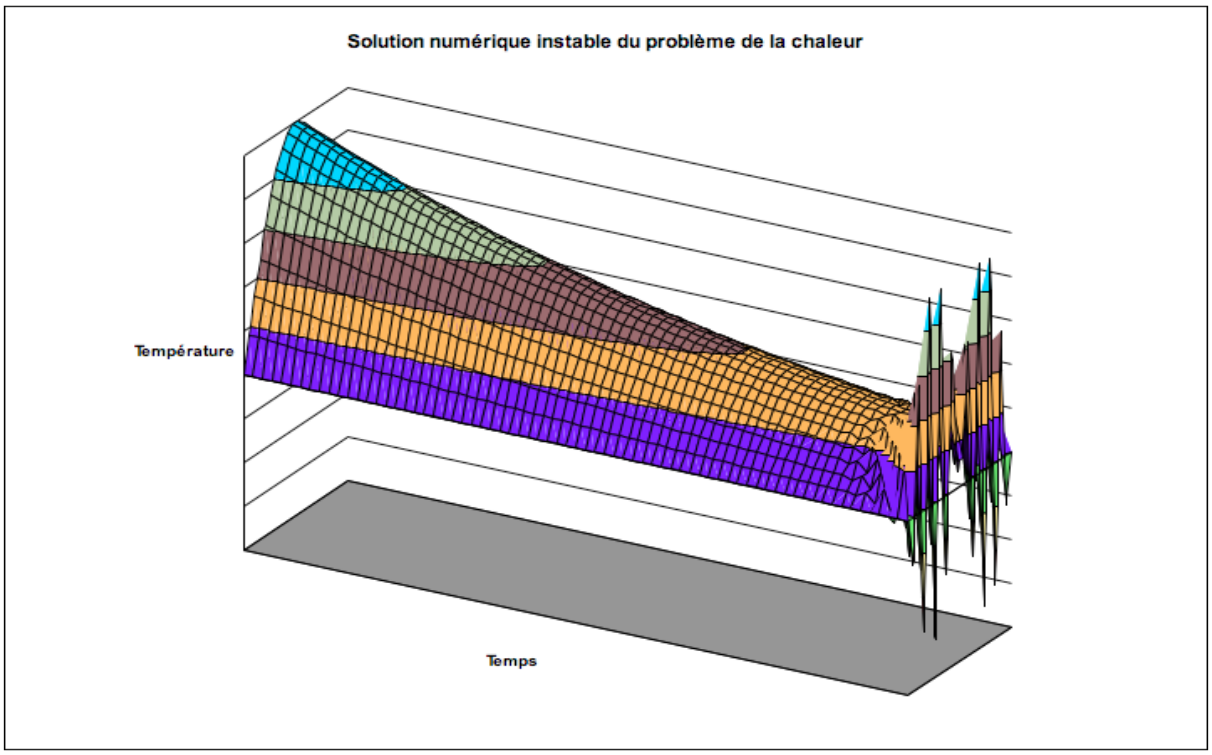
COMPTE RENDU DE TP

OBJECTIF: *ETUDE DE STABILITE ET PRECISION DE SCHEMAS NUMERIQUES DIFFERENCES FINIES INSTATIONNAIRES(1D) ET STATIONNAIRES(2D)*



PROFESSEUR: REMY Benjamin

ETUDIANTS: GANOU Arouna
GHARBI Mahdi

Le but de ce TP est d'approfondir la méthode des différences finies et de formaliser les différentes étapes du passage de l'équation du modèle jusqu'à l'obtention d'une solution numérique.

Nous supposons que nous avons déjà nos modèle, donc une équation. Il s'agit en général d'une équation au dérivées partielles. En plus du modèle, pour penser à utiliser des méthodes numériques, il faut avoir des conditions aux limites et des conditions initiales dans le cas instationnaire (nous supposons que le modèle faisant intervenir des dérivées temporelles)
En général, il n'est pas simple(voir impossible) de trouver une solution analytique au problème. Donc on a recourt aux méthodes numériques. Par contre il faut que notre problème vérifie certaines conditions pour avoir une solution satisfaisante. On parle de **problème bien posé**. Les définitions de cette notion sont nombreuses selon le domaine au le modèle issu(*thermique* avec les problèmes inverses par exemple, mécanique *des milieux continus*, etc). Nous allons citer ici la définition donnée par **Hadamard**:

1. La solution existe (On peut utiliser *l'analyse fonctionnelle* ou *les distributions)*
2. La solution est unique (On peut également utiliser également *l'analyse fonctionnelle* ou *les distributions*)
3. La solution dépend continûment des données dans une certaine topologie choisie.

Nous avons un problème dont on peut espérer trouver une solution numérique. Par contre pour l'approche numérique, il faut choisir une méthode parmi les méthodes bien connues: **Différence finies, volumes finis et éléments finis.** Dans ce TP, nous avons étudier uniquement la première méthode.
La méthode des différences finies consiste à résoudre un système liant les valeurs des fonctions en des points suffisamment proches les un des autres. Cet ensemble de points constitues le maillage choisi. Dans la suite, nous allons fixer un seul modèle qui est l'équation de la chaleur en un domaine 1D et instationnaire donné sur la fiche de TD:

Cette méthode procède en deux étapes:

1. Discrétisation de l'équation du modèle: On aboutit à un schéma **numérique**
   La discrétisation s'effectue par des **développements de Taylor** en un nœud du maillage choisi. C'est cette méthode que nous avons utiliser dans le TP. Nous pouvons également également utiliser **les différences finies généralisées** qui permet de mieux formaliser le problème pour les ordres assez élevés ou faisant intervenir des produit de fonctions inconnues.
   Il faut séparer la discrétisation spatiale et la discrétisation temporelle. En fait, c'est utile car c'est un enjeux de la stabilité des schémas. Lorsqu'on a la discrétisation spatiale, on deux choix :Soit on considère que ces valeurs de la fonction sont considéré à l'instant $\mathbf{t_n}$ ou à l'instant $\mathbf{t_{n+1}}$. Dans le premier cas, on parle de schéma ***explicite*** et dans le second cas de ***schéma implicite***. On a troisième schéma qui fait une pondération convexe des deux. C'est le schéma ***r-Schéma.***Pour ***r=0,5*** on obtient *le* ***schéma*** **de** ***Crank-Nicolson.*** On peut écrire donc:

   r-Schéma= r*(schéma implicite)+(1-r)*(Schéma explicite)

