



Rapport Master M1

Master Calcul Haute Performance Simulation (CHPS)

# TP Technique de Modélisation

Réalisé par: Aicha Maaoui et Lydia Chouaki

Janvier 2022

Institut des Sciences et Techniques des Yvelines (ISTY)

# Abstract

Ce rapport comporte:

- Le mini-projet du TD1, exercice7 dans le chapitre 1.
- Le TP de technique de modélisation dans le chapitre 2.

# Contents

<b>1</b>	<b>Mini-Projet, Exercice7 TD1</b>	<b>1</b>
<b>2</b>	<b>TP Diffusion de la chaleur</b>	<b>16</b>
2.1	Préliminaire . . . . .	16
2.2	Solution Analytique . . . . .	17
2.3	Solution Numérique: Schéma Explicite: . . . . .	22
2.4	Solution Numérique: Schéma Implicite: . . . . .	30
2.5	Solution Numérique: Schéma de Crank-Nicholson: . . . . .	36
	<b>References</b>	<b>44</b>
<b>A</b>	<b>Codes des Schémas Numériques en C</b>	<b>45</b>
<b>A</b>	<b>Dépot Github</b>	<b>50</b>

# List of Figures

1.1	Variation des erreurs des schémas d'Euler implicite et explicite en fonction du pas du temps $\Delta t$ . . . . .	7
1.2	Variation des erreurs du schéma numérique en fonction du pas du temps $\Delta t^2$ . . .	7
1.3	Variation des erreurs des schémas d'Euler implicite et explicite en fonction du pas du temps $\Delta t$ , système d'équation (1.26). . . . .	14
1.4	Variation des erreurs du schéma numérique en fonction du pas du temps $\Delta t^2$ , système d'équation (1.26). . . . .	14
1.5	Résultats du schéma d'Euler explicite en fonction du nombre de points $m$ avec <i>Scilab</i> , système d'équations (1.1). . . . .	15
2.1	Evolution de l'état de la barre $[0, 1]$ dans le temps (Maillage en temps et en espace). .	17
2.2	Variation de $u(x, t)$ en fonction de $x$ dans le cas du schéma explicite, $f = f_1$ et $t$ fixe. . . . .	27
2.3	Variation de $u(x, t)$ en fonction d'espace $x$ et du temps $t$ dans le cas du schéma explicite, $f = f_1$ . . . . .	28
2.4	Variation de $u(x, t)$ en fonction de $x$ dans le cas du schéma explicite, $f = f_2$ et $t$ fixe. . . . .	29
2.5	Variation de $u(x, t)$ en fonction de $x$ et $t$ dans le cas du schéma explicite, $f = f_2$ . .	29
2.6	Variation de $u(x, t)$ en fonction de $x$ dans le cas du schéma implicite, $f = f_1$ et $t$ fixe. . . . .	33
2.7	Variation de $u(x, t)$ en fonction d'espace $x$ et du temps $t$ dans le cas du schéma implicite, $f = f_1$ . . . . .	34
2.8	Variation de $u(x, t)$ en fonction de $x$ dans le cas du schéma implicite, $f = f_2$ et $t$ fixe. . . . .	35

---

2.9	Variation de $u(x, t)$ en fonction de $x$ et $t$ dans le cas du schéma implicite, $f = f_2$ .	35
2.10	Variation de $u(x, t)$ en fonction de $x$ dans le cas du schéma de Crank-Nicholson, $f = f_1$ et $t$ fixe. . . . .	40
2.11	Variation de $u(x, t)$ en fonction d'espace $x$ et du temps $t$ dans le cas du schéma de Crank-Nicholson, $f = f_1$ . . . . .	41
2.12	Variation de $u(x, t)$ en fonction de $x$ dans le cas du schéma de Crank-Nicholson, $f = f_2$ et $t$ fixe. . . . .	42
2.13	Variation de $u(x, t)$ en fonction de $x$ et $t$ dans le cas du schéma de Crank- Nicholson, $f = f_2$ . . . . .	42

# List of Tables

1.1	Variation des erreurs en fonction du nombre de points $m$ et du pas du temps $\Delta t$ .	6
1.2	Pente des courbes des erreurs en fonction du pas du temps $\Delta t$ . . . . .	8
1.3	Variation des erreurs en fonction du nombre de points $m$ et du pas du temps $\Delta t$ , système d'équation (1.26). . . . .	13
1.4	Pente des courbes des erreurs en fonction du pas du temps $\Delta t$ , système d'équation (1.26). . . . .	13

# Chapter 1

## Mini-Projet, Exercice7 TD1

On considère le problème suivant:

$$\begin{cases} u'(t) &= -u(t) = f(t) \\ u(0) &= 1 \end{cases} \quad (1.1)$$

1/ **Solution exacte de ce problème:**

La solution exacte du problème donné dans l'équation (1.1) est:

$$u(t) = u(0) \times e^{-t} = e^{-t} \quad (1.2)$$

2/ **Méthode d'Euler explicite:**

La méthode d'Euler explicite nous donne:

$$U_{m+1} = U_m + h \times f(U_m) = U_m + h \times (-U_m) = (1 - \Delta t) \times U_m \quad (1.3)$$

avec  $\Delta t = h$ .

3/ **Convergence de la solution en  $t = 1$  quand  $\Delta t = 0$ :**

On a:

$$U_m = (1 - \Delta t)^m, \quad m = 0, 1, \dots \quad (1.4)$$

Soit le pas du temps  $\Delta t$  défini par l'équation (1.5), comme suit:

$$\Delta t = \frac{1-0}{m} = \frac{1}{m}, \quad \text{sur } [0, 1] \quad (1.5)$$

En remplaçant  $t$  par  $t = 1$  dans l'équation (1.2), on obtient:

$$u(1) = e^{-1} = \frac{1}{e} \quad (1.6)$$

Calculons maintenant la limite de  $U_m$  définie dans l'équation (1.4):

$$\lim_{\Delta t \rightarrow 0} (1 - \Delta t)^m = \lim_{\Delta t \rightarrow 0} (1 - \Delta t)^{\frac{1}{\Delta t}} \quad (1.7)$$

On applique  $\ln()$  à l'équation (1.7). soit l'équation suivante:

$$\ln((1 - \Delta t)^{\frac{1}{\Delta t}}) = \frac{1}{\Delta t} \times \ln(1 - \Delta t) \quad (1.8)$$

Le taux d'accroissement de l'équation (1.8) est donné dans l'équation (1.9). Soit:

$$\frac{\ln(1 - \Delta t)}{\Delta t} = \frac{\ln(1 - \Delta t) - \ln(1)}{\Delta t} \quad (1.9)$$

L'expression donnée dans l'équation (1.9) tend vers  $f'(0)$  quand  $\Delta t \rightarrow 0$ .

avec:

$$f(x) = \ln(1 - x) \rightarrow f'(x) = \frac{-1}{1 - x} \quad (1.10)$$

D'après les équations (1.9) et (1.10), la limite donnée dans l'équation (1.7) est:

$$\lim_{\Delta t \rightarrow 0} (1 - \Delta t)^m = e^{\frac{-1}{1-0}} = e^{-1} \quad (1.11)$$

4/ **Montrer que:**  $\frac{u(t_{m+1}) - u(t_m)}{\Delta t} \simeq u'(t_{m+1}) = f(u(t_{m+1}))$

On considère le développement de Taylor à l'ordre 1 de  $u(t_m)$ . On aura:

$$u(t_m) = u(t_{m+1} - \Delta t) = u(t_{m+1}) - \Delta t \times u'(t_{m+1}) + O(\Delta t) \quad (1.12)$$



D'où:

$$\frac{u(t_{m+1}) - u(t_m)}{\Delta t} \simeq u'(t_{m+1}) = f(u(t_{m+1})) \quad (1.13)$$

**5/ Schéma d'Euler Implicite:**

Le schéma d'Euler implicite est donné par une déduction à partir de l'équation (1.13). Ainsi, il est donné par:

$$\frac{u_{m+1} - u_m}{\Delta t} \simeq f(u_{m+1}) = -u_{m+1} \quad (1.14)$$

**6/ Déduction sur l'expression de  $u_m$ :**

On a d'après l'équation (1.14),

$$u_{m+1} + \Delta t \times u_{m+1} = u_m \quad (1.15)$$

D'où:

$$u_{m+1} \times (1 + \Delta t) = u_m \quad (1.16)$$

D'après l'équation (1.16):

$$u_{m+1} = \frac{u_m}{1 + \Delta t} \quad (1.17)$$

Il s'agit d'une suite géométrique. Son expression est donnée dans l'équation (1.18).

$$u_{m+1} = \frac{u_0}{(1 + \Delta t)^m} = \frac{1}{(1 + \Delta t)^m} \quad (1.18)$$

7/ On a le système d'équation suivant:

$$\begin{cases} \frac{u(t_{m+1}) - u(t_m)}{\Delta t} = f(u(t_{m+1})) & (1) \\ \frac{u(t_{m+1}) - u(t_m)}{\Delta t} = f(u(t_m)) & (2) \end{cases} \quad (1.19)$$

Faisant la moyenne de (1) et (2). Soit l'équation:

$$\frac{u(t_{m+1}) - u(t_m)}{\Delta t} = \frac{f(u(t_{m+1})) + f(u(t_m))}{2} \quad (1.20)$$

**8/ Dédution du schéma numérique:**

De l'équation (1.19), on en déduit l'équation suivante:

$$u(t_{m+1}) - \frac{\Delta t}{2} f(u(t_{m+1})) = u(t_m) + \frac{\Delta t}{2} f(u(t_m)), \quad m = 0, 1, \dots \quad (1.21)$$

**9/ Solution numérique:**

On a d'après l'équation (1.20):

$$u(t_{m+1}) + \frac{\Delta t}{2} u(t_{m+1}) = u(t_m) - \frac{\Delta t}{2} u(t_m) \quad (1.22)$$

On regroupe les coefficients en  $u(t_{m+1})$  et  $u(t_m)$  de l'équation précédente dans l'équation suivante:

$$u(t_{m+1}) \times \left(1 + \frac{\Delta t}{2}\right) = u(t_m) \times \left(1 - \frac{\Delta t}{2}\right) \quad (1.23)$$

De l'équation (1.22), on aura l'équation:

$$u(t_{m+1}) = u(t_m) \times \frac{2 - \Delta t}{2 + \Delta t} \quad (1.24)$$

D'où la suite géométrique suivante:

$$u(t_m) = \left(\frac{2 - \Delta t}{2 + \Delta t}\right)^m \times u_0 = \left(\frac{2 - \Delta t}{2 + \Delta t}\right)^m \quad (1.25)$$

**10/ Précision des trois méthodes:**

Les programmes des trois méthodes, ainsi que la solution analytique exacte sont donnés dans listings de (1.1) à (1.4), respectivement.

Listing 1.1: Programme du schéma d'Euler explicite.

```
function [u_explicite]=euler_explicite(t,m)

    u_explicite(1)= 1; %solution a t =0
    dt= 1/m; %pas du temps
    for i = 1:m
        u_explicite(i+1) = (1-dt) * u_explicite (i); %u(m+1)
    end
endfunction
```

Listing 1.2: Programme du schéma d'Euler implicite.

```
function [u_implicite]=euler_implicite(t,m)

    u_implicite(1)= 1; %solution a t =0
    dt= 1/m; %pas du temps
    for i = 1:m
        u_implicite(i+1) = u_implicite(i) / (1+dt); %u(m+1)
    end
endfunction
```

Listing 1.3: Programme du schéma numérique.

```
function [u_numerique]=solution_numerique(t,m)

    u_numerique(1)=1; %solution a t =0
    dt= 1/m; %pas du temps
    for i = 1:m
        u_numerique(i+1) = ((2-dt)/(2+dt)) * u_numerique(i); %u(m+1)
    end
endfunction
```

Listing 1.4: Programme de la solution exacte du problème.

```
function [u_exacte]= solution_exacte(t)

    u_exacte(t) = exp(-t); %solution exacte

endfunction
```

Le programme donnant la précision des trois méthodes par rapport à la solution exacte pour  $t = 1$  est donné dans listing (1.5).

