.7) with the initial data U^k by means of the gradient algorithm given at the question 7. We set $U^{k+1} = U_n$ after some iteration of this algorithm. Is this algorithm stable whatever the choice of Δt is made ?

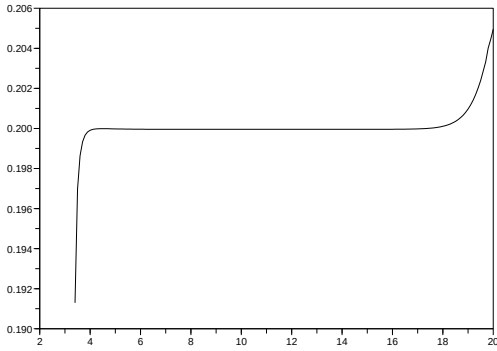
3.4.3 Some numerical results

We show here some numerical results derived from the Finite Element Methods applied to European and American options in dimension 1. Two types of finite elements were used: "hat" functions and cubic splines. We see on Figures (3.1)–(3.2) that the Finite Element Method based on the cubic splines (with the same size of the mesh) computes price much better (in the sense of the implicit volatilities) than "hat functions" does.

We present just below numerical results about the convergence of the projected gradient method for the American options using the same finite elements as in the European case just above. In Fig. 3.3, we can the error of the "inner" iterate U_n (see question 7 in Subsection 3.4.2) as respect to n. At each time step, we can see that the line is broken. It means that the gradient is restarted with the new updated second member. We can see also that more iteration is needed in the cubic spline case than in the "hat" functions case. Indeed more precision gives more information about the free frontier at each time step. It is why we need more iteration.

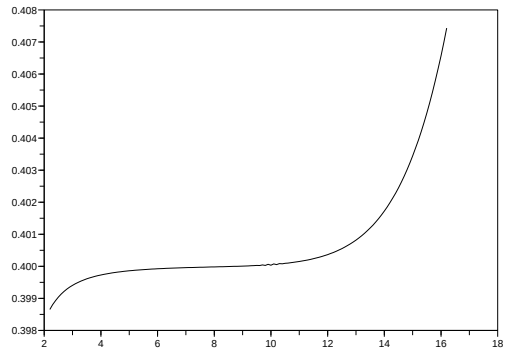


(a)

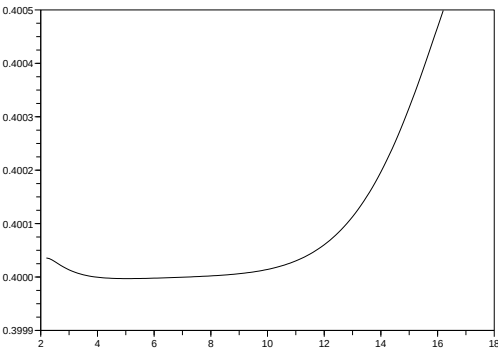


(b)

Figure 3.1: Implicit Volatility as a function of the spot in the case: (a) "hat" functions; (b) cubic splines. European put option $r = 0.1$, $\sigma = 0.2$, $K = 10$, $T = 1$. In both case $M = 200$ elements on $(0, 20)$.

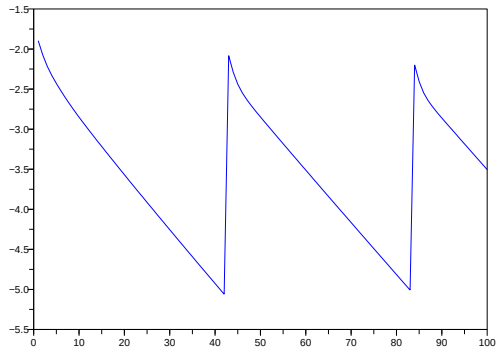


(a)