   Après la discrétisation, nous n'avons que des relations locales. Il faut alors assembler toutes ces relations en une relation liant tous les points du domaines en tenant compte du fait qu'on a choisit une méthode explicite ou implicite.
   Nous avons traité avec le maillage du TD et nous avons gardé la discrétisation temporelle

également.

$$B*U_{n+1}=A*U_n+k*F \quad (1)$$

C'est la structure générale de tout problème de ce type. U est le vecteur inconnu F est le vecteur des sollicitation. Nous l'avons considéré nul dans le calcul sous Matlab.

2. Étude de la convergence du schéma

Avec la schéma de la formule, nous devons étudier sa convergence pour voir si ce schéma est utile. D'abord le schéma doit être stable. *Un* ne doit pas avoir de composants qui tendent vers l'infini lorsque le pas de temps tend vers zéro et quelque soit la condition initiale car nous savons que les solutions sont bien finies.
En plus de la stabilité, le schéma doit être consistant: Lorsque tend vers l'infini, Un doit tendre vers la solution stationnaire lorsque n (dans la formule (1)) devient assez grand.

Dans ce TD, nous avons étudier la convergence du schéma explicite. Pour étudier la stabilité du schéma nous avons utiliser les valeurs propres de la matrice A de la formule(1). Nous avons trouver exactement la même condition de stabilité que celle obtenue par l'étude des valeurs propres de l'opérateur de cette équation(*Étude de la stabilité par la méthode de Von Neumann*).

$$\frac{\partial}{\partial t}-\alpha\frac{\partial^2}{\partial x^2}$$

La condition est la suivante:

$$\alpha\frac{dt}{(dx)^2}<\frac{1}{2}$$

Le schéma explicite est donc conditionnellement stable. Une étude avec la méthode de Von Neumann montre que le schéma implicite est in conditionnement stable.

Pour la consistance,on montre par récurrence cette relation entre U_n pour tout n et la condition initiale.

$$U_n=A^n*U_0+k*\sum_{p=0}^{p=n-1}A^p*F \; avec \; k=dt$$

Pour que U_n soit fini il faut que que les deux termes convergent:

- D'une part la convergence de A^n*U_0 : A^n doit tendre vers zéro Il faut que le rayon spectral de A $\rho(A)<1$
- Il faut que la somme partielle $k*\sum_{p=0}^{p=n-1}A^p*F \; avec \; k=dt$ converge. C'est série géométrique matricielle. Donc il faut également la condition précédente: $\rho(A)<1$.

Pour parler de la consistance, il faut nécessairement que $\rho(A)<1$.
Sous cette condition, en utilisant la formule de la série géométrique matricielle, on retrouve l'équation stationnaire discrétisée.

$$-\alpha\frac{v_{(j+1)}-2*v_j+v_{(j-1)}}{h^2}=f(xj)$$ Avec V le vecteur solution stationnaire.

En utilisant, les développements de Taylor, on trouve que ce schéma est précis d'ordre 2 en espace(dans la cas stationnaire) et précis d'ordre 1 en temps.

Cette étude a été faite dans le cas 1D et instationnaire.

Pendant la deuxième séance, nous avons travaillé sur l'équation de Poisson, qui correspondant à l'équation précédente dans le cas 2D et stationnaire.

Dans ce cas, un problème supplémentaire s'ajoute pour la matrice A de la relation (1) donc la taille augmentera(devient la taille précédente au carré pour notre maillage qui est la même suivant les directions OX et OY).

On choisit les pas de discrétisations suivantes: dx et dy .

Il faut une table d'allocation. Mais dès qu'on a la relation, il est relativement simple de construire la matrice A. Avec une fonction d'allocation bien choisie, on peut facilement résoudre l'équation de Poisson 3D.

Ensuite, notre problème avait une condition de Neumann sur le bord: $y=0$. Suivant la discrétisation choisie, on n'a pas la même précision. Si on utile un schéma d'ordre 1 pour la discrétisation:

$$\frac{\partial u}{\partial y}(x,0)=-g(x);\quad (2.3)$$

En choisissant un schéma d'ordre 1, la précision globale du problème discrétisé est en (dx^2+dy)

On doit choisir de faire des schémas d'au moins d'ordre 2 pour avoir des schémas d'ordre (dx^2+dy^2) . Là également, on a plusieurs possibilités, on privilégiera les schémas dont le points de discrétisation se trouve au centre(la figure suivante pour rendre la phrase plus claire).

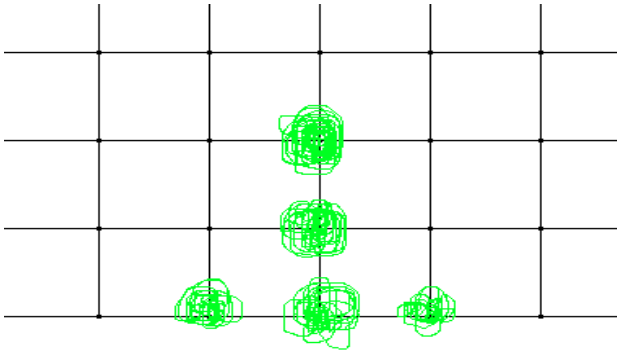


Figure1: Cas à privilégier

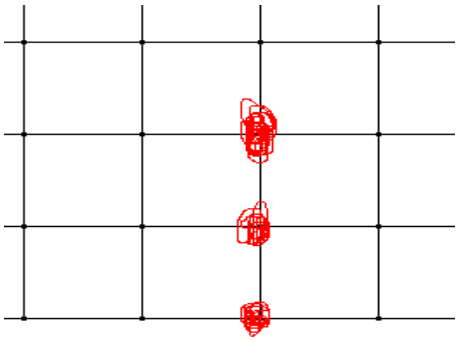


Figure2: Cas simple

Le cas à privilégier offre une vitesse de convergence plus forte mais ça demande plus de calcul. En effet, pour établir la formule, il faut discrétiser la condition de Neumann en utilisant l'EDP elle-même; et comme a plus de points dans la relation on plus de coefficient dans la matrice.

