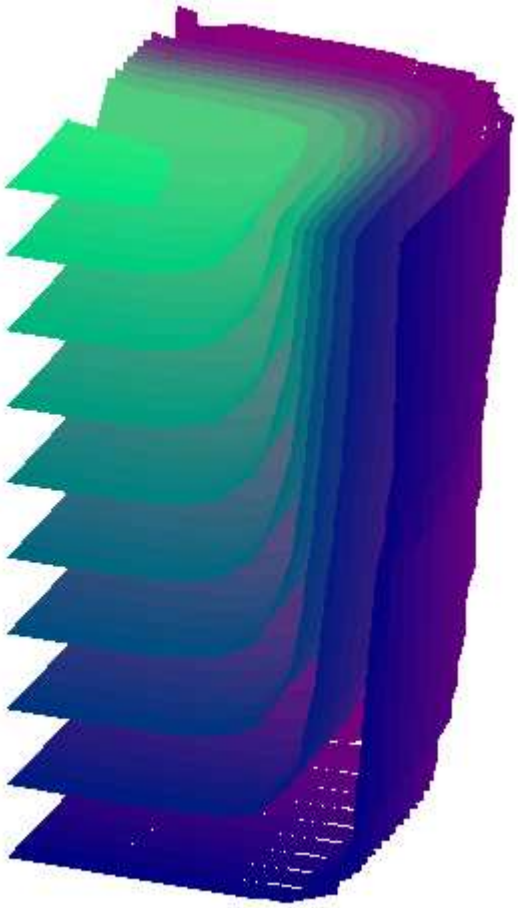
Université de Pierre et Marie CURIE (PARIS VI)
4 Place Jussieu 75005 Paris

Priçing d'option financière
par la méthode des EDP



NGUYEN Chi Thanh
2007-2008

Table des matières

Introduction

Ce projet est le travail à accomplir pour le module Informatique Scientifique du Master 1 à l'Université Pierre et Marie Curie. Le but essentiel est de calculer la solution des Equations aux Dérivées Partielles par les méthodes numériques, de visualiser la solution par des outils graphiques, en changeant les paramètres de l'équation, on peut voir le comportement de la solution par rapport à l'équation. Les équations résolues dans ce projet sont de type Equations de la chaleur. Plusieurs études mathématiques et informatiques ont été étudiés.

Etudes Mathématiques : La démonstration d'existance et d'unicité de la solution d'une EDP, des Méthodes de différences finies, Méthodes des éléments finis, Méthodes matricielles comme LU, Gradient Conjugué, GMRES, Méthode de calculer et présenter des iso-surfaces, Méthode de calculer la dérivée par différences finies et différenciation automatique.

Etudes Informatiques : La programmation C++ pour les calculs scientifiques de haut niveau. La programmation LaTeX, L'initialisation à OpenGL (Open Graphics Library), l'utilisation de FreeFem++, GNUplot, Emacs, Linux.

Le travail : Il s'agit de calculer la solution des deux équations :

$$\partial_t u - \frac{\sigma^2}{2}\partial_{xx} u - \left(r - \frac{\sigma^2}{2}\right)\partial_x u + ru = 0 \tag{1}$$

Le domaine de calcul est : $\Omega \subset R^2$ avec $\Omega =]-L, L[\times]O, T[$ et les conditions aux limites :

$$\begin{aligned} &\text{En } t = 0, \quad u(x,0) = (K - e^x)^+ \\ &\text{En } x = -L, \quad u(-L,t) = Ke^{-rt} \\ &\text{En } x = L, \quad u(L,t) = 0 \end{aligned}$$

$$\frac{\partial u}{\partial t} - \frac{\sigma_1^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{\sigma_2^2}{2}\frac{\partial^2 u}{\partial y^2} - q\sigma_1\sigma_2\frac{\partial^2 u}{\partial x \partial y} + \mu_1\frac{\partial u}{\partial x} + \mu_2\frac{\partial u}{\partial y} + ru = 0 \tag{2}$$

Conditions aux limites :

- Conditions Dirichelet :

$$u(L,y,t) = u(x,L,t) = 0 \qquad (x,y) \in \Gamma_N$$

- Conditions Neumann :

$$\kappa \nabla u.\eta = 0 \ \text{ sur } \ x = -L \ \text{ et } \ y = -L$$

– Conditions initiales :

$$u(x, y, 0) = (K - e^x - e^y)^+ \qquad (x, y) \in \Omega$$

Les maillages sont d'abord générés avec FreeFem++, puis stoqués dans un fichier numérique. Les données de l'équation sont aussi stoké dans un fichier numérique. Le programme en C++ lit ces fichiers, effectue des calculs afin de sortir une solution sous forme un fichier numérique aussi. La solution peut être visualiser avec l'outil GNUplot. D'un autre côté, l'équation peut être résolue par FreeFem++, puis comparée avec la solution calculée en C++. A la fin, on étend le problème en dimension 3, plus le parametre de fonction qui fait que la solution est de dimension 4. GNUplot ne permet plus à visualiser la solution et fait donc appel à OpenGL, qui utilise la couleur comme le 4ème paramètre.

Une fois la solution est calculée, quelques tâches supplémentaires sont demandés : Trouver la valeur locale de la solution (sur un point donné), calculer la dérivée de la solution par la méthode de différences finies, et par la méthode de différenciation automatique, puis comparer ces deux résultats.

Le rapport doit être ridigé sous LaTeX. Tout le projet est travailé sous Linux, ainsi que ses outils.

Chapitre 1

Projet 1 : Equation de la chaleur

1.1 Le Problème Mathématique

Equation à résoudre :

$$\partial_t u - \frac{\sigma^2}{2}\partial_{xx} u - \left(r - \frac{\sigma^2}{2}\right)\partial_x u + ru = 0 \tag{1.1}$$

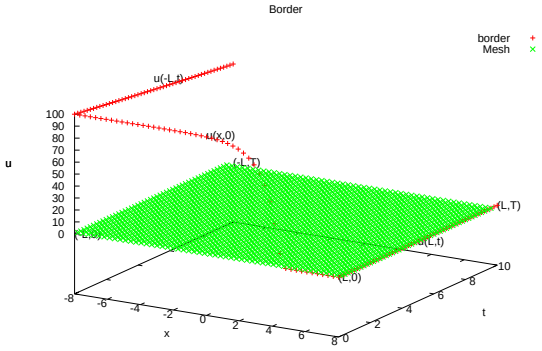


FIG. 1.1 – Border

Conditions aux limites : $\Omega \in R^2$ avec $\Omega =]-L, L[\times]O, T[$

$$\begin{aligned} &\text{En } t = 0, \quad u(x,0) = (K - e^x)^+ \\ &\text{En } x = -L, \quad u(-L,t) = Ke^{-rt} \\ &\text{En } x = L, \quad u(L,t) = 0 \end{aligned} \tag{1.2}$$

Les données Il y a 7 constantes données

- Les réels K, L, T, σ, r
- Les entiers positif M, N qui sont respectivement le nombre de noeuds en temps t et en x

1.2 Résolution numérique

1.2.1 Principes généraux

Pour la résolution numériquement, il faut en premier discrétiser le domaine de

définition. Etant donner N et M qui sont respectivement le nombre des pas de calcul en x et en t, on a de différentes manières de discrétisation. Tout dépend de la méthode qui s'appliquera. Nous allons aborder deux maillages différents (qui seront détaillés plus tard) .Ensuite, bien entendu, le but du projet est de trouver la solution de (1.1). Supposont que la discrétisation a été faite, u à trouver sera concrètement un tableau de dimension $(N+1)\times(M+1)$. Indexons alors les valeurs à calculer u_i^j (i et j indexent respectivement x et t). Avec les conditions aux limites, nous avons aupparavant $u_0^{(j)}$, $u_N^{(j)}$, $u_{(i)}^0$ qui sont les conditions aux limites en $x=-L$, $x=L$ et $t=0$. Grâce aux différentes formules d'approximation variationnelles (les schémas numériques), on se ramènes à la résolution d'un système linéaire de la forme :

$$A.X = B.b^1 \tag{1.3}$$

La résolution du système linéaire possède plusieurs méthodes. Nous allons en faire 2 méthodes essentielles et basiques : Méthode directe LU et Méthode itératif Gradient Conjugué, qui seront détaillées plus loin.