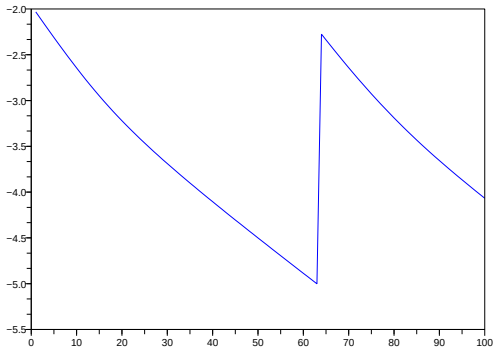


(b)

Figure 3.2: Implicit Volatility as a function of the spot in the case: (a) "hat" functions; (b) cubic splines. European put option $r = 0.1$, $\sigma = 0.4$, $K = 10$, $T = 1$. In both case $M = 200$ elements on $(0, 20)$.



(a)



(b)

Figure 3.3: Error $\|U_{n+1} - U_n\|_2$ with respect to n: iteration of the projected gradient method described at question 7 in Subsection 3.4.2.

Chapter 4

Asian options

We end these notes by an example of path-dependant exotic options: the Asian options and try to give some clues about its numerical implementation. This problem is derived from [2]. See also [6] for the original derivation of the equation.

Let K and T be two positive real. We are interested by the pricing of an Asian option with fixed strike K and maturity T (European contract). Hence, the payoff function depends now on the whole trajectory of the risky asset $S = (S_t)$ ($\neq$ Vanilla options):

$$V_t = \mathbb{E}\left(e^{-r(T-t)}\left(\frac{1}{T}\int_0^T S_t\,\mathrm{d}t - K\right)_+ \mid \mathcal{F}_t\right), \tag{4.0.1}$$

where $\mathcal{F}_t$ denotes the filtration of $B = (B_t)$.

We set for any $t \geq$, $\bar{S}_t = \frac{1}{t}\int_0^t S_s\,\mathrm{d}s$. The same techniques which has allowed us to derive the Black & Scholes PDE in the first Chapter show here that $V_t = v(t, S_t, \bar{S}_t)$ where $v = v(t, x, y)$ satisfies the PDE:

$$\frac{\partial v}{\partial t} + rx\frac{\partial v}{\partial x} + \frac{\sigma^2 x^2}{2}\frac{\partial^2 v}{\partial x^2} + \left(\frac{x-y}{t}\right)\frac{\partial v}{\partial y} - rv = 0, \quad (x,y) \in \mathbb{R}_+^*, \quad t \in [0, T[, \tag{4.0.2}$$

with the terminal condition:

$$v(T, x, y) = (y - K)_+, \quad x > 0, \quad y > 0. \tag{4.0.3}$$

1. We set for any $t \in [0, T]$ and $x, y > 0$,

$$z = \frac{K - yt/T}{x}, \quad \text{and } f(t, z) = \frac{1}{x}v(t, x, y).$$

Show that f is the solution of the following PDE:

$$\frac{\partial f}{\partial t} + \frac{\sigma^2 z^2}{2}\frac{\partial^2 f}{\partial z^2} - \left(\frac{1}{T} + rz\right)\frac{\partial f}{\partial z} = 0, \quad z \in \mathbb{R}, \tag{4.0.4}$$

with the terminal condition:

$$f(T, z) = z_- = \max(0, -z). \tag{4.0.5}$$

Figure 4.1: Domaine de calcul

3. (a) We set now $g(t,x) = f(t, x - t/T)$. Show that g satisfies the following PDE:

$$\frac{\partial g}{\partial t} + \frac{\sigma^2}{2}\left(x - \frac{t}{T}\right)^2 \frac{\partial^2 g}{\partial x^2} - r\left(x - \frac{t}{T}\right)\frac{\partial g}{\partial x} = 0, \quad x \in \mathbb{R}, \quad t \in [0, T[, \tag{4.0.6}$$

with the terminal condition:

$$g(T,x) = (1-x)_+. \tag{4.0.7}$$

3. (b) Let $\widetilde{f}$ the solution of (4.0.4) obtained with the terminal data $\widetilde{f}(T,z) = -z$. Show that

$$\widetilde{f}(t,z) = \frac{1}{rT}\left(1 - e^{-r(T-t)}\right) - z\, e^{-r(T-t)}.$$