Après l'étude théorique, on va analyser les résultats obtenus avec Matlab.

I. SOLUTION D'UN PROBLEME UAX LIMITES D'EVOLUTION PAR LA METHODE DES DIFFERENCES FINIES

1. *STABILITÉ*

Sur les figures, chaque courbe est le profil de température sur le domaine(abscisse OX) à un certain instant valant k*dt.

Schéma explicite:

Nous allons montré que ce schéma est conditionnellement stable. La condition de stabilité est $\alpha\frac{dt}{(dx)^2}<\frac{1}{2}$. Les figures suivantes vont montrer suivant les valeurs de $\alpha\frac{dt}{(dx)^2}$ la stabilité du schéma explicite.

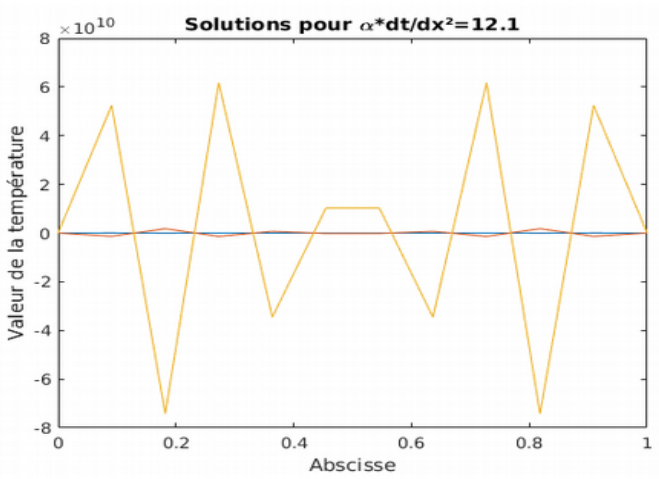


Figure3:Instable et von Neumann clairement non vérifiée(12.1>0.5)

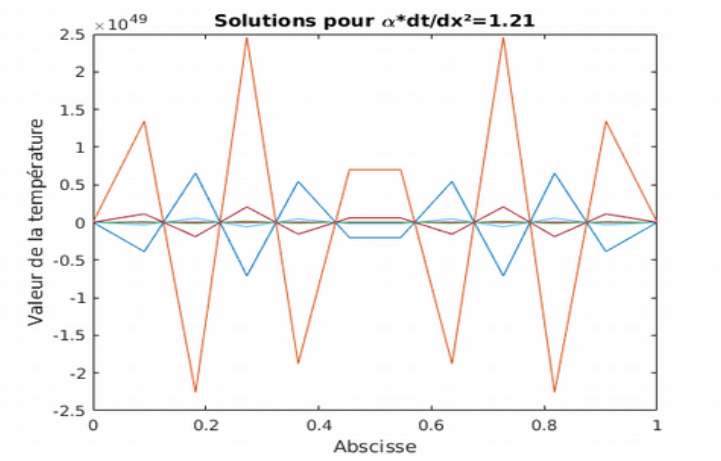


Figure4:Instable et von Neumann clairement non vérifiée(1.21>0.5)

Instabilité: valeur de la température de l'ordre 10^{49} !

Nous avons diminué la valeur du paramètre en diminuant dt, donc en augmentant le nombre de nombre de point de temps. On pourrai bien aussi augmentant dx.

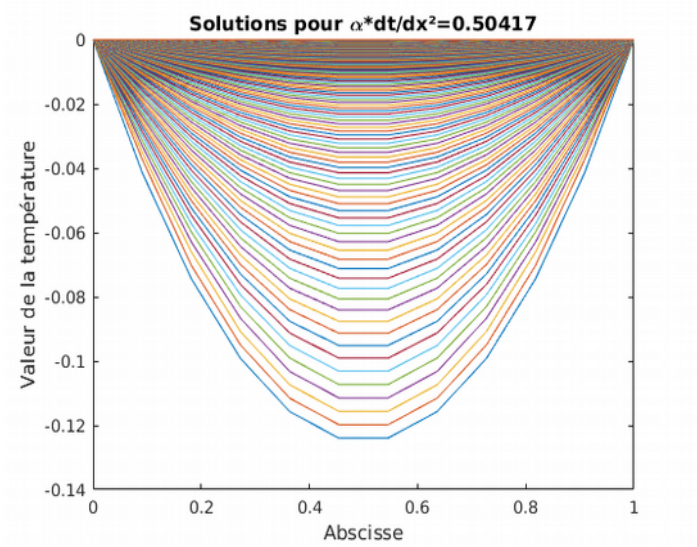


Figure5: Condition de von Neumann non vérifiée mais schéma stable.