1.3 Les schémas

1.3.1 Schéma d'Euler implicite

$$\frac{u_i^j - u_i^{j-1}}{\partial t} - \frac{\sigma^2}{2\partial x^2}\left[u_{i+1}^j - 2u_i^j + u_{i-1}^j\right] - \left(\frac{r}{2\partial x} - \frac{\sigma^2}{4\partial x}\right)\left[u_{i+1}^j - u_{i-1}^j\right] + ru_i^j = 0$$

Par quelques simples calculs :

$$u_{i-1}^j\left[-\frac{\sigma^2}{2\partial x^2} + \frac{r}{2\partial x} - \frac{\sigma^2}{4\partial x}\right] + u_i^j\left[\frac{1}{\partial t} + \frac{\sigma^2}{\partial x^2} + r\right] + u_{i+1}^j\left[-\frac{\sigma^2}{2\partial x^2} - \frac{r}{2\partial x} + \frac{\sigma^2}{4\partial x}\right] = \frac{u_i^{j-1}}{\partial t}$$

On est amené donc à la résolution du système linéaire :

$$\begin{bmatrix} 1 & & & & & & \\ a & b & c & & & & \\ & a & b & c & & & \\ & & . & . & . & & \\ & & & a & b & c & \\ & & & & a & b & c \\ & & & & & & 1 \end{bmatrix}\begin{bmatrix} u_0^j \\ u_1^j \\ . \\ . \\ . \\ u_{N-1}^j \\ u_N^j \end{bmatrix} = \frac{1}{\partial t}\begin{bmatrix} u_0^j \\ u_1^{j-1} \\ . \\ . \\ . \\ u_{N-1}^{j-1} \\ u_N^j \end{bmatrix} \qquad \begin{aligned} a &= \left(-\tfrac{\sigma^2}{2\partial x^2} + \tfrac{r}{2\partial x} - \tfrac{\sigma^2}{4\partial x}\right) \\ \text{avec, } b &= \left(\tfrac{1}{\partial t} + \tfrac{\sigma^2}{\partial x^2} + r\right) \\ c &= \left(-\tfrac{\sigma^2}{2\partial x^2} - \tfrac{r}{2\partial x} + \tfrac{\sigma^2}{4\partial x}\right) \end{aligned} \tag{1.4}$$

Qui est bien de la forme (1.3) : $A.X = B.b$ avec A une matrice tridiagonale, $B = \frac{1}{\partial t}.I_n$,
La résolution de ce système est décrite dans le paragraphe 4

[1] pour des raisons de diversités de plusieurs questions en un seul programme, nous insérons la matrice B dans tous les cas. Même pour le cas non-necessaire, on met $B = I_n$

[2] Les termes aux "bout" du vecteur u^{j-1} doivent s'appliquer pour les conditions aux limites

1.3.2 Schéma de Crank-Nicolson

Rappelons la formule de variationnelle :

$$\partial_t u - \frac{\sigma^2}{2}\partial_{xx} u - \left(r - \frac{\sigma^2}{2}\right)\partial_x u + ru = 0$$

ou bien :

$$\partial_t u = \frac{\sigma^2}{2}\partial_{xx} u + \left(r - \frac{\sigma^2}{2}\right)\partial_x u - ru$$

Par approximation de Crank-Nicolson :

$$\frac{u^j - u^{j-1}}{\partial t} = \frac{1}{2}.\left(\frac{\sigma^2}{2\partial x^2}\left[v_{i+1} - 2v_i + v_{i-1}\right] + \left(\frac{r}{2\partial x} - \frac{\sigma^2}{4\partial x}\right)\left[v_{i+1} - v_{i-1}\right] - rv_i\right)$$

avec $v = u^j + u^{j-1}$. Si on pose encore $w = u^j - u^{j-1}$, on a le système à résoudre :

$$C.w = D.v$$

$$C = \tfrac{1}{\partial t}.I_n \ , \ \ D = \tfrac{1}{2}.\begin{bmatrix} 1 & & & & & & \\ a & b & c & & & & \\ & a & b & c & & & \\ & & . & . & . & & \\ & & & a & b & c & \\ & & & & a & b & c \\ & & & & & & 1 \end{bmatrix}$$

$$\text{avec,} \quad \begin{aligned} a &= \tfrac{\sigma^2}{2\partial x^2} - \tfrac{r}{2\partial x} + \tfrac{\sigma^2}{4\partial x} \\ b &= -\tfrac{\sigma^2}{\partial x^2} - r \\ c &= \tfrac{\sigma^2}{2\partial x^2} + \tfrac{r}{2\partial x} + \tfrac{\sigma^2}{4\partial x} \end{aligned}$$

Ce système est équivalent à $[C - D].u^j = [C + D].u^{j-1}$. En posant $A = [C - D]$, $B = [C + D]$ on revient évidemment au système habituel $A.X = B.b$.

1.4 Les résolutions du système linéaire

En reprenant de ce qui précède, on doit résoudre le système de la forme $A.X = B.b$. Mais comme les A, B, b sont données, par un calcul simple de $B.b$ on revient à résoudre un système de la forme $A.X = B$, avec A une matrice tridiagonale. Il ne faut pas oublier dans notre cas, avant l'appel de la résolution du système linéaire, il faut imposer les valeur "au bout" du vecteur B convenable aux conditions de limite.

1.4.1 Méthodes de décomposition LU

Soit le système à résoudre :

$$A.X = B^3$$

La méthode de décomposition LU fait parti de la famille de méthodes directes. Le principe est décomposer la matrice A en produit d'une matrice supérieure et une matrice inférieure,puis retrouver la solution en deux étapes par des calculs directs. Donnons dans le cas concret, A est

[3] La méthode LU s'applique ssi A est inversible

une matrice tridiagonale :

$$\begin{bmatrix} 1 & & & & & & & \\ a_2 & b_2 & c_2 & & & & & \\ & a_3 & b_3 & c_3 & & & & \\ & & . & . & . & & & \\ & & & a_{n-2} & b_{n-2} & c_{n-2} & & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} & \\ & & & & & & 1 & \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ . \\ . \\ . \\ B_{n-1} \\ B_n \end{bmatrix} \tag{1.5}$$

Par des calculs de récurrences :

$$\begin{cases} b_1^* = b_1 \\ c_1^* = \frac{c_1}{b_1} \end{cases} \qquad \begin{cases} b_j^* = b_j - a_j.c_{j-1}^* \\ c_j^* = \frac{c_j}{b_j^*} \end{cases}, j = 1, 2...n., \text{ en mettant} \qquad \begin{cases} b_1 = b_n = 1 \\ a_1 = a_n = c_1 = c_n = 0 \end{cases}$$