Listing 1.5: Programme de la précision des trois méthodes par rapport à la solution exacte.

```
t =1; %temps fixe a t =1
m=20; %nombre de points (varie a chaque fois)
dt= 1/m; %pas du temps

[u_exacte]= solution_exacte(t); %solution exacte
[u_numerique]=solution_numerique(t,m); %solution numerique
[u_explicite]=euler_explicite(t,m); %Euler explicite
[u_implicite]=euler_implicite(t,m); %Euler implicite

% ***Calcul des Erreurs relatives***

erreur_num= norm(u_exacte - u_numerique(m+1))/norm(u_exacte); %precision schema
numerique
disp(erreur_num); %Erreur numerique
```

```

erreur_explicite= norm(u_exacte - u_explicite(m+1))/norm(u_exacte); %precision
    Euler explicite
disp(erreur_explicite); %Erreur explicite

erreur_implicite= norm(u_exacte - u_implicite(m+1))/norm(u_exacte); %precision
    Euler implicite
disp(erreur_implicite); %Erreur implicite

% ***Calcul et comparaison de k***

k_num = sqrt(erreur_num/dt^2); %k pour le schema numerique
disp(k_num);

k_explicite = erreur_explicite/dt; %k pour le schema explicite
disp(k_explicite);

k_implicite = erreur_implicite/dt; %k pour le schema implicite
disp(k_implicite);

```

On se propose dans ce qui suit de comparer la précision des trois méthodes pour un temps  $t = 1$  à partir de listing (1.5). Pour ce faire, on varie le nombre de points  $m$ , d'où la variation du pas du temps  $\Delta t$  (Equation (1.5)).

La variation des erreurs des trois méthodes considérées en fonction du pas du temps  $\Delta t$  et du nombre de points  $m$  est donné dans le tableau (1.1).

$m$	$\Delta t$	Schéma Numérique	Euler Explicite	Euler Implicite
10	0.1	0.0008342	0.0521937	0.0480153
20	0.05	0.0002084	0.0255342	0.0244918
30	0.0333	0.0000926	0.0169021	0.0164390
40	0.025	0.0000521	0.0126319	0.0123714
50	0.02	0.0000333	0.0100842	0.0099175
60	0.01667	0.0000231	0.0083917	0.0082759
70	0.014286	0.0000170	0.0071857	0.0071006
80	0.0125	0.0000130	0.0062828	0.0062176
90	0.0111	0.0000103	0.0055814	0.0055814
100	0.01	0.0000083	0.0050209	0.0049793

Table 1.1: Variation des erreurs en fonction du nombre de points  $m$  et du pas du temps  $\Delta t$ .

On trace ensuite la variation des erreurs en fonction du pas du temps  $\Delta t$ . Soit:

- La variation des erreurs des schémas d'Euler implicite et explicite en fonction du pas du temps  $\Delta t$ , comme illustré dans la figure (1.1),
- La variation des erreurs du schéma numérique en fonction du pas du temps  $\Delta t^2$ , comme illustré dans la figure (1.2).

Cherchons maintenant la pente des courbes  $erreur_i = f(\Delta t)$  avec  $i = \{\text{explicite, implicite}\}$  (figure (1.1)) et  $erreur_{numerique} = f(\Delta t^2)$  pour l'erreur numérique (figure (1.2)).

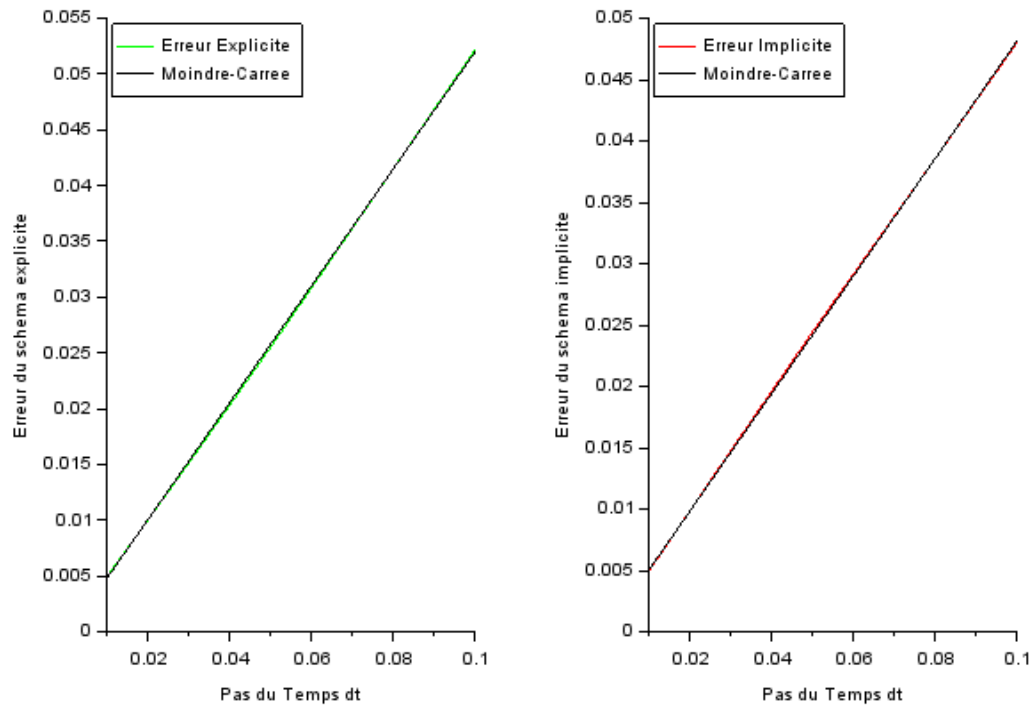


Figure 1.1: Variation des erreurs des schémas d'Euler implicite et explicite en fonction du pas du temps  $\Delta t$ .

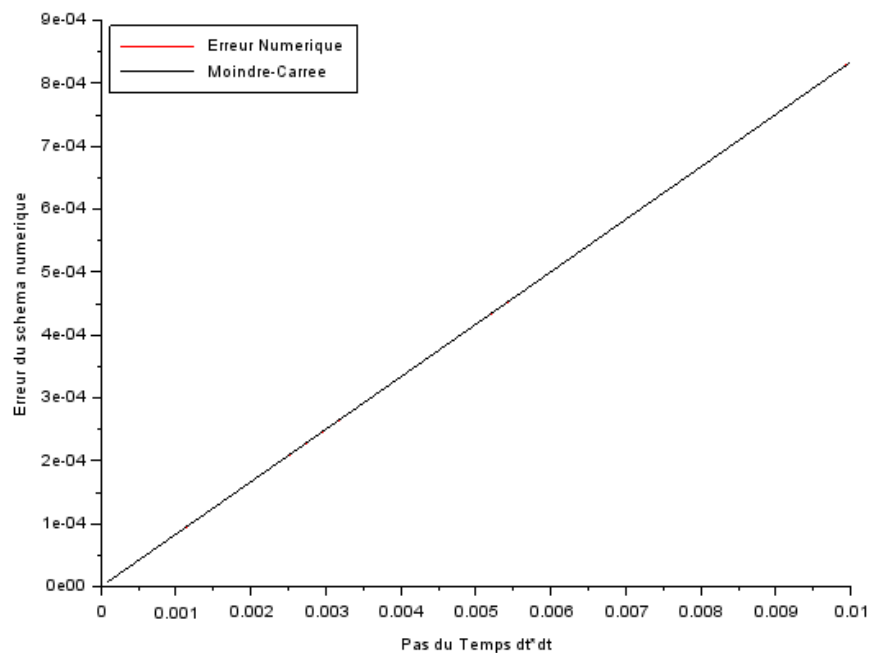


Figure 1.2: Variation des erreurs du schéma numérique en fonction du pas du temps  $\Delta t^2$ .

Le tableau (1.2) regroupe les pentes  $k_i$  des courbes des figures (1.1) et (1.2).

Schéma Numérique	Euler Explicite	Euler Implicite
0.2886849	0.5052744	0.4948559

Table 1.2: Pente des courbes des erreurs en fonction du pas du temps  $\Delta t$ .

### 11/ Étude du problème 2:

On considère le problème suivant:

$$\begin{cases} u'(t) = -u^2(t) = f(t) \\ u(0) = 1 \end{cases} \quad (1.26)$$

#### a) Solution exacte du problème:

D'après le système (1.26), on a:

$$u'(t) = -u^2(t) \Rightarrow \frac{-u'(t)}{u^2(t)} = 1 \Rightarrow \frac{1}{u(t)} = t + cst, \quad cst \in \mathbb{R} \quad (1.27)$$

Ainsi:

$$u(t) = \frac{1}{t + cst}, \quad cst \neq t \quad (1.28)$$

Pour retrouver  $cst$ , on remplace  $t$  dans l'équation (1.28) par  $t = 0$ . Soit:

$$cst = \frac{1}{u(0)} = 1 \quad (1.29)$$

Par conséquent, la solution exacte du problème donné dans l'équation (1.26) est:

$$u(t) = \frac{1}{t + \frac{1}{u(0)}} = \frac{1}{t + 1} \quad (1.30)$$

#### b) Schémas numériques:

On s'intéresse dans la suite aux solutions des schémas d'Euler explicite et implicite, ainsi que le dernier schéma numérique.

Schéma d'Euler explicite:

Le schéma d'Euler explicite est donné par l'équation (1.31).

$$U_{n+1} = U_n + h \times f(U_n) = U_n - h \times U_n^2 \quad (1.31)$$

On en déduit l'expression de  $U_{n+1}$  en fonction de  $U_n$ . Soit:

$$U_{n+1} = U_n \times (1 - \Delta t \times U_n) \quad (1.32)$$

Schéma d'Euler Implicite:

Le schéma d'Euler implicite est donné par l'équation (1.33).

$$\frac{U_{n+1} - U_n}{\Delta t} \approx U_{n+1} = -U_{n+1}'^2 \quad (1.33)$$

D'où:

$$\frac{U_{n+1}}{\Delta t} - \frac{U_n}{\Delta t} = -U_{n+1}^2 \quad (1.34)$$

Soit finalement:

$$U_{n+1} \times \left( \frac{1}{\Delta t} + U_{n+1} \right) = \frac{U_n}{\Delta t} \Rightarrow U_{n+1} \times (1 + U_{n+1} \times \Delta t) = U_n \quad (1.35)$$

L'équation (1.35) est une équation de second degré avec inconnu  $U_{n+1}$ . Elle s'écrit sous la forme:

$$\Delta t \times U_{n+1}^2 + U_{n+1} - U_n = 0 \Rightarrow a \times U_{n+1}^2 + b \times U_{n+1} + c = 0 \quad (1.36)$$

On calcule le discriminant  $\Delta$ . Soit:

$$\Delta = b^2 - 4 a \times c = 1 + 4 \Delta t \times U_n > 0 \quad (1.37)$$

Les solutions  $U_{n+1}$  sont alors:

$$\begin{cases} U_{n+1} = \frac{-1 - \sqrt{1 + 4 \Delta t \times U_n}}{2 \times \Delta t} < 0 \text{ à rejeter} \\ U_{n+1} = \frac{-1 + \sqrt{1 + 4 \Delta t \times U_n}}{2 \times \Delta t} > 0 \text{ notre solution} \end{cases} \quad (1.38)$$

Donc le schéma d'Euler implicite est donné par l'équation (1.38.2).