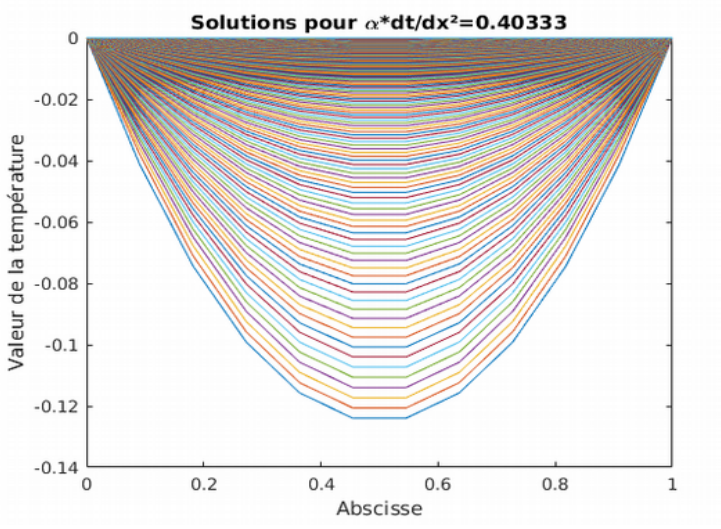


Figure6: Condition de von Neumann vérifiée

La figure 5 montre un schéma stable ne vérifiant pas la condition de Von Neumann. Ceci est dû au

faite que la condition de Von Neumann est une condition suffisant mais non nécessaire. Lorsqu'on établissait cette formule, on a pris une borne inférieure dans la majoration de $\alpha \frac{dt}{(dx)^2}$.

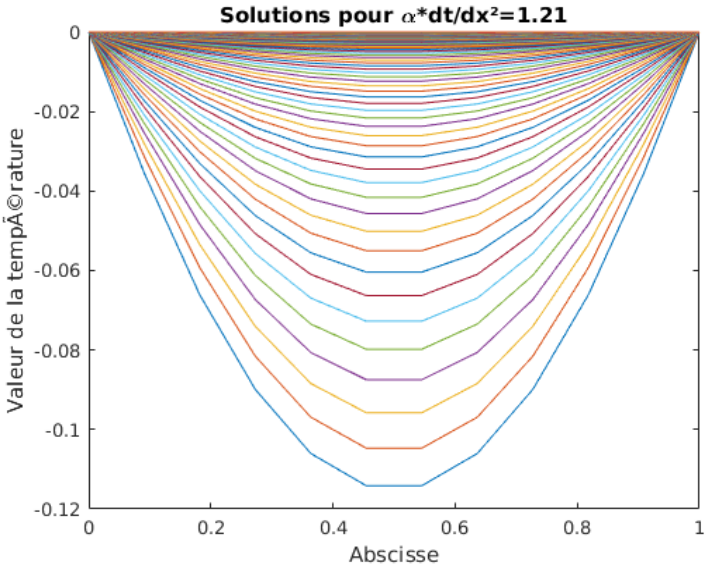
<CODE_SCHEMA_EXPLICITE>

```
%% Paramètre du problème Ã  résoudre
alpha=1;% paramètre de la modélisation physique
%% Borne du domaine
xmin=0;
xmax=1;
tc=1;% temps caracteristique
%% Discretisation du domaine
N=10;%
dx=xmax/(N+1);% pas en espace
nt=300;
dt=tc/nt;
% facteur apparaissant frequemment dans nos schemas, on  a preferé le
% poser de la sorte pour faciler l'écriture dans la suite.
theta=alpha*dt/dx^2;
x=0:dx:1;% variable spaciale discretisée
% Taille du vecteur espace
[~,N]=size(x);

%k=1e-3;
%On doit enlever 2 par la suite car on veut travailler sur les solutions
% sont cherchées pour les noeuds internes, les valeurs deux noeuds externes
% etant imposés par les conditions aux limites nul de Dirichlet dans notre cas
% actuels.)
%% Creation d'une matrice triadiagonale,
% c'est une matrice recurrente dans la plus part des problÃ¨me faisant
% intervenir des derivÃ©es d'ordres 2.
C=full(gallery('tridiag',N-2,-1,2,-1));
gamma=alpha*dt/(dx)^2; % Paramètre modulant la stabilitÃ© dans la methode von
Neumann
%% Schema explicite
Aexp=eye(N-2)-theta*C;
Bexp=eye(N-2);

%% Solution initiale
U0=(x.*(x-1)/(2*alpha))';
%% Solution par methode explicite
Solexp=U0(2:end-1,:);
Bin=Bexp\eye(N-2);
U1=U0(2:end-1,:);
for i=1:nt
    U2=Bin*Aexp*U1;
    plot(x,[0;U1;0];%Les zeros sont pour les conditions aux limites de Dirichlet
    xlabel('Abscisse')
    ylabel('Valeur de la tempÃ©rature')
    titre=sprintf('Solutions pour \\alpha*dt/dxÂ²=%0.5g',gamma);
    title(titre);
    hold on
    U1=U2;
    Solexp=[Solexp U1];
end
%Conditions aux bords  de Dirichlet
```

Schéma implicite:

Les figures suivantes montrent que le schéma explicite est inconditionnellement stable, mais aux prix d'une inversion de matrice.

Même si ce n'est pas nécessaire, nous avons laissé les valeurs de $\alpha \frac{dt}{(dx)^2}$.



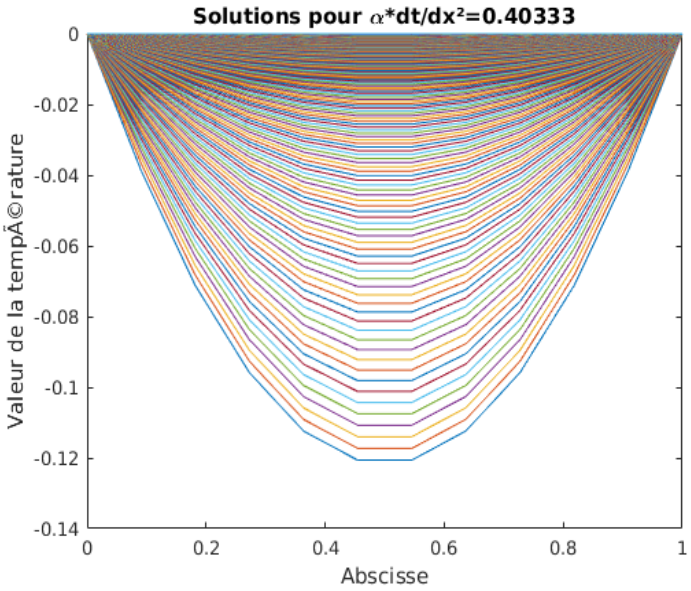


Figure7: Schéma implicite sans la conditionne de von Neumann

Figure8: Schéma implicite avec la la condition de von Neumann

Nous allons prendre un cas assez extrême pour voir qu'on n' a pas besoin de la condition de von Neumann pour rester stable(des pas de temps et d'espace assez petit, beaucoup matlab prendra beaucoup de temps).