Deduce from this that for any $x \le t/T$, we have

$$g(t,x) = \frac{1}{rT}\left(1 - e^{-r(T-t)}\right) - \left(x - \frac{t}{T}\right) e^{-r(T-t)}. \tag{4.0.8}$$

4. We remark that (4.0.6) is equivalent to

$$\frac{\partial g}{\partial t} + \frac{\sigma^2}{2}\frac{\partial}{\partial x}\left(\left(x - \frac{t}{T}\right)^2 \frac{\partial g}{\partial x}\right) - (r + \sigma^2)\left(x - \frac{t}{T}\right)\frac{\partial g}{\partial x} = 0, \quad x \in \mathbb{R}, \quad t \in [0, T[, \tag{4.0.9}$$

Let $N \ge 1$ an integer and $\Delta t = 1/N$. We discretize (4.0.9) by means of a Crank–Nicolson scheme with uniform time step Δt. As regard the space, we want to exploit the fact that we know the solution at the left of the line $x = t/T$ (cf (4.0.9). Therefore, we discretize $(0, L)$ as shown in Figure 4.1 and we set $\Delta x = 1/N$ in order to numerically capture the line $x = t/T$. We complete between 1 and L by adding J intervals in order to have $L = (N + J)\Delta x$.

We set $x_i = i\Delta x$ for $1 \le i \le N + J$ et $x_{i\pm 1/2} = (i \pm 1/2)\Delta x$.

What are the order of approximation of the following finite differences ?

(a)

$$\frac{1}{\Delta x}\left\{\left(x_{i+1/2} - t/T\right)^2 \frac{g(t,x_{i+1}) - g(t,x_i)}{\Delta x} - \left(x_{i-1/2} - t/T\right)^2 \frac{g(t,x_i) - g(t,x_{i-1})}{\Delta x}\right\}$$

for $\dfrac{\partial}{\partial x}\left((x - t/T)^2 \dfrac{\partial g}{\partial x}\right)(t, x_i)$ and

(b)

$$\frac{1}{2}\left\{ \left(x_{i+1/2} - t/T\right) \frac{g(t,x_{i+1}) - g(t,x_i)}{\Delta x} + \left(x_{i-1/2} - t/T\right) \frac{g(t,x_i) - g(t,x_{i-1})}{\Delta x} \right\}$$

for $(x - t/T)\,\dfrac{\partial g}{\partial x}(t,x_i)$?

5. We choose Dircihlet boundary conditions on the frontier $x = t/T$ and Neumann type boundary conditions on $x = L$ compatible with the terminal data. Give a second order approximation of the Neumann boundary condition on $x = L$ for (4.0.7)–(4.0.9).

6. Derive the following matrix formulation

$$(I + \Delta t B_k)G^{k+1} = (I - \Delta t \widetilde{B_k})G^k, \quad 0 \le k \le N,$$

where G^k is the vector whose composants are $g(t_k, x_i)$.

Give details about the matrices B_k and $\widetilde{B_k}$ (in particular their sizes).