Le système devient $L.U.X = B$:

$$\begin{bmatrix} 1 & & & & & & \\ a_2 & b_2^* & & & & & \\ & a_3 & b_3^* & & & & \\ & & . & . & & & \\ & & & a_{n-2} & b_{n-2}^* & & \\ & & & & a_{n-1} & b_{n-1}^* & \\ & & & & & & 1 \end{bmatrix} . \begin{bmatrix} 1 & & & & & & \\ & 1 & c_2^* & & & & \\ & & 1 & c_3^* & & & \\ & & & . & . & & \\ & & & & 1 & c_{n-2}^* & \\ & & & & & 1 & c_{n-1}^* \\ & & & & & & 1 \end{bmatrix} . \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ . \\ . \\ . \\ B_{n-1} \\ B_n \end{bmatrix} \tag{1.6}$$

On pose $U.X = Y$, le système se résoud ensuite en deux étapes :

$$L.Y = B, \text{ se résoud par} \begin{cases} Y_1 = \frac{B_1}{b_1^*} \\ Y_j = \frac{B_j - a_j.Y_{j-1}}{b_j^*} \end{cases} ,j = 2...n$$

$$U.X = Y, \text{ se résoud par} \begin{cases} X_n = Y_n \\ X_j = Y_j - c_j^*.X_{j+1} \end{cases} ,j = (n-1)...1$$

1.4.2 Méthodes du Gradient Conjugué

Soit le système à résoudre :

$$A.X = B^4$$

La méthode du Gradient Conjugué fait parti de la famille de méthodes itératives. Le principe est de donner une valeur initiale x_0, construire la valeur suivante x_1...et par récurrence, construire une suite x_k. Et par la théorie mathématique, on démontre que la suite $x_{(k)}$ converge vers la solution cherchée lorsque k tend vers l'infini. La partie théorie mathématique se trouve dans tous les libres de d'analyse numérique matricielle. On expose ici seulement la partie algorithme qui permet de programmer la résolution du système.

[4] La méthode du Gradient Conjugué s'applique ssi A est symétrique définie positive

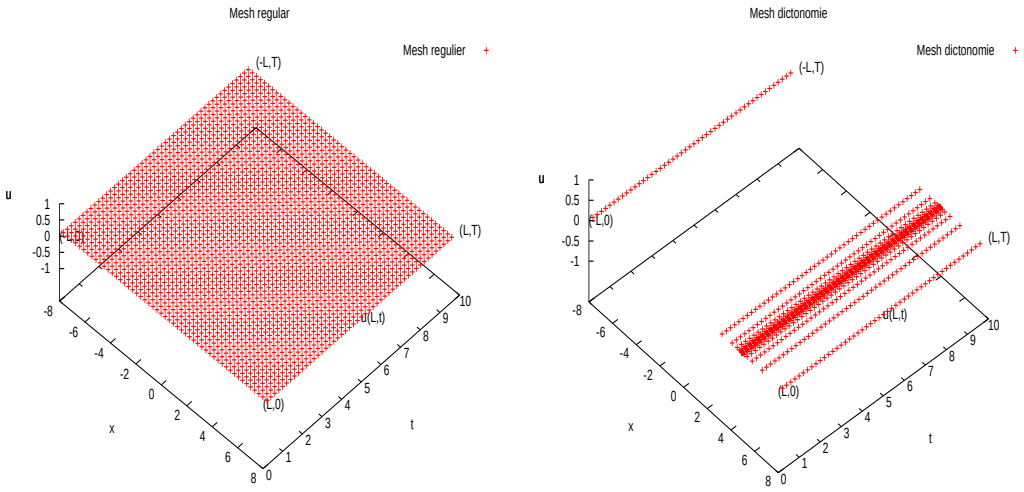
Soient les valeur données : A, B, x_0,eps :

$$
\begin{array}{l}
g^0 = A.x^0 - B \\
h^0 = -g^0 \\
\text{Pour } k = 0 \text{ à n} \\
\rho = -\frac{(g^k, h^k)}{(h^k, Ah^k)} \\
\| \; x^{k+1} = x^k + \rho h^k \\
g^{k+1} = g^k + \rho A h^k \\
\gamma = \frac{(g^{k+1}, g^{k+1})}{(g^k, g^k)} \\
h^{k+1} = -g^{k+1} + \gamma h^k \\
\text{si } (g^{k+1}, g^{k+1}) \leq \epsilon \text{ stop}
\end{array}
$$

1.5 Maillage

Nous n'abordons ici que des maillages rectangulaires, ce qui fait que les paramètres x et t seront distribués indépendemment. Pour la première étape, on distribue les points régulièrement (FiG-gauche). Ensuite, pour une demande de précision autour d'un point ($x = Log(K)$) , on distribue les points par méthode dite dictonomie (FiG-droite).

Les Maillages



1.6 Les résultats en graphique

Par de différentes compositions de schémas et de méthodes de résolution linéaire, on a de différents résultats :