Schéma Numérique:

On considère le développement de Taylor à l'ordre 1 de  $u(t_m)$ . On aura:

$$u(t_m) = u(t_{m+1} - \Delta t) = u(t_{m+1}) - \Delta t \times u'(t_{m+1}) + O(\Delta t) \quad (1.39)$$

Donc:

$$\frac{u(t_{m+1}) - u(t_m)}{\Delta t} \simeq u'(t_{m+1}) = f(u(t_{m+1})) = -u(t)^2 \quad (1.40)$$

D'après l'équation (1.20) on a:

$$\frac{u(t_{m+1}) - u(t_m)}{\Delta t} = \frac{f(u(t_{m+1})) + f(u(t_m))}{2} \quad (1.41)$$

On peut en déduire donc que:

$$u(t_{m+1}) - \frac{\Delta t}{2} f(u(t_{m+1})) = u(t_m) + \frac{\Delta t}{2} f(u(t_m)), \quad m = 0, 1, \dots \quad (1.42)$$

On a d'après l'équation précédente:

$$u(t_{m+1}) + \frac{\Delta t}{2} (u(t_{m+1}))^2 = u(t_m) - \frac{\Delta t}{2} (u(t_m))^2, \quad m = 0, 1, \dots \quad (1.43)$$

On met les coefficients de  $U_{m+1} + 1$  à une côté et ceux de  $U_m$  à l'autre côté. On obtient:

$$u(t_{m+1}) \times (1 + \frac{\Delta t}{2} u(t_{m+1})) = u(t_m) (1 - \frac{\Delta t}{2} u(t_m)), \quad m = 0, 1, \dots \quad (1.44)$$

Notre équation se ramène a une équation de second degré en  $u_{m+1}$ . Soit donc:

$$\frac{\Delta t}{2} \times u_{m+1}^2 + u_{m+1} + (\frac{\Delta t}{2} \times u_m^2 - u_m) = 0 \quad (1.45)$$

On calcule le discriminant  $\Delta$ . Soit:

$$\Delta = b^2 - 4 a \times c = 1 - 4 \times \frac{\Delta t}{2} \times (\frac{\Delta t}{2} u_m^2 - u_m) = 1 - \Delta t^2 u_m^2 + 2 \times \Delta t \times u_m \quad (1.46)$$

On a  $\Delta$  positive, car le pire scénario est la valeur maximale de  $u_m = 1$ . Dans ce cas, on a  $\Delta = 1 - \Delta t^2 + 2 \times \Delta t > 0$  car  $\Delta t$  est faible  $< 1$ .

Les solutions  $U_{n+1}$  sont alors:

$$\begin{cases} U_{n+1} = \frac{-1 - \sqrt{1 - \Delta t^2 u_m^2 + 2 \times \Delta t \times u_m}}{\Delta t} < 0 \text{ à rejeter} \\ U_{n+1} = \frac{-1 + \sqrt{1 - \Delta t^2 u_m^2 + 2 \times \Delta t \times u_m}}{\Delta t} > 0 \text{ notre solution} \end{cases} \quad (1.47)$$



Donc le schéma numérique est donné par l'équation (1.47.2).

### Précision des trois méthodes:

Les programmes des trois méthodes, ainsi que la solution analytique exacte sont donnés dans listings de (1.6) à (1.9), respectivement.

Listing 1.6: Programme du schéma d'Euler explicite associé au système d'équation (1.26).

```
function [u_explicite]=euler_explicite(t,m)

%Condition initial
u_explicite(1)= 1; %solution a t =0
dt= 1/m; %pas du temps, m=nombre des points

for i = 1:m
    u_explicite(i+1) = u_explicite(i)*(1- dt*u_explicite (i)); %Euler
    explicite
end

endfunction
```

Listing 1.7: Programme du schéma d'Euler implicite associé au système d'équation (1.26).

```
function [u_implicite]=euler_implicite(t,m)

%Condition initial
u_implicite(1)= 1; %solution a t =0
dt= 1/m; %pas du temps
for i = 1:m
    u_implicite(i+1)=(-1+sqrt(1+4*dt*u_implicite(i)))/(2*dt); %Euler
    implicite
end;

endfunction
```

Listing 1.8: Programme du schéma numérique associé au système d'équation (1.26).

```
function [u_numerique]=solution_numerique(t,m)

%Condition initial
u_numerique(1)=1;
dt= 1/m; %pas du temps

for i = 1:m
    u_numerique(i+1) = (-1 + sqrt(1-dt*dt*u_numerique(i)*u_numerique(i)+2*
dt*u_numerique(i)))/dt; %schema numerique
end

endfunction
```

Listing 1.9: Programme de la solution exacte associé au système d'équation (1.26).

```
function [u_exacte]= solution_exacte(t)

    u_exacte = (1/(t+1)); %solution exacte

endfunction
```

Le programme donnant la précision des trois méthodes par rapport à la solution exacte pour  $t = 1$  est donné dans listing (1.10).

Listing 1.10: Programme de la précision des trois méthodes par rapport à la solution exacte.

```
t =1; %temps fixe a t =1
m=20; %nombre de points (varie a chaque fois)
dt= 1/m; %pas du temps

[u_exacte]= solution_exacte(t); %solution exacte
[u_numerique]=solution_numerique(t,m); %solution numerique
[u_explicite]=euler_explicite(t,m); %Euler explicite
[u_implicit]=euler_implicit(t,m); %Euler implicite

% ***Calcul des Erreurs relatives***

erreur_num= norm(u_exacte - u_numerique(m+1))/norm(u_exacte); %precision schema
    numerique
disp(erreur_num); %Erreur numerique

erreur_explicite= norm(u_exacte - u_explicite(m+1))/norm(u_exacte); %precision
    Euler explicite
disp(erreur_explicite); %Erreur explicite

erreur_implicit= norm(u_exacte - u_implicit(m+1))/norm(u_exacte); %precision
    Euler implicite
disp(erreur_implicit); %Erreur implicite

% ***Calcul et comparaison de k***

k_num = sqrt(erreur_num/dt^2); %k pour le schema numerique
disp(k_num);

k_explicite = erreur_explicite/dt; %k pour le schema explicite
disp(k_explicite);

k_implicit = erreur_implicit/dt; %k pour le schema implicite
disp(k_implicit);
```

On se propose dans ce qui suit de comparer la précision des trois méthodes pour un temps  $t = 1$  à partir de listing (1.10). Pour ce faire, on varie le nombre de points  $m$ , d'où la variation du pas du temps  $\Delta t$  (Equation (1.5)).

La variation des erreurs des trois méthodes considérées en fonction du pas du temps  $\Delta t$  et du nombre de points  $m$  associé au système d'équation (1.26) est donné dans le tableau (1.3).

$m$	$\Delta t$	Schéma Numérique	Euler Explicite	Euler Implicite
10	0.1	0.0012537	0.0365742	0.0329878
20	0.05	0.0003127	0.0177902	0.0168979
30	0.0333	0.0001389	0.0117551	0.0113589
40	0.025	0.0000781	0.0087777	0.0085548
50	0.02	0.0000500	0.0070037	0.0068612
60	0.01667	0.0000347	0.0058263	0.0057273
70	0.014286	0.0000255	0.0049878	0.0049150
80	0.0125	0.0000195	0.0043603	0.0043046
90	0.0111	0.0000154	0.0038730	0.0038290
100	0.01	0.0000125	0.0034837	0.0034480

Table 1.3: Variation des erreurs en fonction du nombre de points  $m$  et du pas du temps  $\Delta t$ , système d'équation (1.26).

On trace ensuite la variation des erreurs en fonction du pas du temps  $\Delta t$ . Soit:

- La variation des erreurs des schémas d'Euler implicite et explicite en fonction du pas du temps  $\Delta t$ , comme illustré dans la figure (1.3),
- La variation des erreurs du schéma numérique en fonction du pas du temps  $\Delta t^2$ , comme illustré dans la figure (1.4).

Cherchons maintenant la pente des courbes  $erreur_i = f(\Delta t)$  avec  $i = \{\text{explicite, implicite}\}$  (figure (1.3)) et  $erreur_{numérique} = f(\Delta t^2)$  pour l'erreur numérique (figure (1.24)).

Le tableau (1.2) regroupe les pentes  $k_i$  des courbes des figures (1.3) et (1.4).

Schéma Numérique	Euler Explicite	Euler Implicite
0.3535585	0.3483677	0.344840

Table 1.4: Pente des courbes des erreurs en fonction du pas du temps  $\Delta t$ , système d'équation (1.26).

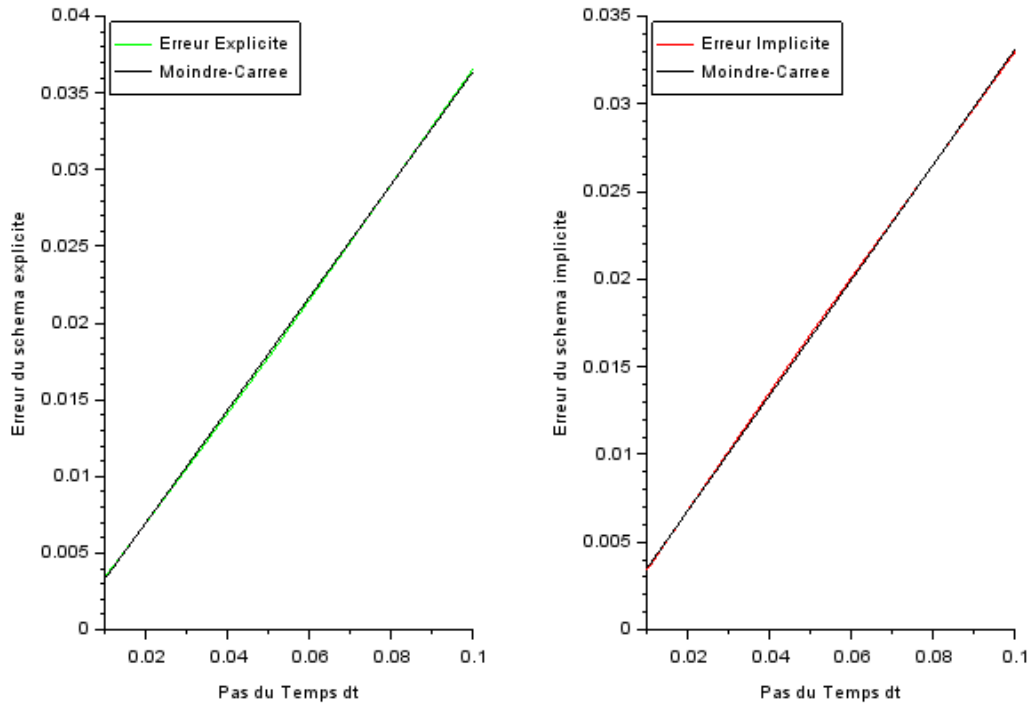


Figure 1.3: Variation des erreurs des schémas d'Euler implicite et explicite en fonction du pas du temps  $\Delta t$ , système d'équation (1.26).

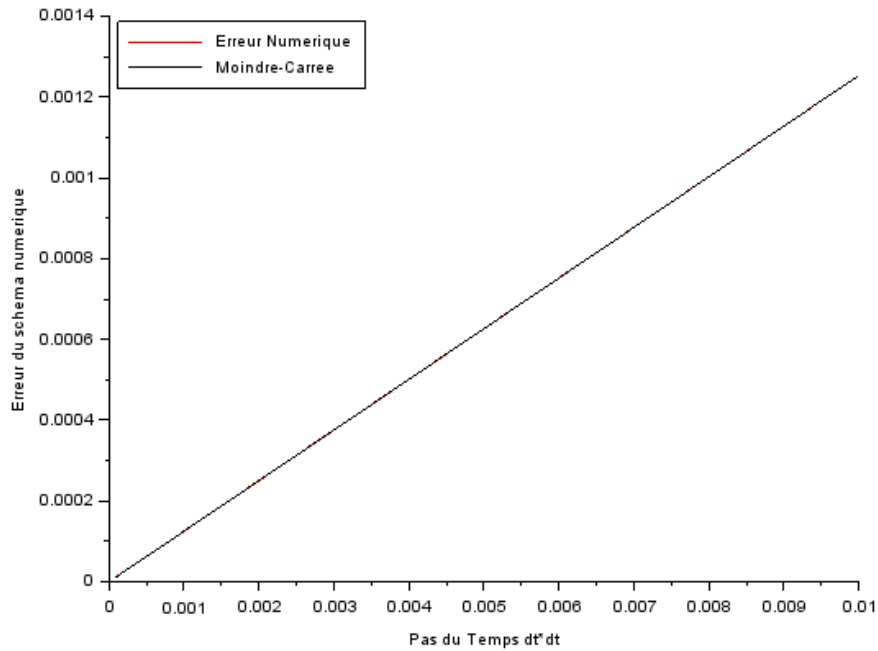


Figure 1.4: Variation des erreurs du schéma numérique en fonction du pas du temps  $\Delta t^2$ , système d'équation (1.26).