Appendix A : Some useful Scilab scripts (volimp.sci file)

```
function [y]=myerf(x)
y=0.5*(1+erf(x/sqrt(2)));
endfunction

////////////////////////////////////////
// Put BS closed form : scalar version
////////////////////////////////////////
function [p,p_sig]=BSPut_s(x,K,r,sig,theta) // scalar definition

if (theta<=%eps) then
  p=max(K-x,0);
  p_sig=0;
else
  d1=(log(x/K)+(r+sig^2/2)*theta)/(sig*sqrt(theta));
  d2=d1-sig*sqrt(theta);

  p=K*exp(-r*theta)*myerf(-d2) - x(~(x==0)).*myerf(-d1)             // Prix Put

  p_sig=x*exp(-d1^2/2)*((r+sig^2/2)*theta-log(x/K)) ...   // Drive Put/Vol
      - K*exp(-r*theta-d2^2/2)*((r-sig^2/2)*theta-log(x/K))
  p_sig=p_sig/(sqrt(2*%pi*theta)*sig^2)

end

////////////////////////////////////////////
// Put BS closed form : vectorized version
////////////////////////////////////////////
function [p,p_sig]=BSPut_v(x,K,r,sig,theta)    // vectorized def

// x Vecteur des spots
// sigma vecteur des vol
// p en sortie : vecteur des prix de Puts

sig=abs(sig);
p=zeros(x);
p_sig=zeros(x);
un=ones(x);
zer=zeros(x);

if (theta<=%eps) then
```

```
  p = max( K - x, 0)

else

  p(x==0)     = K*exp(-r*theta)*un(x==0);           // Cas x==0
  p_sig(x==0) = zer(x==0);

  p(sig==0) = max(K*exp(-r*theta) - x(sig==0),0); // Cas sig=0
  p_sig(sig==0) = zer(sig==0);

  reste=~(x==0 | sig==0);

  signo=sig(reste);
  xno=x(reste);

  d1=(log(xno/K)+(r+signo.^2/2)*theta)./(signo*sqrt(theta));
  d2=d1-signo*sqrt(theta);

  p(reste)=K*exp(-r*theta)*myerf(-d2) - xno.*myerf(-d1)                  // Put Price

  p_sig(reste)=xno.*exp(-d1^2/2).*((r+signo.^2/2)*theta-log(xno/K)) ...   // Put Delta
      - K*exp(-r*theta-d2^2/2).*((r-signo.^2/2)*theta-log(xno/K))
  p_sig(reste)=p_sig(reste)./(sqrt(2*%pi*theta)*signo.^2)

end

endfunction

////////////////////////////////////////
//  Implicit volatility from Put prices
////////////////////////////////////////
function [s,y,ind,iternwt,err]=volimpP_v(x,K,r,sig0,theta,Put,tol)

// x    vecteur des spots
// sig0 vecteur des vol initiales du Newton
// Put  vecteur des prix de Put
// Mme taille pour les trois ( + tous vecteurs lignes) !

sig=abs(sig0);
sig1=zeros(sig0);

err=ones(x);

imxnwt=100;

stop=(norm(err)<=tol);

iternwt=1;
while (iternwt<=imxnwt)&(~stop)

  [p,p_sig]=BSPut_v(x,K,r,sig,theta);

  tronc=(abs(p_sig)<=tol);
  err(~tronc)=abs(p(~tronc)-Put(~tronc));
```

```
  sig(~tronc) = sig(~tronc) - (p(~tronc) - Put(~tronc))./p_sig(~tronc);

  stop=(norm(err(~tronc))<=tol)
  iternwt=iternwt + 1;

end

s=sig(~tronc)
y=x(~tronc)

ind=~tronc

endfunction
```