Schéma Euler utilisant méthode LU

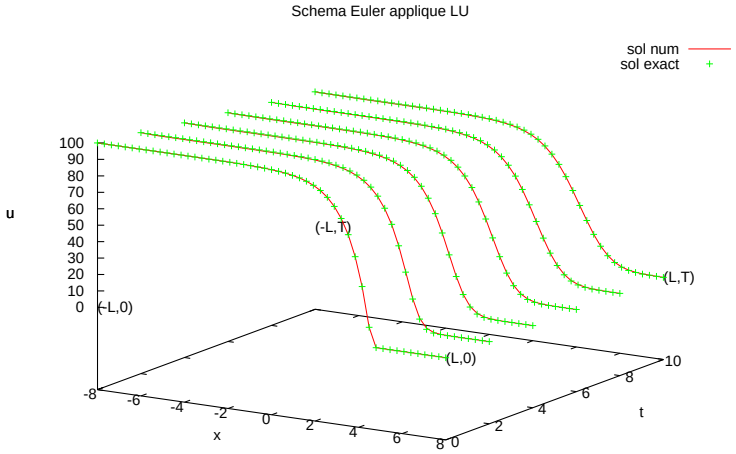


Schéma Euler utilisant méthode GC

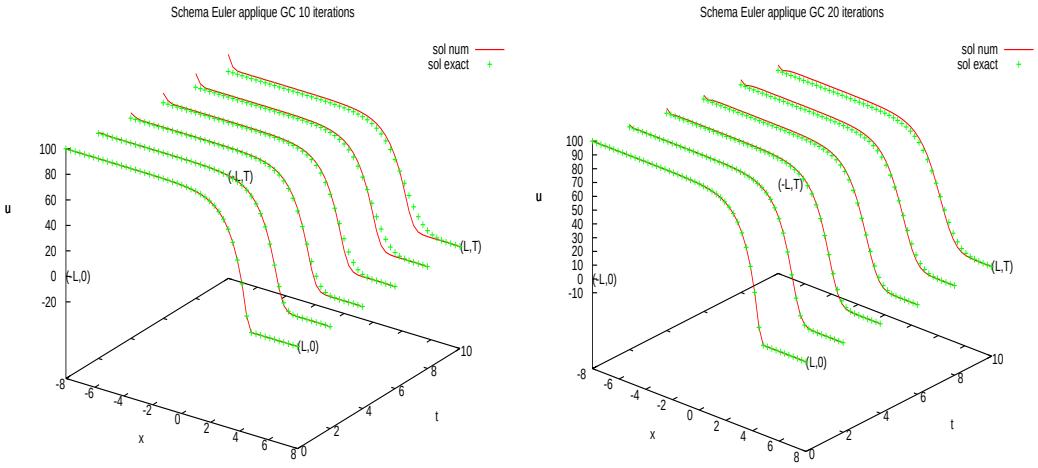


Schéma Crank-Nicolson utilisant LU

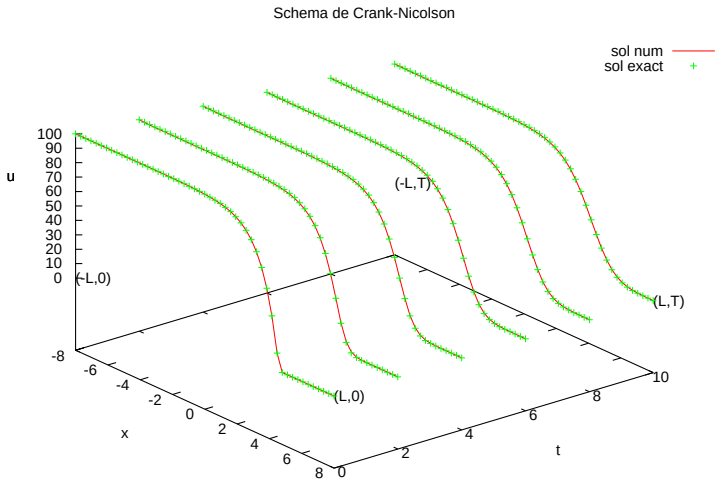
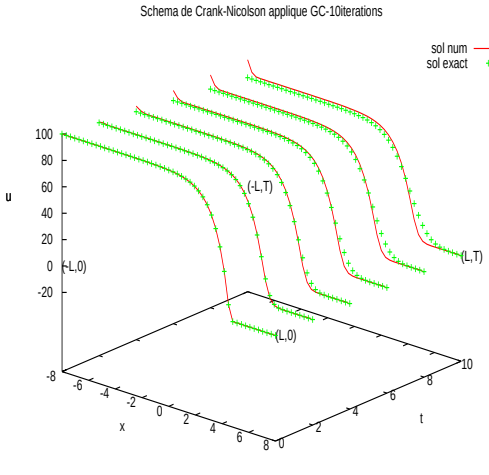
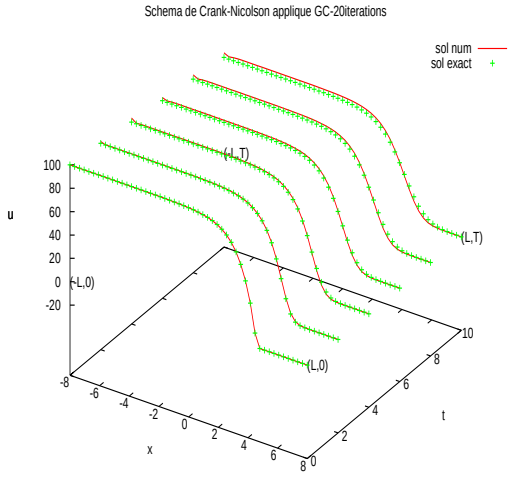


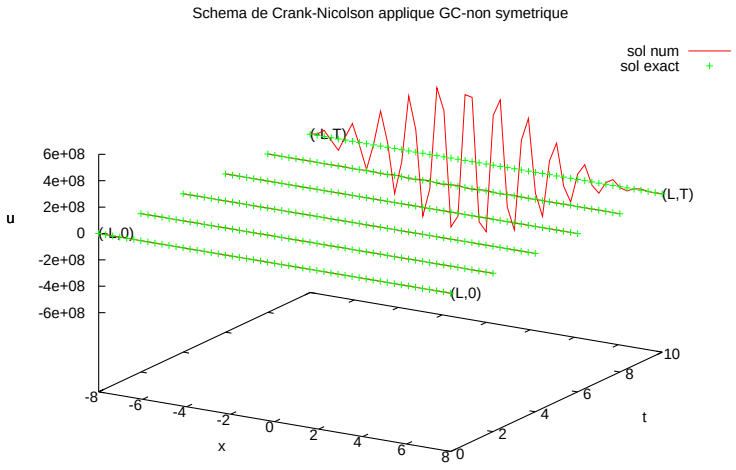
Schéma Crank-Nicolson utilisant GC





Le Schéma de Crank-Nicolson ici est théoriquement non symétrique. Mais dans ce cas, la dissymétrie de la matrice est assez faible pour qu'il y ait lieu de la convergence de Gradient Conjugué

Gradient Conjugué divergente si on change $r = 2$, on aura la dissymétrie forte et on ne peut plus appliquer la méthode Gradient Conjugué dans ce cas.


Schema de Crank-Nicolson applique GC-non symetrique
sol num
sol exact
(-L,T)
(L,T)
(-L,0)
(L,0)
u
6e+08
4e+08
2e+08
0
-2e+08
-4e+08
-6e+08
-8
-6
-4
-2
0
2
4
6
8
x
0
2
4
6
8
10
t

Chapitre 2

Projet 2 : Pricing d'option panier sur 2 sous-jacents

2.1 Présentation du problème

Modèle Black-Scholes : Le modèle Black-Schole est, à l'origine, un modèle à deux actifs : l'un risqué, l'autre pas. Typiquement, l'actif risqué est une action (l'action sous-jacente à l'option) tandis que l'actif non risqué s'apparente à une obligation. A l'instant t, le prix de l'obligation est R_t et le prix de l'action est S_t. L'évolution de l'obligation est relativement simple puisque l'on suppose que :

$$dR_t = r_t R_t dt \qquad \text{soit} \qquad R_t = R_0 e^{\int_0^t r_s ds}$$

où $r_t \geq 0$ représente le taux d'intérêt instantanné. Nous supposerons toujours que $R_0 = 1$.

Le prix de l'action, $S_{t\,t\geq 0}$ est régi par l'équation différentielle stochastique (EDS)

$$dS_t = S_t(\mu_t dt + \sigma_t dt), \qquad S_0 > 0 \text{ donné},$$

où μ_t est un paramètre réel, et $\sigma_t \geq 0$, le paramètre σ s'appelle la volatilité.

Pricing d'option panier sur 2 sous-jacents : On prend le modèle suivant

$$\dot{S}^1_t = S^1_t(r\dot{t} + \dot{W}^1_t), \quad \dot{S}^2_t = S^2_t(r\dot{t} + \dot{W}^2_t) \tag{2.1}$$

où W^1_t, W^2_t sont des browniens tels que

$$\mathcal{E}(\dot{W}^i\dot{W}^j) = \dot{t}\sigma_i\sigma_j\rho_{ij} \text{ avec } = \rho_{11} = \rho_{22} = 1, \quad \rho_{12} = \rho_{21} = q$$

Une option put construite sur la somme $S^1 + S^2$ pour un strike K et une maturité T aura donc comme prix l'espérance du bénéfice escompté, soit

$$P_t = e^{-r(T-t)}\mathcal{E}(K - S^1_T - S^2_T)^+$$

Equation aux dérivées partielles : Dès qu'une quantité déterministe dépend de la solution d'une équation différentielle stochastique, le calcul de Itô s'applique. Ici, il donne (pour une loi de probabilité adaptée) $P_t = u(S^1_t, S^2_t, T-t)$ où u est la solution de

$$\frac{\partial u}{\partial t} - \frac{\sigma_1^2 S_1^2}{2}\frac{\partial^2 u}{\partial S_1^2} - \frac{\sigma_2^2 S_2^2}{2}\frac{\partial^2 u}{\partial S_2^2} - q\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 u}{\partial S_1 \partial S_2} - rS_1\frac{\partial u}{\partial S_1} - rS_2\frac{\partial u}{\partial S_2} + ru = 0$$

$$u(S_1, S_2, 0) = (K - S_1 - S_2)^+ \qquad (S_1, S_2) \in \Re_+^2$$

On change de variable en posant $x = \log S_1,\ y = \log S_2$. Il est facile de voir que l'équation devient

$$\frac{\partial u}{\partial t} - \frac{\sigma_1^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{\sigma_2^2}{2}\frac{\partial^2 u}{\partial y^2} - q\sigma_1\sigma_2\frac{\partial^2 u}{\partial x \partial y} + \mu_1\frac{\partial u}{\partial x} + \mu_2\frac{\partial u}{\partial y} + ru = 0$$

avec :

$$u(x, y, 0) = (K - e^x - e^y)^+ \qquad \text{pour tout} (x, y) \in R^2$$

.