**12/ Conclusions générales et comparaison:**

Suite à la comparaison entre les deux problèmes considérés, on se ramène aux conclusions suivantes:

- Un problème numérique évoluant en fonction du temps (avec discrétisation d'un pas du temps  $\Delta t$ ) peut être modélisé de manière implicite ou explicite,
- Le schéma numérique est plus stable numériquement en terme des erreurs de calcul, en fonction de  $\Delta t$ , que les schémas d'Euler implicite et explicite,
- Le schéma d'Euler implicite conduit à des erreurs plus faibles que celles explicite,
- Le schéma d'Euler implicite est inconditionnellement stable, tandis que le schéma d'Euler explicite est conditionnel. D'après "*Différences finies*" 2009, le schéma d'Euler explicite est stable pour  $\Delta t = \frac{1}{m} \leq 1$ , i.e.,  $m \geq 1$  (figure (1.5)),

```
-->m=0
m =

    0.

-->[u_explicite]=euler_explicite(t,m)
u_explicite =

    1.

-->m=1
m =

    1.

-->[u_explicite]=euler_explicite(t,m)
u_explicite =

    1.
    0.

-->m=2
m =

    2.

-->[u_explicite]=euler_explicite(t,m)
u_explicite =

    1.
    0.5
    0.25
```

Figure 1.5: Résultats du schéma d'Euler explicite en fonction du nombre de points  $m$  avec *Scilab*, système d'équations (1.1).

- Les valeurs des pentes  $k_i$  des méthodes considérées dans le problème quadratique sont plus proches que celles considérées dans le problème linéaire.

## Chapter 2

# TP Diffusion de la chaleur

### 2.1 Préliminaire

Soit  $u(x, t)$  un champ scalaire. On se propose dans ce chapitre de résoudre l'équation de chaleur donnée dans l'équation (2.1) (avec un coefficient de conductivité  $D = 1$ ), analytiquement et numériquement. Soit:

$$\left\{ \begin{array}{l} u_t(x, t) = u_{xx}(x, t), \quad x \in ]0, 1[ \text{ et } \forall t > 0 \\ u(0, t) = u(1, t) = 0, \quad \forall t > 0 \\ u(x, 0) = f(x), \quad x \in ]0, 1[ \end{array} \right. \quad (2.1)$$

où  $u(0, t) = u(1, t) = 0$  représente les conditions au bord et  $u(x, 0)$  représente la condition initiale à  $t = 0$ , comme illustré dans la figure (2.1).

Le deuxième chapitre de ce rapport comporte:

- La solution analytique de l'équation (E1),
- La solution numérique de l'équation (E1) en utilisant un schéma explicite,
- La solution numérique de l'équation (E1) en utilisant un schéma implicite,
- La solution numérique de l'équation (E1) en utilisant un schéma de Crank-Nickolson.

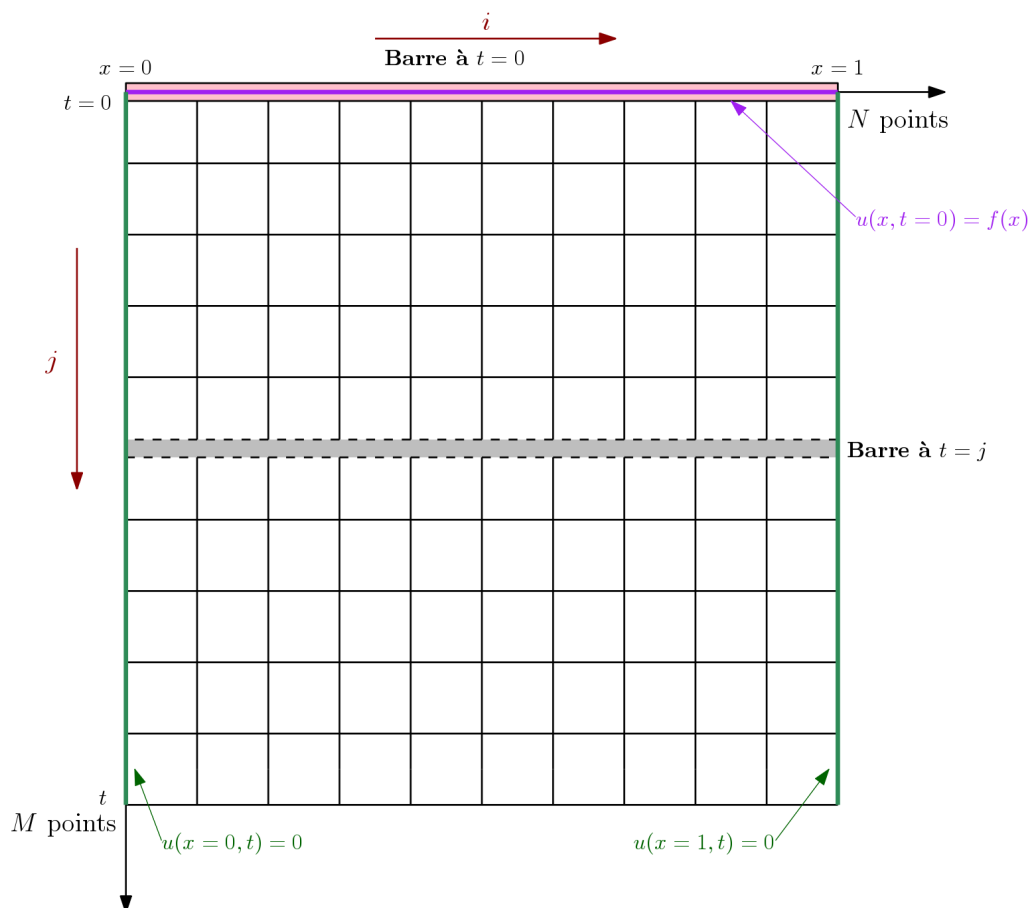


Figure 2.1: Evolution de l'état de la barre  $[0, 1]$  dans le temps (Maillage en temps et en espace).

## 2.2 Solution Analytique

L'objectif de cette section est de résoudre analytiquement l'EDP donnée dans l'équation (2.1).

**a) Résoudre (E1) dans le cas où  $f(x) = \sin(2\pi x)$  pour  $x \in ]0, 1[$  :**

D'après le système (2.1), l'équation à résoudre est:

$$u_t(x, t) = u_{xx}(x, t) \Leftrightarrow \frac{\partial u}{\partial t}(x, t) = \frac{\partial^2 u}{\partial x^2}(x, t) \quad (2.2)$$

En appliquant la méthode de séparation de variables,  $u(x, t)$  s'écrit sous la forme:

$$u(x, t) = X(x).T(t) \quad (2.3)$$

avec les fonctions  $X(x)$  ne dépend que de  $x$  et  $T(t)$  ne dépend que de  $t$ .

On injecte l'équation (2.3) dans l'EDP (2.2). On obtient l'équation (2.5).

$$\begin{cases} \frac{\partial u}{\partial t}(x, t) &= X(x).T'(t) \\ \frac{\partial^2 u}{\partial x^2}(x, t) &= T(t).X''(x) \end{cases} \quad (2.4)$$

$$(2.2) \Leftrightarrow X(x).T'(t) = T(t).X''(x) \quad (2.5)$$

Comme  $\frac{T'(t)}{T(t)}$  ne depend que de  $t$  et  $\frac{X''(x)}{X(x)}$  ne depend que de  $x$ , alors on aura:

$$\frac{T'(t)}{T(t)} = \frac{X''(x)}{X(x)} = -\lambda \text{ (par convention)} \quad (2.6)$$

D'où le système:

$$\begin{cases} T'(t) + \lambda T(t) &= 0 \\ X''(x) + \lambda X(x) &= 0 \end{cases} \quad (2.7)$$

Résolution de l'équation en  $x$ :

On commence par la résolution de l'équation (2.7.2) i.e :

$$X''(x) + \lambda X(x) = 0$$

On tire des conditions limites sur  $x$  :

$$\begin{cases} u(0, t) = 0 \Rightarrow u(0, t) = X(0).T(t) = 0 \\ u(1, t) = 0 \Rightarrow u(1, t) = X(1).T(t) = 0 \end{cases} \quad (2.8)$$

Etudions les différents cas:

- $T(t)=0$  : C'est un cas trivial.
- $X(0)=X(1)=0$ : Pour trouver les solutions possibles, il faut étudier les cas possibles selon le signe de  $\lambda$ .

1. **cas  $\lambda < 0$ :**

L'équation caractéristique de l'EDP est donnée par:

$$r^2 + \lambda = 0 \Rightarrow r^2 = -\lambda \Rightarrow r = \pm\sqrt{-\lambda} \text{ (car } \lambda < 0) \quad (2.9)$$



Dans ce cas, on a des solutions réelles. Soit:

$$X(x) = c_1 \exp(\sqrt{-\lambda}x) + c_2 \exp(-\sqrt{-\lambda}x) \quad (2.10)$$

Pour  $x = 0$ , on a:  $X(0) = c_1 + c_2 \Rightarrow c_2 = -c_1$

Pour  $x = 1$ , on a:  $X(1) = c_1.e^{\sqrt{-\lambda}} + c_2.e^{-\sqrt{-\lambda}} \Rightarrow c_1.e^{\sqrt{-\lambda}} - c_1.e^{-\sqrt{-\lambda}}$

Et comme  $(e^{\sqrt{-\lambda}} - e^{-\sqrt{-\lambda}} \neq 0)$  Donc:  $c_1 = 0$

*Conclusion:*  $X = 0 \Rightarrow u = 0$  c'est une solution nulle. Donc, ce cas ne nous intéresse pas.

## 2. cas $\lambda = 0$ :

On a donc de l'équation (2.7):  $X' = 0$

la solution est donc calculée comme suit:

$$X(x) = c_0 + c_1 \times x \quad (2.11)$$

Pour  $x = 0$ , on a:  $X(0) = c_0 = 0$

Pour  $x = 1$ , on a:  $X(1) = c_0 + c_1 = c_1 = 0 \Rightarrow c_1 = 0$

*conclusion:*  $X = 0 \Rightarrow u = 0$  c'est une solution nulle. Donc, ce cas ne nous intéresse pas.