Appendix B : Scilab scripts for Finite Difference methods for pricing

B.1. European and American, dimension 1

```
////////////////////////////////////////////////////
//       CIMPA HANOI 23/04/2007 - 4/05/2007
//   Finite Difference Method for American options
////////////////////////////////////////////////////
clear;

//////////////////////////////
//   MODEL PARAMETER (BS)
//////////////////////////////
r=0.1;
sigma=0.2;

//////////////////////////////////////
//       OPTION PARAMETER
//////////////////////////////////////
// Cf. Option Pricing, Wilmott et al, p. 335
T=1;
K=10;
deff('[payoff]=h(x)',['payoff=max(K-x,0)']);

// Spatial discretization
M=200;
L=2*K;
Deltax=L/(M+1);

// Time discretization
N=100;
Deltat=T/N;
theta=0.5;
imxtmp=N;

// Centrage, dcentrage des drives d'ordre 1 en x
centrex=%T;

// Conditions aux bords
```

```
D = spzeros(M+2,M+2);  // Dirichlet
N = spzeros(M+2,M+2);  // Neumann

D(1,1) = 1.0;          // Dirichlet Boundary Conditions at x=0
N(M+2,M+2) = 1.5;      // Neumann 2nd order at x=L
N(M+2,M+1) = -2.0;
N(M+2,M)   = 0.5;
//N(M+2,M+2) = 1.0;    // Neumann 1st order
//N(M+2,M+1) = -1.0;
//D(M+2,M+2) = 1.0;      // D. B. C. at x=L


////////////////////////////
// DO NOT EDIT BELOW
////////////////////////////
// Elementary matrices
x=linspace(0,L,M+2);
xint=x(2:M+1);

// Sparse definition of the elementary matrices
Lx = spzeros(M+2,M+2);
Gd = spzeros(M+2,M+2);
Gc = spzeros(M+2,M+2);

auxLx = -2.0/Deltax^2*speye(M+2,M+2) + ...
         1.0/Deltax^2*diag(sparse(ones(M+1,1)),1) + ...
         1.0/Deltax^2*diag(sparse(ones(M+1,1)),-1);
auxGd = 1.0/Deltax*diag(sparse(ones(M+1,1)),1) - ...
        1.0/Deltax*speye(M+2,M+2);
auxGc = 0.5/Deltax*diag(sparse(ones(M+1,1)),1) - ...
        0.5/Deltax*diag(sparse(ones(M+1,1)),-1);

Lx = [ zeros(1,M+2) ; auxLx(2:M+1,:) ; zeros(1,M+2)];
Gd = [ zeros(1,M+2) ; auxGd(2:M+1,:) ; zeros(1,M+2)];
Gc = [ zeros(1,M+2) ; auxGc(2:M+1,:) ; zeros(1,M+2)];

Bext = D + N;

Idint = speye(M+2,M+2) - spones(diag(spones(diag(Bext))));

Bint  = 0.5*sigma^2*diag(sparse(x.*x))*Lx - r*Idint;

if centrex==%F then
  Bint = Bint + r*diag(sparse(x))*Gd;
else
  Bint = Bint + r*diag(sparse(x))*Gc;
end;

// Matrix to be inversed
A  = Idint - theta*Deltat*Bint     + Bext;

[hh,rk]=lufact(A);
[P,L,UU,Q]=luget(hh);

// Donne initiale
```

```
U0=h(x)';
// Cash-or-Nothing Call
//U0=1*(x>K)';

//////////////////////
// Time iterates
//////////////////////
itertmp=1;
U=U0;    // European price
V=U0;    // american price

while itertmp<=imxtmp

  // Second member updated at each time iteration (European case)
  Fint    = Idint*U + (1 - theta)*Deltat*Bint*U;
  Fext    = zeros(M+2,1);
  Fext(1) = K*exp(-r*Deltat*itertmp);

  // European price
  U=lusolve(hh,Fint + Fext);

  // Second member updated at each time iteration (American case)
  Fint_am = Idint*V + (1 - theta)*Deltat*Bint*V;
  Fext_am = zeros(M+2,1);
  Fext_am(1)   = U0(1);
  Fext_am(M+2) = U0(M+2);

  // Projected LU : American price
  V = P'*(Fint_am + Fext_am);
  PU0 = P'*U0;
  V = L\V;
  for i=M+2:-1:1,
     V(i) = max(PU0(i),V(i) - sum(UU(i,i+1:M+2)'.*V(i+1:M+2)));
  end
  V = Q'*V;

  xbasc();
  //plot2d(xx,v,leg="Iter="+string(itertmp*Dt),rect=[a,ymin,b,ymax])
  plot2d(x,[V,U,U0],leg="Iter="+string(itertmp*Deltat))

  itertmp = itertmp + 1;
end
```