$$\mu_1 = \frac{\sigma_1^2}{2} - r \qquad \text{et} \qquad \mu_2 = \frac{\sigma_2^2}{2} - r$$

2.2 Modélisation mathématique

Conditions aux limites :

$$u(S_1, S_2, 0) = (K - S_1 - S_2)^+ \qquad (S_1, S_2) \in \Re_+^2$$

Equation à résoudre :

$$\frac{\partial u}{\partial t} - \frac{\sigma_1^2 S_1^2}{2}\frac{\partial^2 u}{\partial S_1^2} - \frac{\sigma_2^2 S_2^2}{2}\frac{\partial^2 u}{\partial S_2^2} - q\sigma_1\sigma_2 S_1 S_2\frac{\partial^2 u}{\partial S_1 \partial S_2} - rS_1\frac{\partial u}{\partial S_1} - rS_2\frac{\partial u}{\partial S_2} + ru = 0$$

On effectue un changement de variables en posant $x = logS_1, y = logS_2$. il est facile de voir que l'équation devient :

$$\frac{\partial u}{\partial t} - \frac{\sigma_1^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{\sigma_2^2}{2}\frac{\partial^2 u}{\partial y^2} - q\sigma_1\sigma_2\frac{\partial^2 u}{\partial x \partial y} + \mu_1\frac{\partial u}{\partial x} + \mu_2\frac{\partial u}{\partial y} + ru = 0$$

avec :

$$u(x, y, 0) = (K - e^x - e^y)^+ \qquad \text{pour tout} (x, y) \in R^2$$

.

$$\mu_1 = \frac{\sigma_1^2}{2} - r \qquad \text{et} \qquad \mu_2 = \frac{\sigma_2^2}{2} - r$$

2.3 Résolution EDP

Problème aux limites

$$\frac{\partial u}{\partial t} - \frac{\sigma_1^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{\sigma_2^2}{2}\frac{\partial^2 u}{\partial y^2} - q\sigma_1\sigma_2\frac{\partial^2 u}{\partial x \partial y} + \mu_1\frac{\partial u}{\partial x} + \mu_2\frac{\partial u}{\partial y} + ru = 0 \tag{2.2}$$

Conditions aux limites :

- Conditions Dirichelet :

$$u(L, y, t) = u(x, L, t) = 0 \qquad (x, y) \in \Gamma_N$$

- Conditions Neumann :

$$\kappa\nabla u.\eta = 0 \ \text{ sur } \ x = -L \ \text{ et } \ y = -L$$

- Conditions initiales :

$$u(x, y, 0) = (K - e^x - e^y)^+ \qquad (x, y) \in \Omega$$

Formulation variationelle

Soit l'espace variationel :

$$V_0 = \{v \in H^1(\Omega) : v(L, y, t) = v(x, L, t) = 0\}$$

Trouver $u \in V_0$ telle que pour tout $w \in V_0$, on a :

$$\int_\Omega [\frac{\partial u}{\partial t} w + \frac{\sigma_1^2}{2} \frac{\partial u}{\partial x} \frac{\partial w}{\partial x} + \frac{\sigma_2^2}{2} \frac{\partial u}{\partial y} \frac{\partial w}{\partial y} + \frac{q}{2} \sigma_1 \sigma_2 (\frac{\partial u}{\partial x} \frac{\partial w}{\partial y} + \frac{\partial w}{\partial x} \frac{\partial u}{\partial y}) + w(\mu_1 \frac{\partial u}{\partial x} + \mu_2 \frac{\partial u}{\partial y} + ru)] = 0 \quad (2.3)$$

2.3.1 L'équivalence de deux problèmes

Soit u une solution suffisamment régulière du problème aux limites, u appartient à $H^2(\Omega)$ par exemple. Alors on peut appliquer la formule de Green pour toute fonction w de $H^1(\Omega)$, en particulier de V_0. De sorte que si on multiplie (2.2) par w de V_0, puis intégrer :

$$\int_\Omega [\frac{\partial u}{\partial t} - \text{div}(\kappa \nabla u) + \mu \cdot \nabla u + ru].w.d\Omega = 0$$

$$\int_\Omega [\frac{\partial u}{\partial t} + \mu \cdot \nabla u + ru].w.d\Omega + \int_\Omega (\kappa \nabla u).\nabla w.d\Omega - \int_{\partial \Gamma_N} (\kappa \nabla u).w.\eta.d\Gamma_N = 0$$

en tenant compte que $\kappa \nabla u.\eta = 0$ sur le borne de Neumann, on retrouve :

$$\int_\Omega [\frac{\partial u}{\partial t} + \mu \cdot \nabla u + ru].w.d\Omega + \int_\Omega (\kappa \nabla u).\nabla w.d\Omega = 0$$

Soit maintenant u une solution du problème variationel. Alors pour toute fonctions w de V_0, en particulier pour toute fonction de test w de $D(\Omega)$, on aura :

$$\int_\Omega [\frac{\partial u}{\partial t} w + \frac{\sigma_1^2}{2} \frac{\partial u}{\partial x} \frac{\partial w}{\partial x} + \frac{\sigma_2^2}{2} \frac{\partial u}{\partial y} \frac{\partial w}{\partial y} + \frac{q}{2} \sigma_1 \sigma_2 (\frac{\partial u}{\partial x} \frac{\partial w}{\partial y} + \frac{\partial w}{\partial x} \frac{\partial u}{\partial y}) + w(\mu_1 \frac{\partial u}{\partial x} + \mu_2 \frac{\partial u}{\partial y} + ru)] = 0$$

On applique la formule de Green , on retrouve :

$$\int_\Omega [\frac{\partial u}{\partial t} - \text{div}(\kappa \nabla u) + \mu \cdot \nabla u + ru].w.d\Omega = 0$$

pour toute w de $D(\Omega)$. Ce qui veut dire que u satisfait (2.2) au sens de distribution. Si on suppose en plus que $u \in H^2(\Omega)$ alors on peut appliquer la formule de Green pour toute fonction w de $H^1(\Omega)$, en particulier de V_0, ce qui donne :

$$\int_\Omega [\frac{\partial u}{\partial t} - \text{div}(\kappa \nabla u) + \mu \cdot \nabla u + ru].w.d\Omega - \int_{\partial \Gamma_N} (\kappa \nabla u).w.\eta.d\Gamma_N = 0$$

Par densité des traces dans l'espace $L^2(\partial\Omega)$, on en déduit que $u = 0$ sur de borne de Neumann $L^2(\partial\Omega)$-p.p.

Par limite du programme, l'étude de l'existence et l'unicité de la solution du problème variationel, comme la convergence du problème variationel discret ne seront pas étudiés ici. On suppose que tout ceci a été démontrés, et on ne fait exposer que des méthodes permettant de calculer la solution dans un espace discret donné dans la suite.