## 3. cas $\lambda > 0$

L'équation caractéristique de l'EDP est donnée par:

$$r^2 + \lambda = 0 \Rightarrow r^2 = -\lambda = (i\sqrt{\lambda})^2 \Rightarrow r = i\sqrt{\lambda} \quad (2.12)$$

On a donc des solutions complexes. L'expression de la solution est de la forme:

$$X(x) = c_1 \cdot \cos \sqrt{\lambda}x + c_2 \cdot \sin \sqrt{\lambda}x \quad (2.13)$$

On remplace l'équation (2.13) par les conditions initiales. Soit:

$$\begin{cases} X(0) = c_1 + 0 = 0 \Rightarrow c_1 = 0 \\ X(1) = c_2 \cdot \sin \sqrt{\lambda} = 0 \end{cases} \Rightarrow \begin{cases} c_2 = 0 \Rightarrow x = 0 \Rightarrow c_1 = 0 \\ \text{ou} \\ \sin \sqrt{\lambda} = 0 \Rightarrow \sqrt{\lambda} = k.\pi, k \in \mathbb{N}^* \end{cases} \quad (2.14)$$

D'où l'expression de  $\lambda$  suivante:  $\lambda = (k.\pi)^2$

Résolution de l'équation:

On applique la formule de la série de Fourier en sinus. D'après l'équation (2.7.1), on a:

$$T'(t) + \lambda T(t) = 0 \Rightarrow T_k = b_k \times e^{-(k\pi)^2 t} \quad (2.15)$$

En utilisant le théorème de superposition des solutions, on a comme expression générale de la solution :

$$u(x, t) = \sum_{k=1}^{+\infty} \alpha_k \times e^{-(k\pi)^2 t} \times \sin(k\pi x) \quad (2.16)$$

avec  $\alpha_k$  les coefficients de la série de Fourier qu'on va calculer en utilisant la fonction  $f$  dans le cas où:

$$u(x, 0) = f_1(x) = \sin(2\pi x), \quad x \in ]0, 1[ \quad (2.17)$$

**Calcul des coefficients  $\alpha_k$ :**

Les coefficients  $\alpha_k$  se calculent en utilisant l'équation:

$$\alpha_k = 2 \int_0^1 f(x) \times \sin(k\pi x) dx \quad (2.18)$$

avec  $f(x) = \sin(2\pi x)$ ,  $x \in ]0, 1[$  On aura dans ce cas:

$$\alpha_k = \alpha_2 = 2 \int_0^1 \sin(2\pi x) \times \sin(k\pi x) dx \quad (2.19)$$

L'équation devient pour  $k = 2$ :

$$\alpha_2 = 2 \int_0^1 \sin^2(2\pi x) dx, \quad \text{sachant que: } \sin^2 x = \frac{1 - \cos(2x)}{2} \quad (2.20)$$

D'où:

$$\alpha_2 = 2 \int_0^1 \left( \frac{1 - \cos(4\pi x)}{2} \right) dx = \left[ x - \frac{1}{4} \sin(4\pi x) \right]_0^1 = 1 \quad (2.21)$$

*Conclusion:* On injecte l'équation (2.21) dans l'équation (2.16). La solution sera dans le cas où  $f(x) = f_1(x)$ :

$$u(x, t) = \alpha_2 \times e^{-(2\pi)^2 t} \times \sin(2\pi x) \Rightarrow u(x, t) = e^{-(2\pi)^2 t} \times \sin(2\pi x) \quad (2.22)$$

**b) Résoudre (E1) dans le cas où:**

$$f(x) = \begin{cases} 2x & \text{si } x \in ]0, \frac{1}{2}[ \\ 2(1-x) & \text{si } x \in ]\frac{1}{2}, 1[ \end{cases} \quad (2.23)$$

De la même manière que la question (a), on calcul les coefficients  $\alpha_k$  en utilisant l'équation (2.18).

On remplace par la fonction  $f$  qui est continue dans un intervalle compact. Soit:

$$\alpha_k = 2 \int_0^{1/2} 2x \times \sin(k\pi x) dx + 2 \int_{1/2}^1 2(1-x) \times \sin(k\pi x) dx \quad (2.24)$$

On nomme les deux parties  $A$  et  $B$  de l'intégrale comme suit:

$$A = 2 \int_0^{1/2} 2x \times \sin(k\pi x) dx$$

Et

$$B = 2 \int_{1/2}^1 2(1-x) \times \sin(k\pi x) dx$$

Calcul de A:

Pour résoudre  $A$ , on utilise le théorème de l'intégration par parties. On pose:

$$\begin{cases} u &= 2x \\ dv &= \sin(k\pi x) dx \end{cases} \Rightarrow \begin{cases} du &= 2 \\ v &= \frac{-\cos(k\pi x)}{k\pi} \end{cases} \quad (2.25)$$

Rappel du théorème de l'intégration par parties:  $\int u dv = u.v - \int v du$

On applique le théorème de l'intégration par parties sur  $A$ . On aura:

$$A = \left[ \frac{-2x \times \cos(k\pi x)}{k\pi} \right]_0^{1/2} - \int_0^{1/2} \frac{2 \cos(k\pi x)}{k\pi} dx = \frac{\cos(\frac{k\pi}{2})}{k\pi} + \frac{1}{k\pi} \left[ \frac{2 \sin(k\pi x)}{k} \right]_0^{1/2} \quad (2.26)$$

Ainsi le resultat final de  $A$  est:

$$A = \frac{-\cos(\frac{k\pi}{2})}{k\pi} + \frac{2}{(k\pi)^2} \times \sin(\frac{k\pi}{2}) \quad (2.27)$$

Calcul de B:

Pour résoudre  $B$ , on utilise le théorème de l'intégration par parties. On pose:

$$\begin{cases} u &= 2(1-x) \\ dv &= \sin(k\pi x) dx \end{cases} \Rightarrow \begin{cases} du &= -2 \\ v &= \frac{-\cos(k\pi x)}{k\pi} \end{cases} \quad (2.28)$$

De la même manière, on applique l'intégration par parties:

$$B = [2(1-x) \times \frac{-\cos(k\pi x)}{k\pi}]_{1/2}^1 - \int_{1/2}^1 \frac{2\cos(k\pi x)}{k\pi} dx \quad (2.29)$$

Soit:

$$B = \frac{\cos(k\pi/2)}{k\pi} - [2\frac{\sin(k\pi x)}{(k\pi)^2}]_{1/2}^1 = \frac{\cos(k\pi/2)}{k\pi} + 2 \times \frac{\sin(k\pi/2)}{(k\pi)^2} \quad (2.30)$$

De (2.24), on a:  $\alpha_k = A + B$ . On remplace cette équation par les expressions de  $A$  et  $B$ :

$$\alpha_k = \frac{-\cos(k\pi/2)}{k\pi} + \frac{2}{(k\pi)^2} \times \sin(k\pi/2) + \frac{\cos(k\pi/2)}{k\pi} + 2 \times \frac{\sin(k\pi/2)}{(k\pi)^2} \quad (2.31)$$

Soit l'expression finale de  $\alpha_k$ :

$$\alpha_k = \frac{8 \times \sin(k\pi/2)}{(k\pi)^2} \quad (2.32)$$

En remplaçant l'expression générale de la solution dans l'équation (2.16) par la valeur de  $\alpha_k$ , on obtient la solution analytique de l'EDP (2.2) dans le cas où  $f(x) = f_2(x)$ :

$$u(x, t) = \sum_{k=1}^{\infty} \frac{8 \times \sin(k\pi/2)}{(k\pi)^2} \times e^{-(k\pi)^2 t} \sin(k\pi x) \quad (2.33)$$

## 2.3 Solution Numérique: Schéma Explicite:

### a) Schéma explicite d'ordre 1 en temps et 2 en espace:

On pose deux pas de temps et d'espace  $\Delta t$  et  $\Delta x$ . Soit la discrétisation en temps et en espace comme suit:

Partie temporelle: Discrétisation en temps (Formule progressive):

On s'intéresse dans cette section à déterminer un schéma de  $u_t(x, t)$ .

Soit:

$$u_t(x, t_k) = \lim_{\Delta t \rightarrow 0} \frac{u(x, t_{k+1}) - u(x, t_k)}{\Delta t} \approx \frac{u(x, t_{k+1}) - u(x, t_k)}{\Delta t} \quad (2.34)$$

Soit la formule progressive de  $u_t(x, t)$  suivante:

$$\frac{\partial u}{\partial t} = \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} \quad (2.35)$$

Partie spatiale: Discrétisation en espace (Formule centrée):

On s'intéresse dans cette section à déterminer un schéma de  $u_{xx}(x, t)$ .

On a:

$$u_x(x_k, t) = \lim_{\Delta x \rightarrow 0} \frac{u(x_{k+1}, t) - u(x_k, t)}{\Delta x} \approx \frac{u(x_{k+1}, t) - u(x_k, t)}{\Delta x} \quad (2.36)$$

On derive l'équation (2.36) une autre fois. On obtient:

$$u_{xx}(x_k, t) = \lim_{\Delta x \rightarrow 0} \frac{u'(x_{k+1}, t) - u'(x_k, t)}{\Delta x} = \frac{\frac{u(x_{k+1}, t) - u(x_k, t)}{\Delta x} - \frac{u(x_k, t) - u(x_{k-1}, t)}{\Delta x}}{\Delta x} \quad (2.37)$$

Soit la formule progressive de  $u_t(x, t)$  suivante:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x + \Delta x, t) - 2 \times u(x, t) + u(x - \Delta x, t)}{(\Delta x)^2} \quad (2.38)$$

Passage aux indices discrets:

Le passage en indices discrets consiste à remplacer l'espace  $x$  par  $i$  et le temps  $t$  par  $j$ , comme illustré dans la figure (2.1). Ainsi, on remplace l'équation (2.35) par l'équation (2.39) et l'équation (2.38) par l'équation (2.40). Soit:

$$u_t(x, t) = \frac{u_i^{j+1} - u_i^j}{\Delta t} \quad (2.39)$$

Et:

$$u_{xx}(x, t) = \frac{u_{i+1}^j - 2 \times u_i^j + u_{i-1}^j}{(\Delta x)^2} \quad (2.40)$$

En remplaçant les équations (2.39) et (2.40) dans l'EDP (2.2). Alors:

$$u_i^{j+1} = \frac{\Delta t}{(\Delta x)^2} \times (u_{i+1}^j - 2 \times u_i^j + u_{i-1}^j) + u_i^j \quad (2.41)$$

Il s'agit d'une forme explicite car on peut prédire la nouvelle valeur de  $u$  (temps futur) en fonction de l'ancienne valeur (temps présent) d'une manière directe (explicite).

**b) Stabilité du schéma explicite:**

Pour ne pas confondre dans l'analyse de stabilité  $i$  par celle du nombre complexe, on considère l'indice  $j$  pour l'espace et l'indice  $m$  pour le temps.

On considère l'équation (2.41) et on remplace  $u_j^m$  par:

$$u_j^m = (a_k)^m \times e^{ik\pi x_j} \quad (2.42)$$

Soit:

$$\frac{a_k - 1}{\Delta t} = \frac{e^{ik\pi\Delta x} - 2 + e^{-ik\pi\Delta x}}{(\Delta x)^2} \quad (2.43)$$

D'où:

$$a_k = 1 - 4 \times \sin^2\left(\frac{k\pi\Delta x}{2}\right) \times \frac{\Delta t}{(\Delta x)^2} \quad (2.44)$$

La condition de stabilité de Neumann est obtenu pour:

$$\|a_k\| \leq 1 \quad (2.45)$$

On remplace l'équation (2.44) dans l'équation (2.45). On obtient:

$$-1 \leq 1 - 4 \times \sin^2\left(\frac{k\pi\Delta x}{2}\right) \times \frac{\Delta t}{(\Delta x)^2} \leq 1 \quad (2.46)$$

On conclut que la condition de stabilité sera dans ce cas:

$$\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \quad (2.47)$$

**c) Schéma sous forme matricielle:**

On pose dans ce cas:

$$r = \frac{\Delta t}{(\Delta x)^2} \quad (2.48)$$

L'équation (2.41) devient dans ce cas:

$$u_i^{j+1} = r \times u_{i-1}^j + (1 - 2r) \times u_i^j + r \times u_{i+1}^j \quad (2.49)$$

On représente dans ce qui suit cette équation sous forme matricielle. Soit:

$$(u_i)^{j+1} = A \times (u_i)^j \quad (2.50)$$

avec:

$$\begin{cases} (u_i)^{j+1} & : \text{vecteur dans le temps futur} \\ A & : \text{la matrice à déterminer} \\ (u_i)^j & : \text{vecteur dans le temps présent} \end{cases} \quad (2.51)$$

Soit la forme matricielle donnée par le système suivant:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ r & (1-2r) & r & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & r & (1-2r) & r \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \end{bmatrix}}_{(\mathbf{u}_i)^j} = \underbrace{\begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \end{bmatrix}}_{(\mathbf{u}_i)^{j+1}} \quad (2.52)$$

#### d) Implémentation:

Pour une simulation initiale du schéma explicite, on pose:

$$\begin{cases} \Delta t & = 10^{-4} \\ \Delta x & = 10^{-2} \\ i & \in [0, N] \\ j & \in [0, M] \end{cases} \quad (2.53)$$

Les valeurs des pas du temps et d'espace dans (2.53) vérifient bien la condition de stabilité donnée dans l'équation (2.47). Soit:

$$\frac{\Delta t}{(\Delta x)^2} = \frac{10^{-4}}{10^{-2}} = 10^{-2} \leq \frac{1}{2} \quad (2.54)$$

Soit l'algorithme du schéma explicite donné dans listing (2.1).