A.2. European dimension 2

```
////////////////////////////////////////////////////
//       CIMPA HANOI 23/04/2007 - 4/05/2007
//   Finite Difference Method for 2D European options
//
//   Ex : Put on the minimum of two assets
////////////////////////////////////////////////////
clear
```

```
L1=20;
L2=20;
M1=50;
M2=50;
centrex=%T;
centrey=%T;

//////////////////////////////////////////
// Loading of the elementary DF matrices
//////////////////////////////////////////
load('BS2d_assemb_DF_centrex' + string(centrex) + ...
                   '_centrey' + string(centrey) + ...
                   '_L1-' + string(L1)          + ...
                   '_L2-' + string(L2)          + ...
                          '_Mx'+string(M1)      + ...
                          '_My'+string(M2) + '.save');

//////////////////////////
// BS MODEL PARAMETER
//////////////////////////
s1  = 0.2;
s2  = 0.2;
mu1 = 0.;
mu2 = 0.;
rho = 0.5;
r   = 0.1;

////////////////////////////
//   OPTION PARAMETER
////////////////////////////
T=1;
K=10;
deff('[payoff]=h(x,y)',['payoff=max(K - min(x,y),0)']);

////////////////////////////
//   MATRICES
////////////////////////////
wLx=matrix(x'*ones(1,M2+2), (M1+2)*(M2+2),1)^2*s1^2/2;
wMxy=matrix(x'*y,           (M1+2)*(M2+2),1)*rho*s1*s2;
wLy=matrix(ones(M1+2,1)*y,  (M1+2)*(M2+2),1)^2*s2^2/2;

wGx=r*matrix(x'*ones(1,M2+2),(M1+2)*(M2+2),1);
wGy=r*matrix(ones(M1+2,1)*y, (M1+2)*(M2+2),1);

// BOUNDARY CONDITIONS
D = spzeros((M1+2)*(M2+2),(M1+2)*(M2+2));  // Dirichlet
N = spzeros((M1+2)*(M2+2),(M1+2)*(M2+2));  // Neumann

Fext=zeros((M1+2)*(M2+2),1);

i=M1+2;
for j=2:M2+1      // x = L1
  k=i+(j-1)*(M1+2);

  N(k,k)        =  1.5;
```

```
  N(k,k-1)   = -2.0;
  N(k,k-2) =  0.5;
end

i=1;              // x = 0
for j=1:M2+2
  k=i+(j-1)*(M1+2);

  D(k,k) = 1.0;
end

j=M2+2;            // y = L2
for i=2:M1+1
  k=i+(j-1)*(M1+2);

  N(k,k)        =  1.5;
  N(k,k-M1-2)   = -2.0;
  N(k,k-2*M1-4) =  0.5;
end

j=1;            // y = 0
for i=2:M1+2
  k=i+(j-1)*(M1+2);

  D(k,k) = 1.0;
end

Bext = D + N;

// MATRICES OF THE BS OPERATOR
Idint=speye((M1+2)*(M2+2),(M1+2)*(M2+2)) - spones(diag(spones(diag(Bext))));

Bint=diag(sparse(wLx))*Lx + diag(sparse(wLy))*Ly ...
   + diag(sparse(wMxy))*Mxy + diag(sparse(wGx))*Gx ...
   + diag(sparse(wGy))*Gy ...
    - r*Idint;

////////////////////////////
//    TIME ITERATIONS
////////////////////////////
N=100
Deltat=T/N;
theta=0.5;
imxtmp=100;

// Matrix to be inversed
A = Idint - theta*Deltat*Bint + Bext;

[hh,rk]=lufact(A);

// Initial data
U0=matrix(h(x'*ones(1,M2+2),ones(M1+2,1)*y),(M1+2)*(M2+2),1);

itertmp=1;
U=U0;
```

```
while itertmp<=imxtmp
  Fint=Idint*U + (1-theta)*Deltat*Bint*U;

  Fext(1+(0:M2+1)*(M1+2)) = K*exp(-r*Deltat*itertmp); // Non homogeneous Dirichlet at x=0
  Fext(1:M1+2)            = K*exp(-r*Deltat*itertmp); // Non homogeneous Dirichlet at y=0

  U=lusolve(hh,Fint+Fext);

  itertmp=itertmp+1;
end

// Plotting of the solution
m=228;
n= fix(3/8*m);
rr=[(1:n)'/n; ones(m-n,1)];
g=[zeros(n,1); (1:n)'/n; ones(m-2*n,1)];
b=[zeros(2*n,1); (1:m-2*n)'/(m-2*n)];
hhh=[rr g b];
xset("colormap",hhh)

v=matrix(U,M1+2,M2+2);
xbasc(); plot3d1(x,y,v);
```