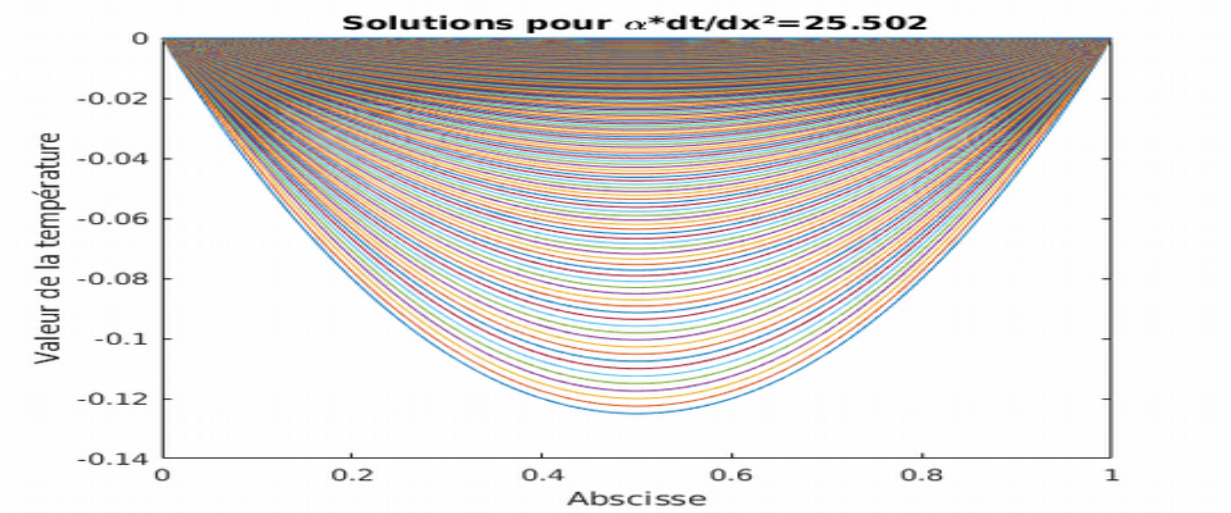


Figure 9: Schéma implicite avec des pas très petits

<CODE_SCHEMA_IMPLICITE>

```
%% Paramètre du problème à  résoudre
alpha=1;% parametre de la modelisation physique
%% Borne du domaine
xmin=0;
xmax=1;
```

```
tc=1;% temps caracteristique
%% Discretisation du domaine
N=100;%
dx=xmax/(N+1);% pas en espace
nt=400;
dt=tc/nt;
% facteur apparaissant frequemment dans nos schemas, on  a prefererr le
% poser de la sorte pour faciler l'ecriture dans la suite.
theta=alpha*dt/dx^2;
x=0:dx:1;% variable spaciale discretisree
% Taille du vecteur espace
[~,N]=size(x);
%k=1e-3;
%On doit enlever 2 par la suite car on veut travailler sur les solutions
% sont cherchÃ©es pour les noeuds internes, les valeurs deux noeuds externes
% etant imposers par les conditions aux limites nul dans notre cas
% actuels.)
%% Creation d'une matrice triadiagonale,
% c'est une matrice recurrente dans la plus part des probleme faisant
% intervenir des derivees d'ordres 2.
C=full(gallery('tridiag',N-2,-1,2,-1));
gamma=alpha*dt/(dx)^2; % Paramètre modulant la stabilité dans la methode von
Neumann

%% Schema implicite
Aimp=eye(N-2);
Bimp=eye(N-2)+theta*C;

%% Solution initiale
U0=(x.*(x-1)/(2*alpha))';


%% Solution methode implicite
Solimp=U0(2:end-1,:);

Bin=Bimp\eye(N-2);

U1=U0(2:end-1,:);
for i=1:nt
    U2=Bin*Aimp*U1;
    plot(x,[0;U1;0])%les zeros sont pour les conditions aux bords  de Dirichlet
    U1=U2;
    hold on
    Solimp=[Solimp U1];

end

xlabel('Abscisse')
ylabel('Valeur de la température')
titre=sprintf('Solutions pour \\alpha*dt/dx²=%0.5g',gamma);
title(titre);
```

Schéma Crank-Nicolson(r-schéma avec r=0.5)

Ce schéma est également inconditionnellement stable comme le schéma implicite. Les commentaires sur la stabilité du schéma implicite sont tous valables pour ce schéma.