Listing 2.1: Schéma Explicite.

```

function [u]=FTCS(N,M)
    %Donnees de x
    Dx= 1/N; %pas en x

    %Donnees de t
    Tob = 1; %temps d' observation
    Dt = 1/M; %pas en t

    r=Dt/(Dx)^2;

    %Conditions aux bord
    u(1) = 0; %temperature fixe pour x = 0
    u(N) = 0; %temperature fixe pour x = N

    %Condition initiale: Temperature de toute la barre a t = 0
    for i = 2:(N-1)
        [f]= func(i*Dt); %fonction definit dans un autre script
        u(i) = f;
    end

    %Expression de u en temps et en espace
    for j = 2:M %En temps
        for i=2:(N-1) %en espace
            u(i)=(1-2*r)*u(i)+r*(u(i+1)+u(i-1));
        end
    end
endfunction

```

avec  $f_i$  conditions aux bords. Soit les deux expressions des fonctions  $f_1$  et  $f_2$  donnés dans listings (2.2) et (2.3).

Listing 2.2: Expression de la fonction  $f_1$ .

```

function [f]= func(x)
    f = sin(2*\%pi*x); %Expression de la fonction pour t = 0
endfunction

```

Listing 2.3: Expression de la fonction  $f_2$ .

```

function [f]= func1(x)

    if ((x > 0) && (x <= 1/2)) then
        f = 2*x; %Expression de la fonction pour x in ]0,1/2]
    end

    if ((x > 1/2) && (x < 1)) then
        f = 2*(1-x); %Expression de la fonction pour x in ]1/2,1[
    end

endfunction

```



La simulation est ainsi donnée dans listing (2.4), en variant le nombre de points  $N$  et  $M$ , pour varier le pas d'espace  $\Delta x$  et de temps  $\Delta t$ .

Listing 2.4: Simulation en utilisant le schéma explicite.

```
N = 100; %Nombre de points selon x
M = 10^(4); %Nombre de points selon y

[u]=FTCS(N,M); %resultat en utilisant le schema explicite
disp(u) %resultat u obtenu avec le schema explicite
```

Variation dans le cas de  $f = f_1$  (Equation (2.17)):

La variation de  $u(x, t)$  en fonction de  $x$  est illustrée dans la figure (2.2), pour la fonction  $f_1(x)$  considérée.

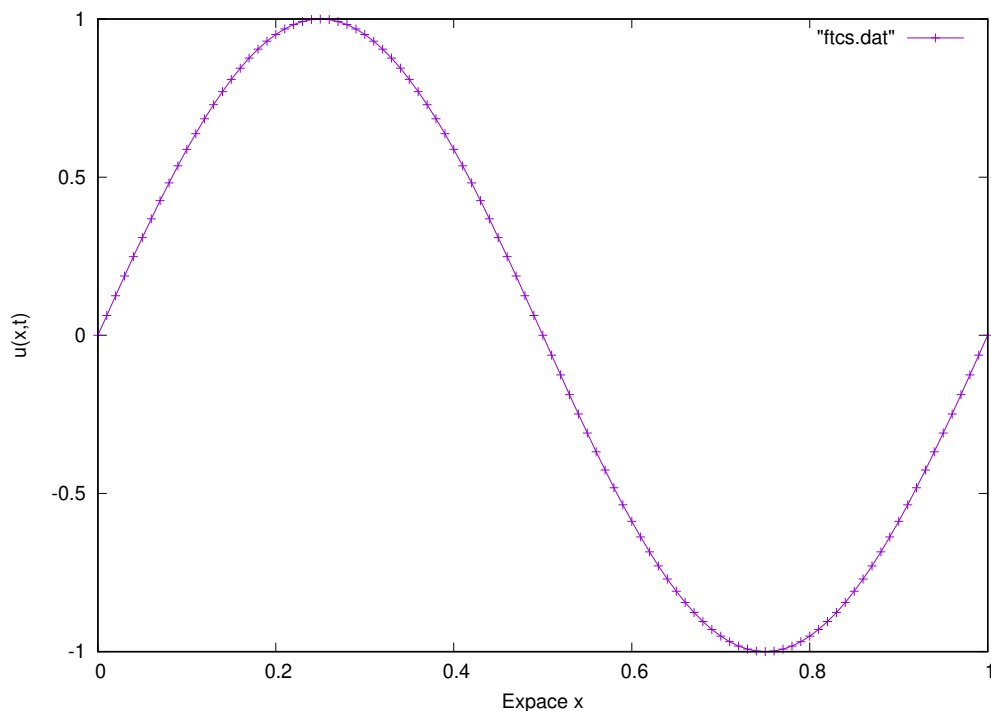


Figure 2.2: Variation de  $u(x, t)$  en fonction de  $x$  dans le cas du schéma explicite,  $f = f_1$  et  $t$  fixe.

D'après la figure (2.2), on remarque que l'allure de la courbe suit la fonction  $\sin(2\pi x)$ . Il y a ainsi deux points importants:

- Un **maximum**  $u = 1$  atteint dans le cas où  $x = \frac{1}{4}$ ,
- Un **minimum**  $u = -1$  atteint dans le cas où  $x = \frac{3}{4}$ .

La figure (2.3) montre la variation de la fonction scalaire  $u(x, t)$  en fonction du temps  $t$  et d'espace  $x$ . Ainsi, on obtient le même comportement observé dans la figure (2.2) pour des valeurs de temps différentes.

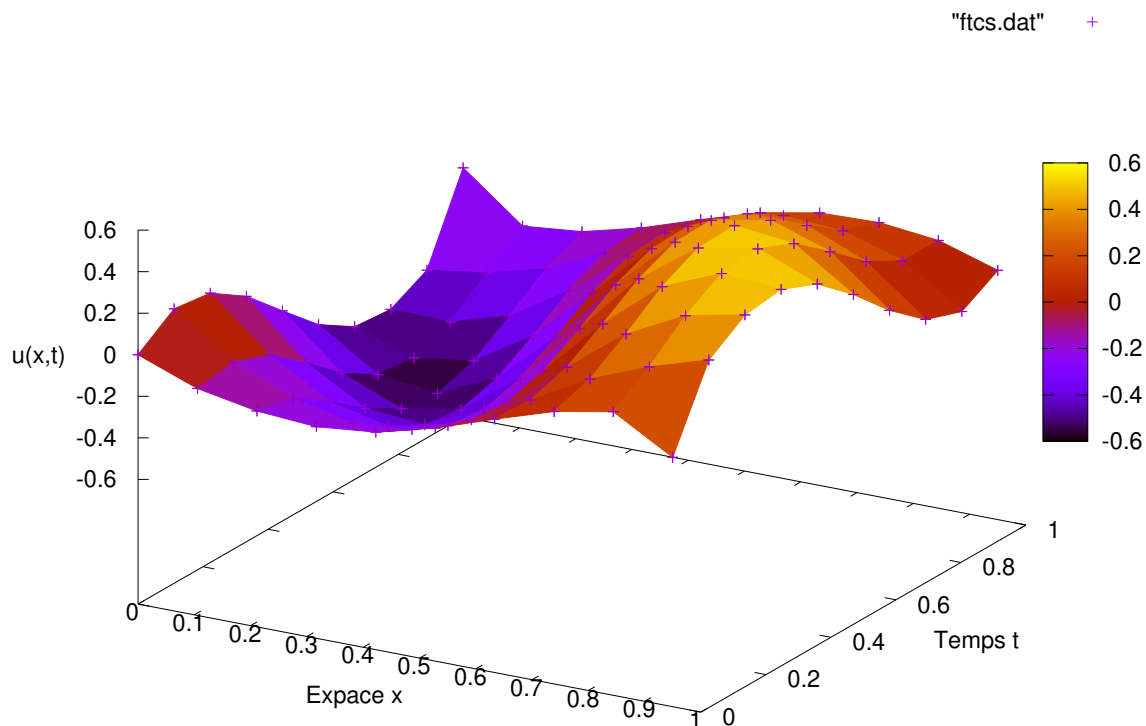


Figure 2.3: Variation de  $u(x, t)$  en fonction d'espace  $x$  et du temps  $t$  dans le cas du schéma explicite,  $f = f_1$ .

Variation dans le cas de  $f = f_2$  (Equation (2.23)):

On se propose dans cette partie de tracer les 2 graphes suivants:

- La variation de  $u(x, t)$  en fonction de  $x$ : Figure (2.4),
- La variation de  $u(x, t)$  en fonction de  $x$  et de  $t$ : Figure (2.5).

Comme la fonction est répartie sur 2 intervalles, elle est alors:

- Croissante sur  $[0, \frac{1}{2}]$ ,
- Décroissante sur  $[\frac{1}{2}, 1]$ .

Elle a comme maximum  $u(x = \frac{1}{2}, t) = 1$  et comme minimum  $u(x = 0, t) = u(x = 1, t) = 0$ . En variant le temps, on obtient également cette allure répartie sur l'intervalle  $[0, T_{obs} = 1]$ .

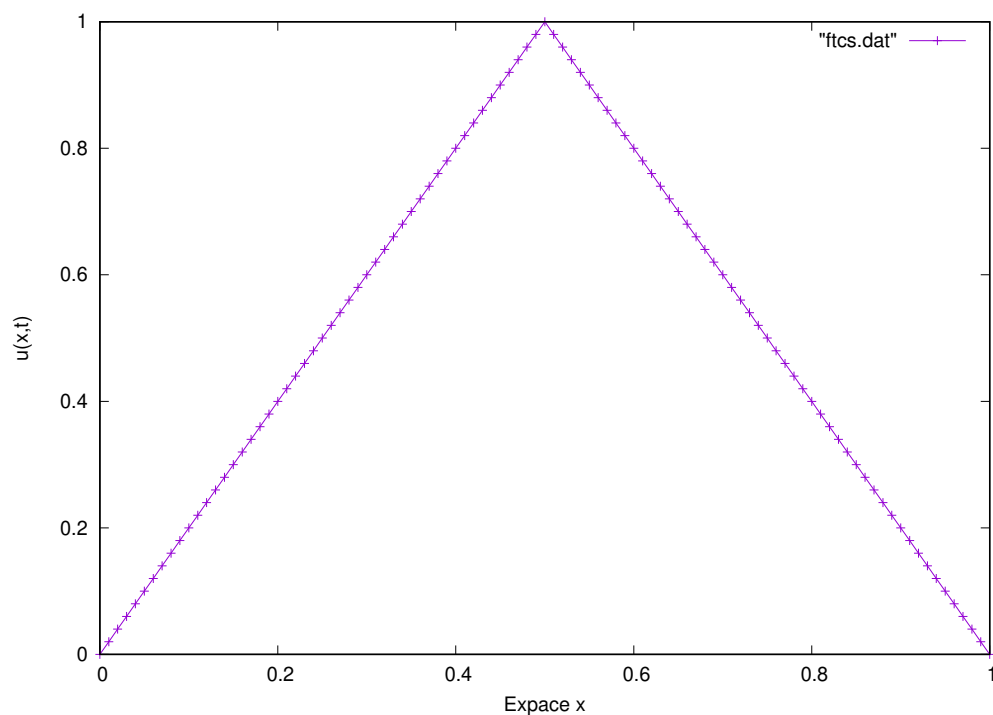


Figure 2.4: Variation de  $u(x,t)$  en fonction de  $x$  dans le cas du schéma explicite,  $f = f_2$  et  $t$  fixe.