Appendix C : Scilab scripts for FEM methods (dimension 1)

Script for the elementary matrices computations (case of the cubic splines functions)

```
clear;

a=0;
b=20;
M=199;     // Spatial discretization

nbbase=4

function [y]=base(i,x)
   select i,
     case 1 then y=x.^3/6.0;
     case 2 then y=-x.^3/2.0 + 0.5*x.^2 + 0.5*x + 1.0/6.
     case 3 then y=0.5*x.^3 - x.^2 + 2.0/3.
     case 4 then y=-x.^3/6.0 + 0.5*x.^2 - 0.5*x + 1.0/6.
   end
endfunction

function [y]=dbase(i,x)
   select i,
     case 1 then y=0.5*x.^2
     case 2 then y=-1.5*x.^2 + x + 0.5
     case 3 then y=1.5*x.^2 - 2*x
     case 4 then y=-0.5*x.^2 + x - 0.5
   end
endfunction

////////////////////////////////////////////////////////////////////////////
// Integration points of Gauss-Legendre on (0,1)
// n=5 --> compute exactly integrals of polynomial whose degree <= 2*n-1 = 9
////////////////////////////////////////////////////////////////////////////
glx=[0.5-1./42*sqrt(245+14*sqrt(70)),...
     0.5-1./42*sqrt(245-14*sqrt(70)),...
     0.5,...
     0.5+1./42*sqrt(245-14*sqrt(70)),...
     0.5+1./42*sqrt(245+14*sqrt(70))];
```

```
glp=[161/900-13/1800*sqrt(70),...
     161/900+13/1800*sqrt(70),...
     64./225,...
     161/900+13/1800*sqrt(70),...
     161/900-13/1800*sqrt(70)];

x=linspace(a,b,M+2);

A=spzeros(M+2,M+2);   // Mass matrix
L=spzeros(M+2,M+2);   // BS Laplacian matrix
G=spzeros(M+2,M+2);   // BS Gradient matrix

// Assemblage of A, L and G
for k=1:M+1     // M+1 elements in the mesh !

    ja = x(k+1)-x(k);

    for i=1:nbbase

      is = k - i + 3;

      if (is<=M+2) & (is>=1) then

         for j=1:nbbase

            js = k - j + 3;

         if (js>=1) & (js<=M+2) then
            A(is,js) = A(is,js) + ja*glp*( base(i,glx).*base(j,glx) )';
            L(is,js) = L(is,js) +    glp*(((x(k)*(1-glx)+x(k+1)*glx).^2).*dbase(i,glx).*dbase(j,glx))'
            G(is,js) = G(is,js) +    glp*((x(k)*(1-glx)+x(k+1)*glx).*base(i,glx).*dbase(j,glx))';
            end
         end
      end
   end
end

save('BS1d_assemb_s3_'+string(a)+'-'+string(b)+'_M'+string(M)+'.save',a,b,M,x,glx,glp,A,L,G);
```