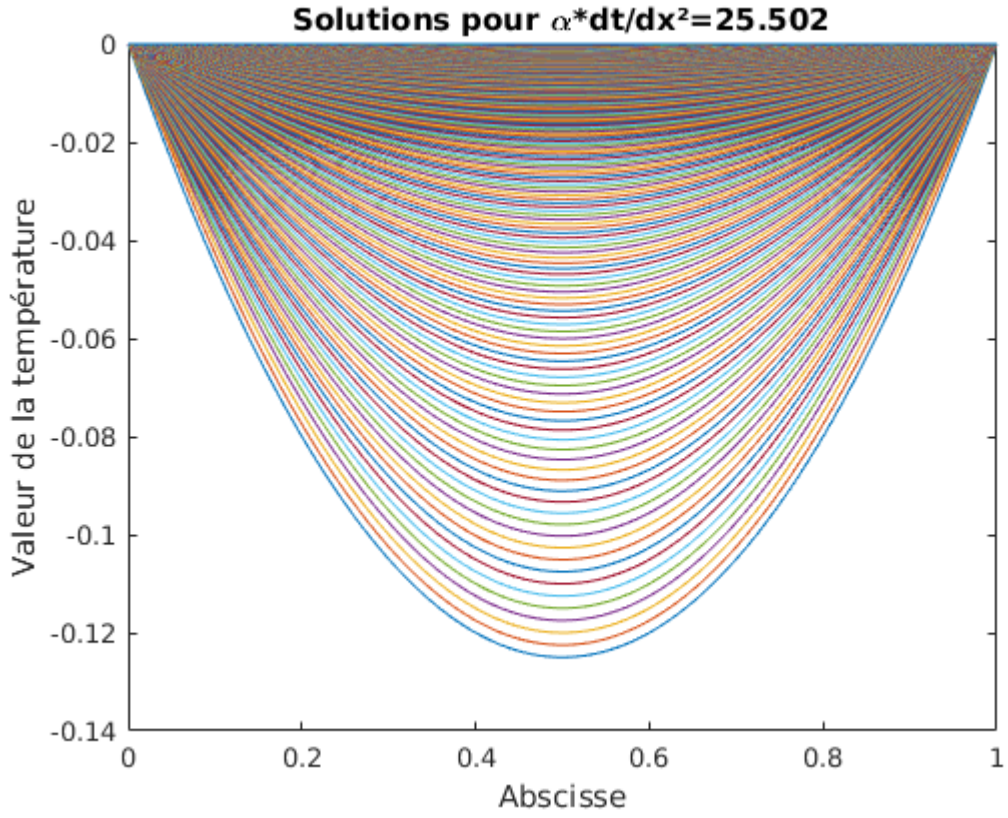


Figure 10: Schéma implicite avec des pas très petits

<CODE_SCHEMA_CRANK-NICOLSON>

```
%% Paramètre du problème à  résoudre
alpha=1;% parametre de la modelisation physique
%% Borne du domaine
xmin=0;
xmax=1;
tc=1;% temps caracteristique
%% Discretisation du domaine
N=100;%
dx=xmax/(N+1);% pas en espace
nt=400;
dt=tc/nt;
% facteur apparaissant frequemment dans nos schemas, on  a prefererr le
% poser de la sorte pour faciler l'ecriture dans la suite.
theta=alpha*dt/dx^2;
x=0:dx:1;% variable spaciale discretisree
% Taille du vecteur espace
[~,N]=size(x);
%On doit enlever 2 par la suite car on veut travailler sur les solutions
% sont cherchÃ©es pour les noeuds internes, les valeurs deux noeuds externes
% etant imposers par les conditions aux limites nul dans notre cas
% actuels.)
%% Creation d'une matrice triadiagonale,
```

```
% c'est une matrice recurrente dans la plus part des probleme faisant
% intervenir des derivees d'ordres 2.
C=full(gallery('tridiag',N-2,-1,2,-1));
gamma=alpha*dt/(dx)^2; % ParamÃ¨tre modulant la stabilitÃ© dans la methode von
Neumann
%% Schema implicite
Aimp=eye(N-2);
Bimp=eye(N-2)+theta*C;
%% Schema explicite
Aexp=eye(N-2)-theta*C;
Bexp=eye(N-2);
%% r_Schema
r=1/2; % Donc le schema de Crank-Nicolson par dÃ©faut dans ce programme
Ar_sch= r*Aimp+(1-r)*Aexp;
Br_sch=r*Bimp+(1-r)*Bexp;
%% Solution initiale
U0=(x.*(x-1)/(2*alpha))';
%% Solution methode semi-implicite
Solr_sch=U0(2:end-1,:);
Bin=Br_sch\eye(N-2);
U1=U0(2:end-1,:);
for i=1:nt
    U2=Bin*Ar_sch*U1;
    plot(x,[0;U1;0])%Les zeros sont pour les conditions aux bords  de Dirichlet
    hold on
    U1=U2;
    Solr_sch=[Solr_sch U1];
end
xlabel('Abscisse')
ylabel('Valeur de la tempÃ©rature')
titre=sprintf('Solutions pour \\alpha*dt/dxÂ²=%0.5g',gamma);
title(titre);
```

Nous avons abordé dans cette section la stabilité des schémas, nous allons maintenant la précision des schémas. Pour le schéma explicite, nous allons rester dans les limites de stabilité.

2. PRECISION

Solution analytique

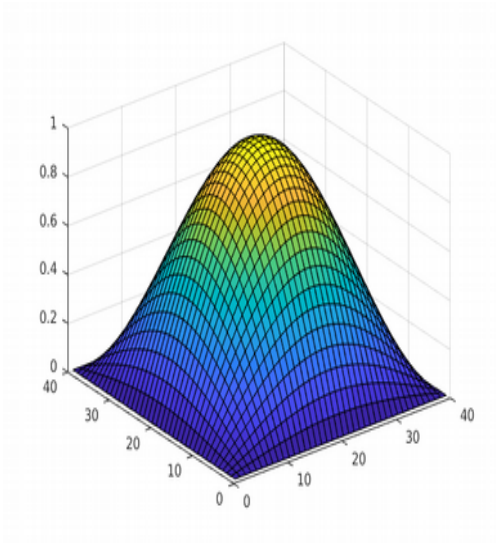
On a trouvé la famille de solutions analytiques non nulles de ce problème par séparation des variables. Ensuite par linéarité de l'équation: $\frac{\partial U(x,t)}{\partial t}-\alpha\frac{\partial^2 U(x,t)}{\partial x^2}=f(x)$, toute combinaison linéaire des éléments cette famille est une solution de l'équation. Après quelques calculs, on trouve cette formule pour la solution analytique:

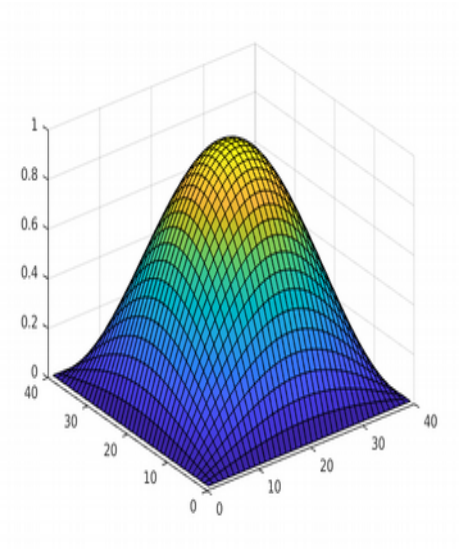
$$U(x,t)=\sum_{n=0}^{+\infty} 2*\int_0^1 U(u,t=0)\sin(n*\pi u)\,du\sin(n*\pi x)\,e^{-n^2*pi^2*\alpha t}$$

La série converge à cause du terme en exponentielle, en plus on peut montrer que la converge est relativement assez rapide. Pour nos calcul en Matlab, nous l'avons tronqué.
Nous allons utiliser cette solution analytique pour étudier la précision des schémas.

II. TRAITEMENT DE L'EQUATION DE POISSON DANS LE PLAN

1. *DISCRETISATION DE LA CONDITION DE NEUMANN*





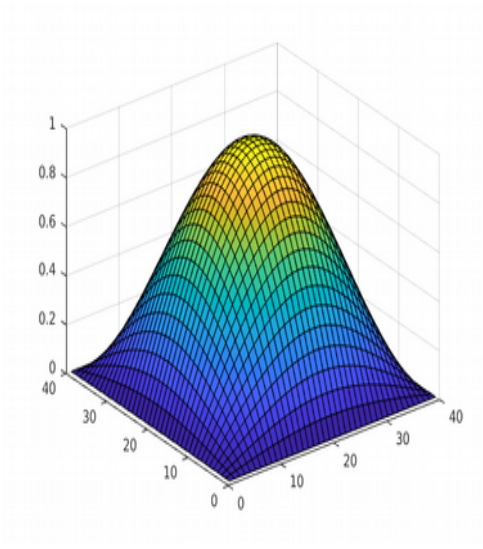
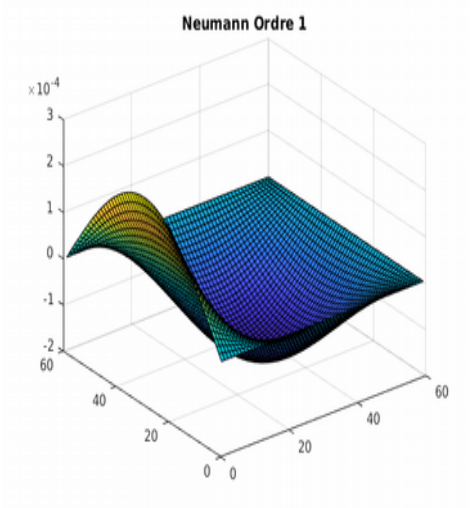


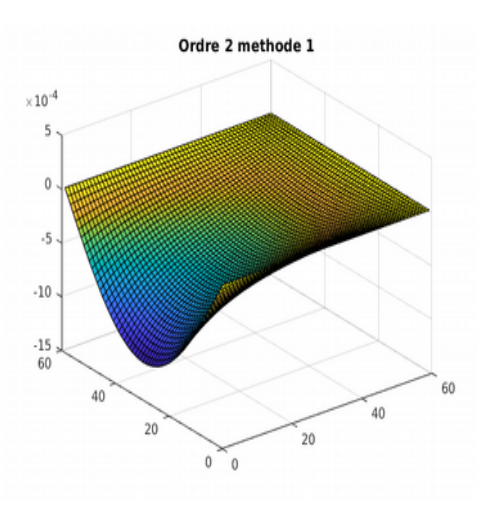
Figure 17: Neumann ordre 1 ***Figure 18: Neumann ordre 2 méthode 1*** ***Figure 19: Neumann ordre2 méthode 2***

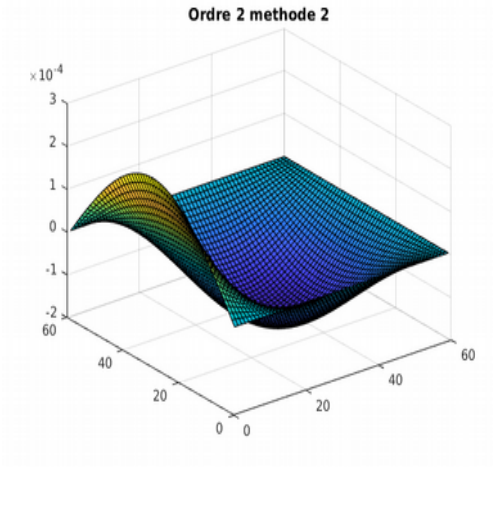
Nous avons avons tracé les figures ci-dessus avec 41 points en x et 41 points en y. En regardant simplement, il n'y a pas une grande différence entre ces trois solutions.

2. *PRESICION*

Nous allons nous contenter d'analyser l'erreur en traçant, la différence entre la solution analytique et les solutions numériques.







Nous constatons que la méthode d'ordre 1 a un meilleur comportement que la méthode 1 d'ordre 2.