2.3.2 Discrétisation du problème variationel

Soit le domaine Ω_h possédant une triangulation T_h. L'espace discret défini par :

$$V_h = \{v \in C^0(\Omega_h) \ \ , v_{|T_k} \in P^1(T_k), \ \ v(L,y,t) = v(x,L,t) = 0\}$$

Le problème variationel discret se pose :
Trouver $u \in V_h$ telle que pour toute $w \in V_h$, on a :

$$\int_{\Omega_h} [\frac{\partial u}{\partial t} w + \frac{\sigma_1^2}{2}\frac{\partial u}{\partial x}\frac{\partial w}{\partial x} + \frac{\sigma_2^2}{2}\frac{\partial u}{\partial y}\frac{\partial w}{\partial y} + \frac{q}{2}\sigma_1\sigma_2(\frac{\partial u}{\partial x}\frac{\partial w}{\partial y} + \frac{\partial w}{\partial x}\frac{\partial u}{\partial y}) + w(\mu_1\frac{\partial u}{\partial x} + \mu_2\frac{\partial u}{\partial y} + ru)].d\Omega_h = 0$$

Sachant que dans cet espace fonctionel discret, on possède des fonctions de base w^i, de sorte que pour toute $u \in V_h$ et toute $w \in V_h$, on a l'écriture :

$$u = \sum_{i=1}^{N_s} u_i.w^i \qquad\qquad w = \sum_{j=1}^{N_s} w_j.w^j \qquad\qquad N_s \ \text{ nombre de sommets}$$

En suite, on discrétise par différence fini par rapport au temps, c'est à dire :

$$\frac{\partial u_i^{m+1}}{\partial t} = \frac{u_i^{m+1} - u_i^m}{\partial t}$$

Ce qui donne le schéma mixte mélanger de différences finies et éléments finis :

$$\begin{aligned}\sum_{i=1}^{N_s}\int_\Omega [\frac{u_i^{m+1}-u_i^m}{\partial t}.w^i w^j &+ u_i^{m+1}\frac{\sigma_1^2}{2}\frac{\partial w^i}{\partial x}\frac{\partial w^j}{\partial x} + u_i^{m+1}\frac{\sigma_2^2}{2}\frac{\partial w^i}{\partial y}\frac{\partial w^j}{\partial y} \\ &+ u_i^{m+1}\frac{q}{2}\sigma_1\sigma_2(\frac{\partial w^i}{\partial x}\frac{\partial w^j}{\partial y} + u_i^{m+1}\frac{\partial w^j}{\partial x}\frac{\partial w^i}{\partial y}) \qquad \forall 1 \le j \le N_s \\ &+ u_i^{m+1}w^j(\mu_1\frac{\partial w^i}{\partial x} + \mu_2\frac{\partial w^i}{\partial y} + rw^i)] = 0\end{aligned}$$

$$\begin{aligned}\sum_{i=1}^{N_s}\int_\Omega [\frac{u_i^{m+1}}{\partial t}.w^i w^j &+ u_i^{m+1}\frac{\sigma_1^2}{2}\frac{\partial w^i}{\partial x}\frac{\partial w^j}{\partial x} + u_i^{m+1}\frac{\sigma_2^2}{2}\frac{\partial w^i}{\partial y}\frac{\partial w^j}{\partial y} \\ &+ u_i^{m+1}\frac{q}{2}\sigma_1\sigma_2(\frac{\partial w^i}{\partial x}\frac{\partial w^j}{\partial y} + u_i^{m+1}\frac{\partial w^j}{\partial x}\frac{\partial w^i}{\partial y}) \qquad \forall 1 \le j \le N_s \\ &+ u_i^{m+1}w^j(\mu_1\frac{\partial w^i}{\partial x} + \mu_2\frac{\partial w^i}{\partial y} + rw^i)] = \int_\Omega \frac{u_i^m}{\partial t}.w^i w^j\end{aligned} \tag{2.4}$$

où l'écriture matricielle :

$$(M + \delta t A)u^{m+1} = Mu^m \quad \text{ avec } M_{ij} = \int_\Omega w^j w^i$$

Pour résoudre ce système linéaire, on se ramène dans une étude locale, i.e dans un triangle. Dans ce cas, les fonctions de base se calculent facilement par les fonctions de coordonnées barycentriques. Dans le cas de l'espace P^1, $w^i = \lambda_i$ avec $i = 1,2,3$, les $\nabla\lambda_i$, $\frac{\partial\lambda_i}{\partial x}$,$\frac{\partial\lambda_i}{\partial y}$ sont toutes constantes et calculables. On possède en plus une formule très générale :

$$\int_{T_k} (\lambda_1)^m(\lambda_2)^n(\lambda_3)^p = \frac{2|T_k|m!n!p!}{(2+m+n+p)!}$$

Partant de (2.4), on retrouve la formule matricielle :

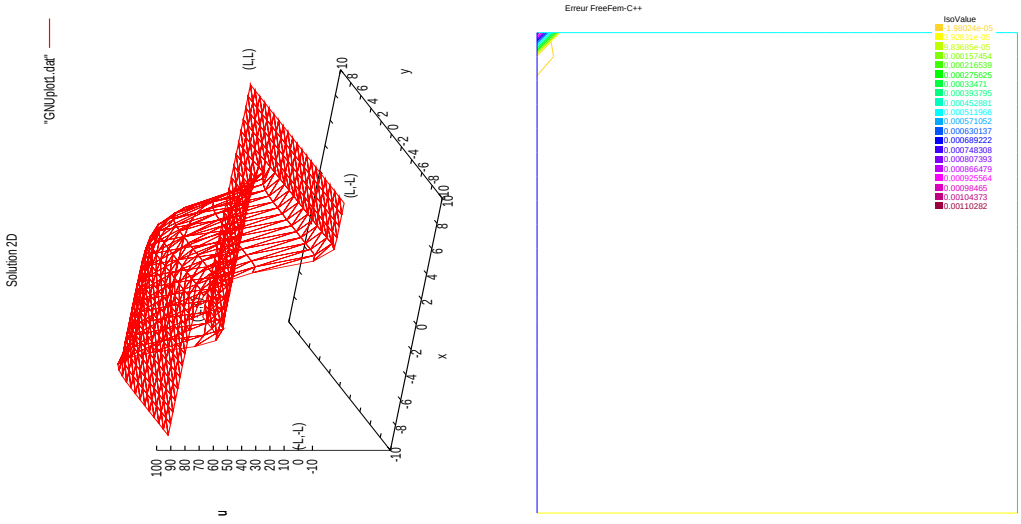
$$(M^k + \delta t A^k).u^{m+1} = M^k.u^m$$

avec

$$\begin{aligned} A_{ij}^k &= \int_{T_k} [\frac{1}{2}\kappa \nabla w^i.\nabla w^i + \mu.\nabla w^j w^i + r w^i w^j].d\Omega \\ M_{ij}^k &= \int_{T_k} w^i w^j d\Omega \\ \kappa &= \begin{pmatrix} \sigma_1^2 & q\sigma_1\sigma_2 \\ q\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \end{aligned} \tag{2.5}$$

2.3.3 Solution calculée

Une fois écrit le système linéaire, on peut résoudre avec plusieurs méthodes numériques matricielles. Ici, on a deux méthodes GMRES et Gradient Conjugué. La méthode Gradient Conjugué est en général appliquable pour des matrices symétriques définies positifs. Mais dans les cas où la dissymétrie de la matrice est faible, on peut essayer de l'utiliser. La solution numérique calculée par C++ est comparée avec celle de FreeFem. L'erreur obtenu est atteint l'odre 10^{-6}. Voici quelques images :



2.4 Etude locale de la solution 2D

Maintenant, on a trouvé la solution numérique sur le maillage donné. Une question à poser est : quelle est la valeur de u sur un point donné. Le calcul s'effectue sur 2 étape :

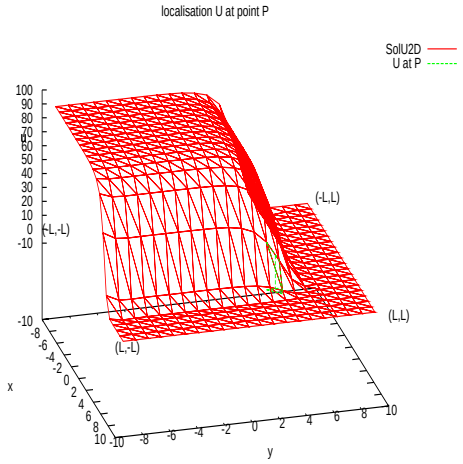
- localisation du point sur le maillage, i.e le point est dans quel triangle
- calcul localement la valeur de u sur ce point.