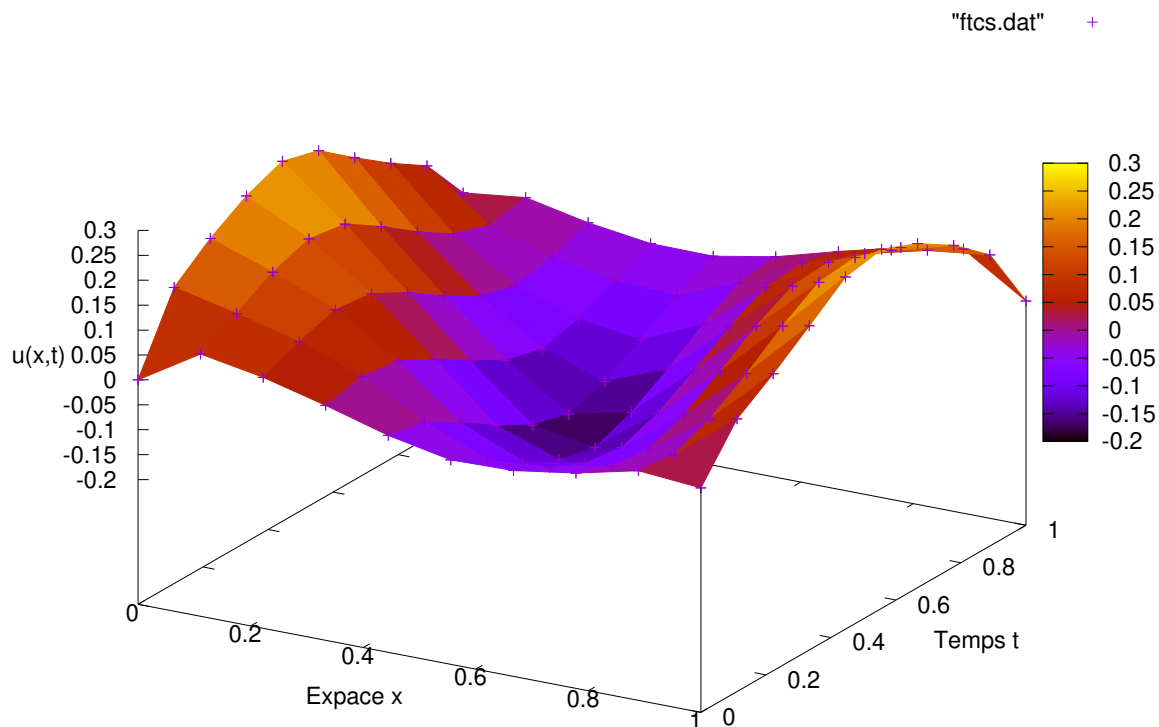


Figure 2.5: Variation de  $u(x,t)$  en fonction de  $x$  et  $t$  dans le cas du schéma explicite,  $f = f_2$ .

## 2.4 Solution Numérique: Schéma Implicite:

### a) Schéma implicite d'ordre 1 en temps et 2 en espace:

On pose deux pas de temps et d'espace  $\Delta t$  et  $\Delta x$ . Le schéma implicite est déduit dans ce cas du schéma explicite, en remplaçant l'instant  $j$  dans l'équation (2.40) par l'instant  $j + 1$ .

Soit l'expression donnée dans l'équation (2.55) qui représente le schéma implicite en question:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{u_{i+1}^{j+1} - 2 \times u_i^{j+1} + u_{i-1}^{j+1}}{(\Delta x)^2} \quad (2.55)$$

Il s'agit d'un schéma implicite, car la résolution de  $u_i^j$  n'est pas directe et implicite. Soit donc:

$$u_i^j = -\frac{\Delta t}{(\Delta x)^2} u_{i+1}^{j+1} + (1 + \frac{2\Delta t}{(\Delta x)^2}) u_i^{j+1} - \frac{\Delta t}{(\Delta x)^2} u_{i-1}^{j+1} \quad (2.56)$$

En posant  $r = \frac{\Delta t}{(\Delta x)^2}$ , on obtient:

$$u_i^j = -r \times u_{i+1}^{j+1} + (1 + 2 \times r) u_i^{j+1} - r \times u_{i-1}^{j+1} \quad (2.57)$$

### b) Stabilité du schéma implicite:

Pour ne pas confondre dans l'analyse de stabilité  $i$  par celle du nombre complexe, on considère l'indice  $j$  pour l'espace et l'indice  $m$  pour le temps.

On considère l'équation (2.55) et on remplace  $u_j^m$  par son expression dans l'équation (2.42).

Soit:

$$\frac{a_k - 1}{\Delta t} = a_k \times \frac{e^{ik\pi\Delta x} - 2 + e^{-ik\pi\Delta x}}{(\Delta x)^2} \quad (2.58)$$

Comme:

$$e^{ik\pi\Delta x} - 2 + e^{-ik\pi\Delta x} = (e^{\frac{ik\pi\Delta x}{2}} - e^{-ik\pi\Delta x})^2 = (2i \times \sin(\frac{k\pi\Delta x}{2}))^2 \quad (2.59)$$

Alors, l'équation (2.58) devient:

$$\frac{a_k - 1}{\Delta t} = a_k \times \frac{(-4 \times \sin^2(\frac{k\pi\Delta x}{2}))}{(\Delta x)^2} \quad (2.60)$$

Tout développement fait, on en déduit l'expression de  $a_k$  suivante:

$$a_k = \frac{1}{1 + 4 \times \sin^2(\frac{k\pi\Delta x}{2}) \times \frac{\Delta t}{(\Delta x)^2}} \quad (2.61)$$

D'après la condition de stabilité dans l'équation (2.45), on a:

$$\|a_k\| = \left\| \frac{1}{1 + 4 \times \sin^2\left(\frac{k\pi\Delta x}{2}\right) \times \frac{\Delta t}{(\Delta x)^2}} \right\| \leq 1 \quad (2.62)$$

Or on a:

$$\begin{cases} \frac{\Delta t}{(\Delta x)^2} > 0 \\ \sin^2\left(\frac{k\pi\Delta x}{2}\right) > 0 \\ \Rightarrow 1 + 4 \times \sin^2\left(\frac{k\pi\Delta x}{2}\right) \times \frac{\Delta t}{(\Delta x)^2} > 0 \end{cases} \quad (2.63)$$

Ainsi, l'inégalité dans l'équation (2.62) est toujours vérifiée. On conclut que le schéma implicite est inconditionnellement stable.

c) **Schéma sous forme matricielle:**

On représente dans ce qui suit l'équation (2.57) sous forme matricielle. Soit:

$$(u_i)^j = A \times (u_i)^{j+1} \quad (2.64)$$

avec:

$$\begin{cases} (u_i)^{j+1} & : \text{vecteur dans le temps futur} \\ A & : \text{la matrice à déterminer} \\ (u_i)^j & : \text{vecteur dans le temps présent} \end{cases} \quad (2.65)$$

Soit la forme matricielle donnée par le système suivant:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -r & (1+2r) & -r & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -r & (1+2r) & -r \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \end{bmatrix}}_{(\mathbf{u}_i)^{j+1}} = \underbrace{\begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \end{bmatrix}}_{(\mathbf{u}_i)^j} \quad (2.66)$$

On en déduit l'équation suivante:

$$(u_i)^{j+1} = A^{-1} \times (u_i)^j \quad (2.67)$$

d) **Implémentation:**

A titre de comparaison, on reprend les valeurs du pas du temps et d'espace définis dans l'équation (2.53). Soit l'algorithme du schéma explicite donné dans listing (2.5).

Listing 2.5: Schéma Implicite.

```
function [u]=implicite(N,M)
    %[M+1, N+1]=size(A);
    A=zeros(N,N);
    u=zeros(N,1);

    %Donnees de x
    Dx= 1/N; %pas en x

    %Donnees de t
    Tobs = 1; %temps d' observation
    Dt = 1/M; %pas en t

    r=Dt/(Dx)^2;

    %Conditions aux bord
    u(1) = 0; %temperature fixe pour x = 0
    u(N) = 0; %temperature fixe pour x = N

    %Condition initiale: Temperature de toute la barre a t = 0
    for i = 2:(N-1)
        [f]= func(i*Dt); //fonction definit dans un autre script
        u(i) = f;
    end

    %Matrice A
    A = diag((1+2*r)*ones(1,N)) + diag(-r * ones (1,N-1),1) + +diag(-r * ones
    (1,N-1),-1);

    %Expression de u en temps et en espace
    for j = 1:M %Boucle de temps
        u=A\u; %solve u
    end
endfunction
```

Les expressions des fonctions  $f_1$  et  $f_2$  sont données dans listings (2.2) et (2.3), respectivement.

La simulation est ainsi donnée dans listing (2.6), en variant le nombre de points  $N$  et  $M$ , pour varier le pas d'espace  $\Delta x$  et de temps  $\Delta t$  (on se place dans les mêmes conditions que précédemment).

Listing 2.6: Simulation en utilisant le schéma implicite.

```
N = 100; %Nombre de points selon x
M = 10^(4); %Nombre de points selon y

[u]=implicite(N,M); %resultat en utilisant le schema implicite
disp(u) %resultat u obtenu avec le schema implicite
```

Variation dans le cas de  $f = f_1$  (Equation (2.17)):

La variation de  $u(x, t)$  en fonction de  $x$  est illustrée dans la figure (2.6), pour la fonction  $f_1(x)$  considérée.

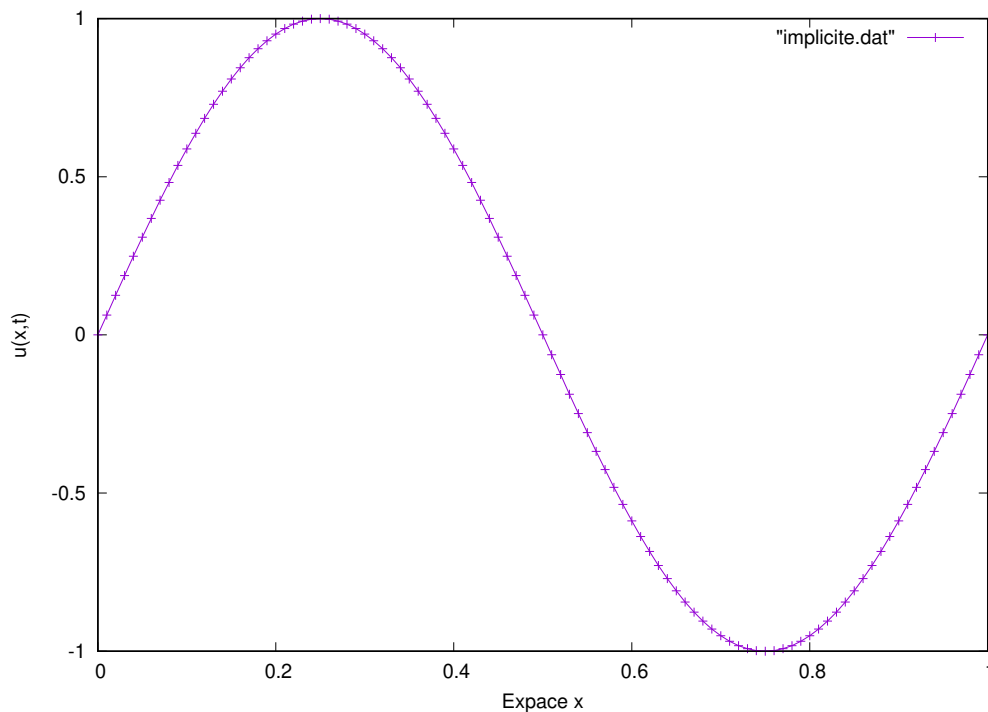


Figure 2.6: Variation de  $u(x, t)$  en fonction de  $x$  dans le cas du schéma implicite,  $f = f_1$  et  $t$  fixe.