<CODE_POISSON_AVEC_NEUMANN_ORDRE_1>

```
%% Definition du domaine
lx=1;
ly=1;

%% Definition des nombres de points de maillage
%pas de temps
Nx=60;
Ny=60;
h=lx/(Nx-1);
k=ly/(Ny-1);
x=0:h:lx;
y=0:k:ly;
% Constantes recurrentes dans les schemas
a=sqrt(k^2/(2*(h^2+k^2)));
b=sqrt(h^2/(2*(h^2+k^2)));
c=sqrt(h^2*k^2/(2*(h^2+k^2)));


%% Assemblage de A et F
A=zeros(Nx*Ny);
F=zeros(Nx*Ny,1);

for j=1:Ny
    for i=1:Nx
        k1=(j-1)*Nx+i;
        k2=(j-1)*Nx+i-1;
        k3=(j-1)*Nx+i+1;
        k4=(j-1-1)*Nx+i;
        k5=(j-1+1)*Nx+i;
        if i>1 && i<Nx && j>1 && j<Ny
            F(k1,1)=-c^2*f(x(i),y(j),lx,ly);
            A(k1,k1)=1;
            A(k1,k2)=-a^2;
            A(k1,k3)=-a^2;
            A(k1,k4)=-b^2;
            A(k1,k5)=-b^2;
        end
        if i==1 || i==Nx || j==Ny
            F(k1,1)=0;
            A(k1,k1)=1;
        end
        if j==1 && i>1 && i<Nx
            F(k1,1)=-k*g(x(i),lx,ly);
            A(k1,k1)=-1;
            A(k1,k5)=1;
        end


    end
end

%% Solution
K=1; % par recherche de valeur propre du laplacien
Ue=K*sin(pi*x/lx)'*sin(pi*y/ly);

%% Solution par DF
U=A\F;

Us=reshape(U,Nx,Ny);
figure();surf(Ue-Us);
```

```
title('Neumann Ordre 1')
```

<CODE_POISSON_AVEC_NEUMANN_ORDRE_2_METHODE_1>

```
%% Definition du domaine
lx=1;
ly=1;

%% Definition des nombres de points de maillage
%pas de temps
Nx=60;
Ny=60;
h=lx/(Nx-1);
k=ly/(Ny-1);
x=0:h:lx;
y=0:k:ly;
% Constantes recurrentes dans les schemas
a=sqrt(k^2/(2*(h^2+k^2)));
b=sqrt(h^2/(2*(h^2+k^2)));
c=sqrt(h^2*k^2/(2*(h^2+k^2)));


%% Assemblage de A et F
A=zeros(Nx*Ny);
F=zeros(Nx*Ny,1);

for j=1:Ny
    for i=1:Nx
        k1=(j-1)*Nx+i;
        k2=(j-1)*Nx+i-1;
        k3=(j-1)*Nx+i+1;
        k4=(j-1-1)*Nx+i;
        k5=(j-1+1)*Nx+i;
        k6=(j-1+2)*Nx+i;
        if i>1 && i<Nx && j>1 && j<Ny
            F(k1,1)=-c^2*f(x(i),y(j),lx,ly);
            A(k1,k1)=1;
            A(k1,k2)=-a^2;
            A(k1,k3)=-a^2;
            A(k1,k4)=-b^2;
            A(k1,k5)=-b^2;
        end
        if i==1 || i==Nx || j==Ny
            F(k1,1)=0;
            A(k1,k1)=1;
        end
        if j==1 && i>1 && i<Nx
            F(k1,1)=-2*k*g(x(i),lx,ly);
            A(k1,k1)=-3;
            A(k1,k5)=4;
            A(k1,k6)=-1;
        end


    end
end
```

```
%% Solution
K=1; % par recherche de valeur propre du laplacien
Ue=K*sin(pi*x/lx)'*sin(pi*y/ly);

%% Solution par DF
U=A\F;

Usm1=reshape(U,Nx,Ny);
figure();surf(Ue-Usm1);
title('Ordre 2 methode 1')
```

<CODE_POISSON_AVEC_NEUMANN_ORDRE_2_METHODE_2>

```
%% Definition du domaine
lx=1;
ly=1;

%% Definition des nombres de points de maillage
%pas de temps
Nx=60;
Ny=60;
h=lx/(Nx-1);
k=ly/(Ny-1);
x=0:h:lx;
y=0:k:ly;
% Constantes recurrentes dans les schemas
a=sqrt(k^2/(2*(h^2+k^2)));
b=sqrt(h^2/(2*(h^2+k^2)));
c=sqrt(h^2*k^2/(2*(h^2+k^2)));


%% Assemblage de A et F
A=zeros(Nx*Ny);
F=zeros(Nx*Ny,1);

for j=1:Ny
    for i=1:Nx
        k1=(j-1)*Nx+i;
        k2=(j-1)*Nx+i-1;
        k3=(j-1)*Nx+i+1;
        k4=(j-1-1)*Nx+i;
        k5=(j-1+1)*Nx+i;
        k6=(j-1+2)*Nx+i;
        if i>1 && i<Nx && j>1 && j<Ny
            F(k1,1)=-c^2*f(x(i),y(j),lx,ly);
            A(k1,k1)=1;
            A(k1,k2)=-a^2;
            A(k1,k3)=-a^2;
            A(k1,k4)=-b^2;
            A(k1,k5)=-b^2;
        end
        if i==1 || i==Nx || j==Ny
            F(k1,1)=0;
            A(k1,k1)=1;
        end
        if j==1 && i>1 && i<Nx
            F(k1,1)=2*k*b^2*g(x(i),lx,ly)-c^2/2*f(x(i),y(j),lx,ly);
            A(k1,k1)=1;
```

```
            A(k1,k2)=-a^2;
            A(k1,k3)=-a^2;
            A(k1,k5)=-2*b^2;
        end


    end
end

%% Solution
K=1; % par recherche de valeur propre du laplacien
Ue=K*sin(pi*x/lx)'*sin(pi*y/ly);

%% Solution par DF
U=A\F;

Usm2=reshape(U,Nx,Ny);
figure();surf(Ue-Usm2);
title('Ordre 2 methode 2')
```