Script for the option valuation using FEM

```
////////////////////////////////////////////////
//       CIMPA HANOI 23/04/2007 - 4/05/2007
//   Finite Element Method for American options
////////////////////////////////////////////////
clear

//////////////////////////////////////
//   SPATIAL DISCRETIZATION
//////////////////////////////////////
M = 199;               // Number of element - 1
load('BS1d_assemb_s3_0-20_M'+string(M)+'.save')
nbbase = 4;
```

```
//ymin=-1;
//ymax=3;

////////////////////////////////////////
//       OPTION PARAMETER
////////////////////////////////////////
r=0.1;
sig=0.2;
Maturity=1.0;
strike=10;
deff('[payoff]=h(x)',['payoff=max(strike-x,0)']);

///////////////////////////////////
//     TIME DISCRETIZATION
///////////////////////////////////
T=Maturity;
N=100;
Dt=T/N;
imxtmp=100;

// Algorithme gnral
xx=linspace(a,b,M+2);
tt=linspace(0,imxtmp*Dt,imxtmp+1);

//////////////////
// INITIAL DATA
//////////////////
rough = %T;

select rough
case %T,          // Rough intial data
   u0 = h(xx)';
case %F,          // Regular initial data
   u0 = BSPut_v(x,strike,r,sig*ones(x),Dt)';
   imxtmp = imxtmp - 1;
end

// Interpolation in the S3 space
Interp = 2./3.*speye(M+2,M+2) + ...
         1./6.*diag(sparse(ones(M+1,1)),1) + ...
         1./6.*diag(sparse(ones(M+1,1)),-1);
u0 = Interp\u0;

// Matrix to be inversed at each time step (European case)
DF = (1 + 0.5*r*Dt)*A + 0.25*Dt*sig^2*L + 0.5*Dt*(sig^2 - r)*G;
// Matrix used in the projected gradient method (American case)
DFsym = (1 + 0.5*r*Dt)*A + 0.25*Dt*sig^2*L;

////////////////////////////////////////////////////
//        TIME ITERATES (Crank-Nicolson scheme)
////////////////////////////////////////////////////
v=u0;
amer=u0;

// Projected Gradient Parameters
```

```
tolam=1e-5;
eigen=spec(full(DFsym));
//rho= 1/max(eigen);
rho=1/1.6121822;
iterammax=100;
iterglob = 1;

for itertmp=1:imxtmp

        // Solve for v = u^{n+1/2} (Dt --> Dt/2)
        u = v;
        F = A*u;

        v = DF\F;
        v = 2*v - u;   // u^{n+1} = 2*u{n+1/2} - u^{n}

        // Projected Gradient Method for the American Price
        iteram = 1;
        Famer  = A*amer - 0.5*Dt*(sig^2 - r)*G*amer;
        amerold = amer;
        err = 1;
        while (iteram<=iterammax & err>tolam),
           amernew = amerold - 2*rho*(DFsym*amerold - Famer);
           amernew = max(amernew,u0);
           err = norm(amernew - amerold);
           errglob(iterglob) = err;
           iteram   = iteram + 1;
           iterglob = iterglob + 1;
           amerold = amernew;
        end

        amer = 2*amernew - amer;

        xbasc();
        //plot2d(xx,v,leg="Iter="+string(itertmp*Dt),rect=[a,ymin,b,ymax])
        plot2d(xx,[Interp*amer,Interp*v,h(xx)'],leg="Iter="+string(itertmp*Dt));

end
```

Bibliography

[1] P. G. CIARLET, The Finite Element Method for Elliptic Problems, Classics in Applied Mathematics **40**, ed. SIAM, 2002.

[2] F. DUBOIS, T. LELIÈVRE, *Efficient pricing of Asian options by the PDE approach*, to appear in J. Comput. Finance (2007).

[3] C. JOHNSON, The finite element method for elliptic problems, 1986.

[4] G. GOLUB and C. VAN LOAN, *Matrix computations*, John Hopkins, 1989.

[5] W. HIRSCH, Computational methods for fluids dynamics, 1978.

[6] L.C.G. ROGERS and Z. SHI, *The value of an Asian option*, J. Appl. Proba. **32**(4), 1077–1088 (1995).

[7] G. STRANG, G.J. FIX, An Analysis of the Finite Element Method, Prentice Hall, 1973.

[8] P. WILMOTT, J. DEWYNNE and S. HOWINSON, Option Pricing, Mathematical Models and computation, Oxford Financial Press, Oxford, 1993.