Première étape : soit un point P donné. On va parcourir tous les triangles du maillage en testant si le point P est dans un triangle. Pour ce test, on a deux méthodes. La première consiste à calculer la surface (toujours positif) des trois triangles formés par le point P et les trois sommets

du triangle T_k du maillage. Si la somme de ces trois surface est exactement 1, le point P est alors dans le triangle T_k. La deuxième méthode consiste à utiliser les fonctions de coordonnées barycentriques. Un point P et un triangle T_k donnés forme trois coordonnées barycentriques λ_i, $i = 1, 2, 3$. Si ces trois coordonnées sont toutes positives, le point P est dans le triangle T_k.

Deuxième étape : Le triangle T_k qui contient le point P a été trouvé. On possède les trois fonctions de coordonnées barycentriques λ_i, $i = 1, 2, 3$, avec les valeurs u_i , $i = 1, 2, 3$ sur les sommets de ce triangle. Alors

$$u(P) = \sum_{i=1}^{3} \lambda_i . u_i$$



2.5 Etude différentielle de la solution 2D

2.5.1 Différence fini

Par la formule de Taylor :

$$u(x + \Delta x) = u(x) + \Delta x.u'(x) + o(\Delta x)$$
$$u(x - \Delta x) = u(x) - \Delta x.u'(x) + o(\Delta x)$$

On peut approximer la dérivée par rapport à K et à q par :

$$\frac{\partial u}{\partial K} \approx \frac{u(K) - u(K - \Delta K)}{\Delta K}$$
$$\frac{\partial u}{\partial q} \approx \frac{u(q) - u(q - \Delta q)}{\Delta q}$$

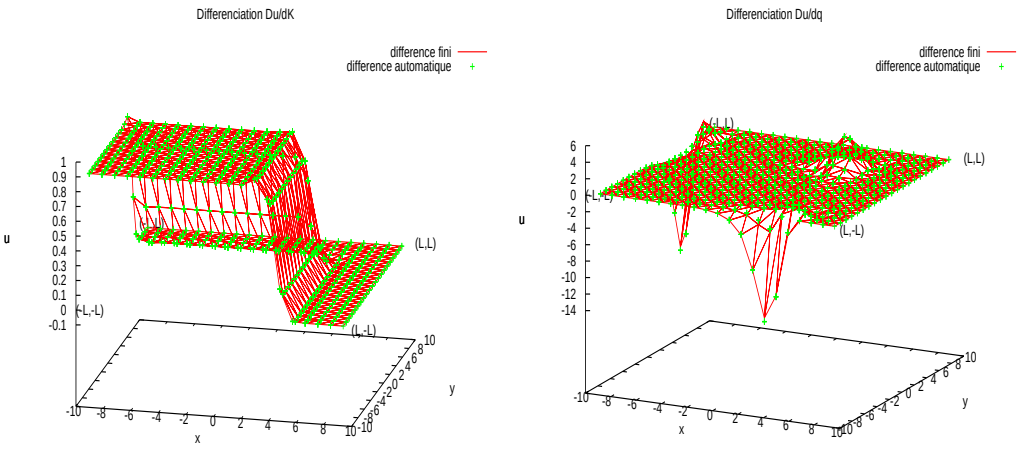
Ou bien par différence centrée :

$$\frac{\partial u}{\partial K} \approx \frac{u(K + \Delta K) - u(K - \Delta K)}{2\Delta K}$$
$$\frac{\partial u}{\partial q} \approx \frac{u(q + \Delta q) - u(q - \Delta q)}{2\Delta q}$$

2.5.2 Différentiation automatique

Comme son nom indique, la Différentiation automatique est une méthode permettant de calculer certaines dérivées numériques d'une fonction automatiquement. Il existe plusieurs bibliothèques pour plusieurs languages pour effectuer ces calculs. On possède ici les bibliothèques ddouble.hpp et R2ddouble.hpp, qui permet de calculer dans les cas réels et R^2 dans le language C++. L'importance est la bonne déclaration des variables concernant les calculs.

graphique



2.6 Extention du problème en dimension 3

2.6.1 Schéma en dimension 3

Quitte à exposer la partie modélisation et variationel, on rentre directement dans le schéma donné :

$$\begin{aligned}\frac{u_i^m - u_i^{m-1}}{\delta t} - \frac{\sigma_1^2}{2}\frac{\partial^2 u_i^m}{\partial x^2} - \frac{\sigma_2^2}{2}\frac{\partial^2 u_i^m}{\partial y^2} - q\sigma_1\sigma_2\frac{\partial^2 u_i^m}{\partial x \partial y} \\ + \mu_1\frac{\partial u_i^m}{\partial x} + \mu_2\frac{\partial u_i^m}{\partial y} + r u_i^m + i\frac{u_i^m - u_{i-1}^m}{(M-m)\delta t} = 0 \\ u_i^0(x,y) = (\frac{i\delta z}{T} - e^x - e^y)^+, \\ u_I^m(x,y) = (\frac{I\delta z}{T} - e^x - e^y)^+ \qquad \forall x, y, i, m\end{aligned}$$

Ce qui donne :

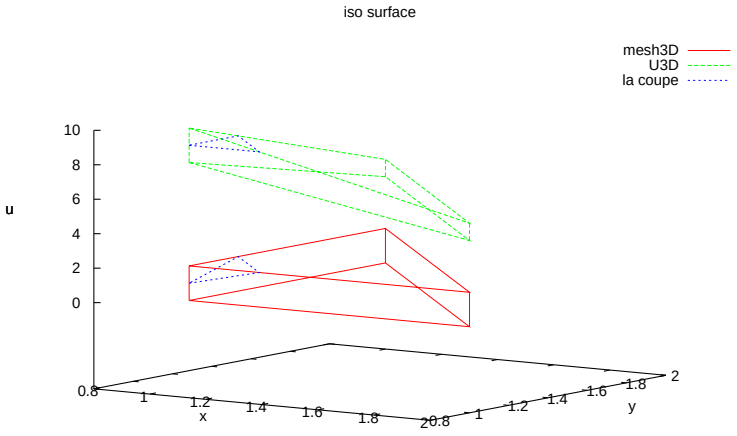
$$\begin{aligned}\frac{1}{\delta t}(u_i^m + i.\frac{u_i^m}{M-m}) - \frac{\sigma_1^2}{2}\frac{\partial^2 u_i^m}{\partial x^2} - \frac{\sigma_2^2}{2}\frac{\partial^2 u_i^m}{\partial y^2} - q\sigma_1\sigma_2\frac{\partial^2 u_i^m}{\partial x \partial y} \\ + \mu_1\frac{\partial u_i^m}{\partial x} + \mu_2\frac{\partial u_i^m}{\partial y} + r u_i^m = i.\frac{u_{i-1}^m}{(M-m)\delta t} + \frac{u_i^{m-1}}{\delta t}\end{aligned}$$

En comparant avec le schéma du cas dimension 2 :

$$\frac{1}{\delta t}(u^m) - \frac{\sigma_1^2}{2}\frac{\partial^2 u^m}{\partial x^2} - \frac{\sigma_2^2}{2}\frac{\partial^2 u^m}{\partial y^2} - q\sigma_1\sigma_2\frac{\partial^2 u^m}{\partial x \partial y} + \mu_1\frac{\partial u^m}{\partial x} + \mu_2\frac{\partial u^m}{\partial y} + ru^m = \frac{u^{m-1}}{\delta t}$$

On peut alors résoudre le cas de dimension 3 en itérant plusieurs résolutions du cas de dimension 2, avec des petits changements des coefficients dans l'écriture matricielle de l'quation linéaire (2.5)

2.6.2 Iso-surface en dimension 3

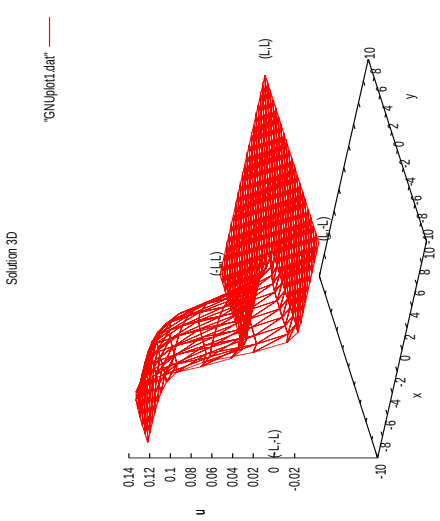


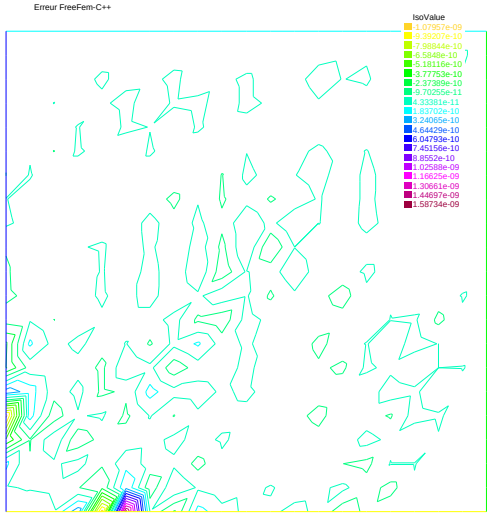
Pour tracer une iso-surface de la solution en 3D, on se ramène dans le cas d'un un parallèlotope simple, construit sur un triangle et deux couches consécutifs de solution et de maillage. Une valeur donnée est considérée comme un plan qui coupe la fonction $u3D$ en certains points. En regardant la proportion des segments formés par ces points dans chaque côté du parallèlotope de u, on peut transformé cette proportion dans le parallèlotope correspondant du maillage. Il reste à associer à chaque iso-surface une couleur spécifique pour avoir enfin une visualisation visible.

2.6.3 Solution trouvée en dimension 3

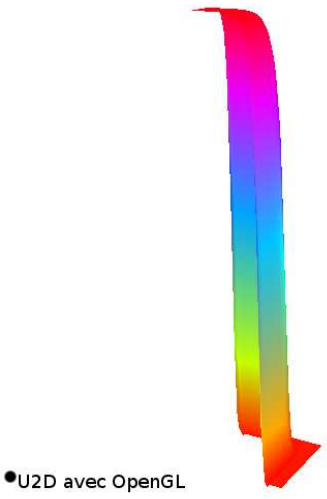
Quelques graphiques résultant des calculs de la solution asisan :

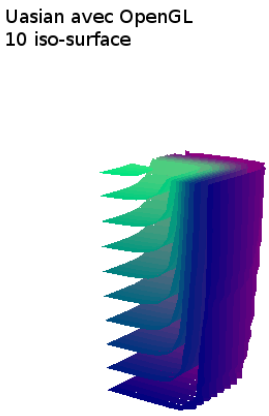
(L,-L)





Les graphiques avec OpenGL :



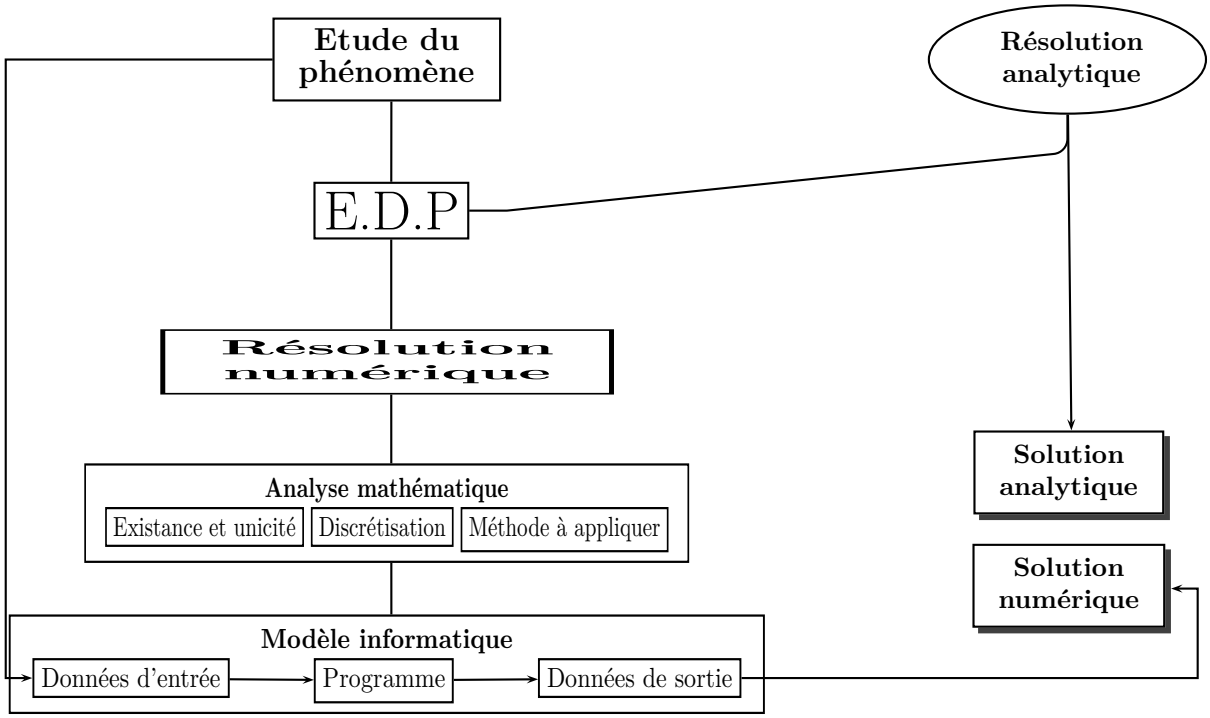


2.7 Annexe

2.7.1 Programmation-Trucs et Astuces

- Création d'un fichier Makefile facilite la compilation du programme.
- Création des script de Gnuplot, qui permet la visualisation imédiate lors de l'exécution du programme, afin de voir le comportement des fonctions.
- Utilisation de l'instruction de C :
  system("commande shell") ;
  Celle-ci peut exécuter le script de Gnuplot, gestion de fichier ....ect.. lors du lancement du programme.

Plan d'étude [1]



[1] Explication du plan d'étude