D'après la figure (2.6), on remarque que l'allure de la courbe suit la fonction  $\sin(2\pi x)$ . Il y a ainsi deux points importants:

- Un **maximum**  $u = 1$  atteint dans le cas ou  $x = \frac{1}{4}$ ,
- Un **minimum**  $u = -1$  atteint dans le cas ou  $x = \frac{3}{4}$ .

La figure (2.7) montre la variation de la fonction scalaire  $u(x, t)$  en fonction du temps  $t$  et d'espace  $x$  dans le cas du schéma implicite. Ainsi, on obtient le même comportement observé dans la figure (2.6) pour des valeurs de temps différentes.

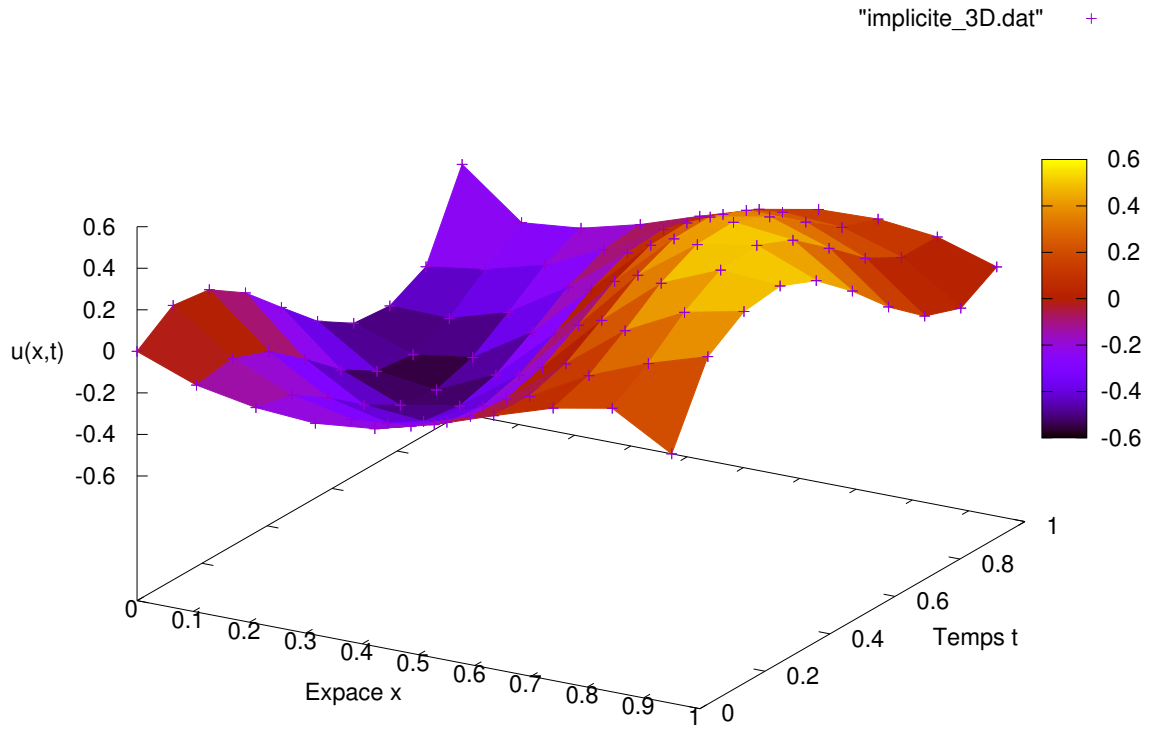


Figure 2.7: Variation de  $u(x,t)$  en fonction d'espace  $x$  et du temps  $t$  dans le cas du schéma implicite,  $f = f_1$ .

Variation dans le cas de  $f = f_2$  (Equation (2.23):

On se propose dans cette partie de tracer les 2 graphes suivants:

- La variation de  $u(x,t)$  en fonction de  $x$ : Figure (2.8),
- La variation de  $u(x,t)$  en fonction de  $x$  et de  $t$ : Figure (2.9).

Comme la fonction est répartie sur 2 intervalles, elle est alors:

- Croissante sur  $[0, \frac{1}{2}]$ ,
- Décroissante sur  $[\frac{1}{2}, 1]$ .

Elle a comme maximum  $u(x = \frac{1}{2}, t) = 1$  et comme minimum  $u(x = 0, t) = u(x = 1, t) = 0$ . En variant le temps, on obtient également cette allure répartie sur l'intervalle  $[0, T_{obs} = 1]$ .



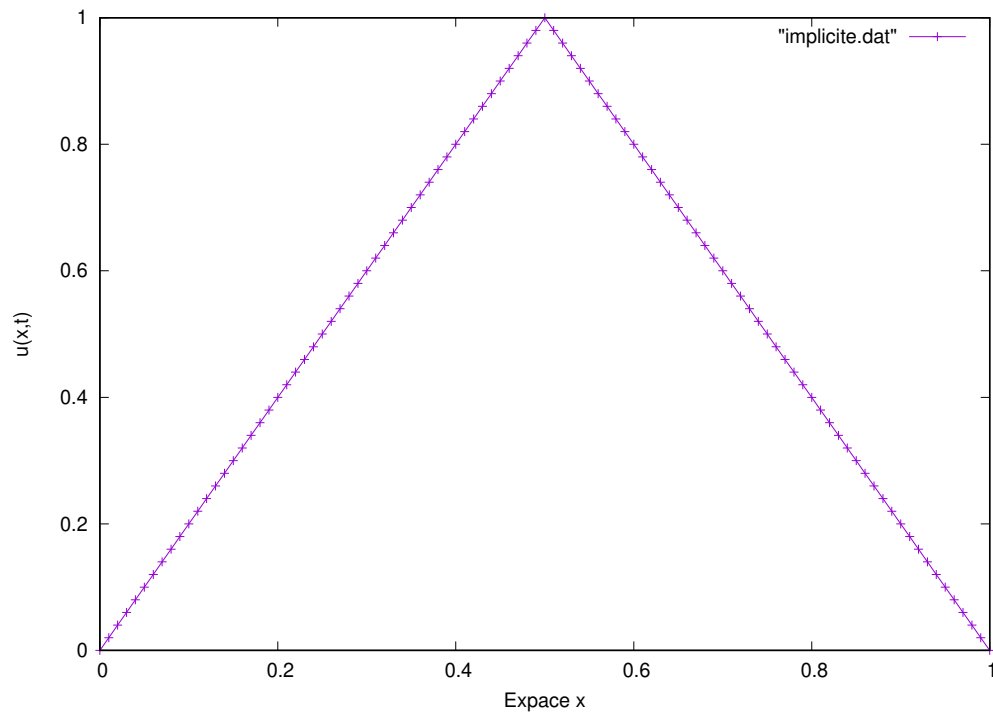


Figure 2.8: Variation de  $u(x,t)$  en fonction de  $x$  dans le cas du schéma implicite,  $f = f_2$  et  $t$  fixe.

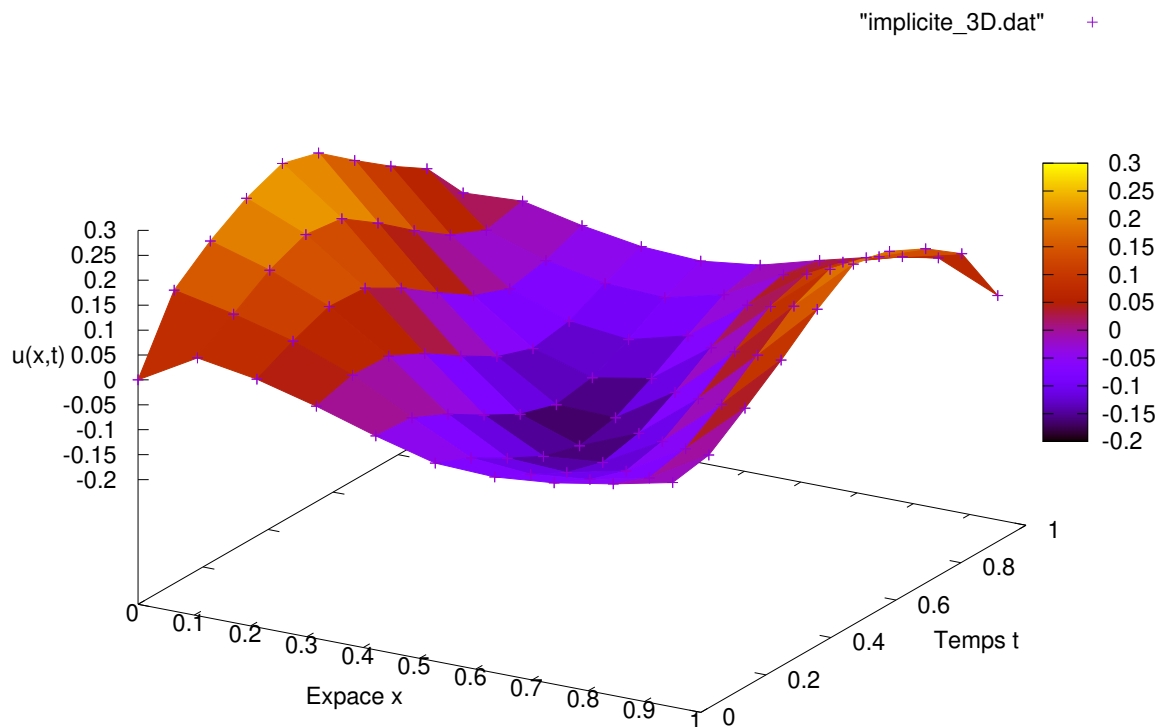


Figure 2.9: Variation de  $u(x,t)$  en fonction de  $x$  et  $t$  dans le cas du schéma implicite,  $f = f_2$ .

## 2.5 Solution Numérique: Schéma de Crank-Nicholson:

### a) Dédution du schéma de Crank-Nicholson:

Le schéma de Crank-Nicholson est déduit en faisant la moyenne du schéma implicite et explicite.

En faisant la moyenne des deux schémas implicite et explicite à partir des équations (2.39) =(2.40) et (2.55), on obtient:

$$\frac{u_i^{j+1} - u_i^j}{2\Delta t} = \frac{1}{2} \times \left( \frac{u_{i-1}^j - 2 \times u_i^j + u_{i+1}^j}{(\Delta x)^2} + \frac{u_{i-1}^{j+1} - 2 \times u_i^{j+1} + u_{i+1}^{j+1}}{(\Delta x)^2} \right) \quad (2.68)$$

Dans ce qui suit, on se propose de faire la démonstration de l'équation (2.68).

#### Partie temporelle:

En faisant le développement de Taylor, le dérivé de  $u$  par rapport au temps s'écrit comme suit:

$$u_t(x, t_k) \approx \frac{u(x, t_{k+1}) - u(x, t_{k-1})}{2 \times \Delta t} \quad (2.69)$$

On a donc deux points de temps  $t_{k+1}$  et  $t_{k-1}$ . Par passage aux indices discrets, on obtient:

$$u_t(x, t) \approx \frac{u_i^{j+1} - u_i^{j-1}}{2 \times \Delta t} \quad (2.70)$$

#### Partie Spatiale:

Dans cette partie, on se propose d'exprimer  $\frac{\partial^2 u(x, t)}{\partial x^2}$  en fonction de  $\frac{\partial^2 u(x, t - \Delta t)}{\partial x^2}$  et  $\frac{\partial^2 u(x, t + \Delta t)}{\partial x^2}$ .

Faisant le développement de Taylor, on obtient:

$$\begin{cases} u_{xx}(x, t + \Delta t) &= u_{xx}(x, t) + \Delta t \times \frac{\partial}{\partial t}(u_{xx}(x, t)) + O((\Delta t)^2) \\ u_{xx}(x, t - \Delta t) &= u_{xx}(x, t) - \Delta t \times \frac{\partial}{\partial t}(u_{xx}(x, t)) + O((\Delta t)^2) \end{cases} \quad (2.71)$$

En sommant les deux équations dans le système (2.71) et on néglige le  $O((\Delta t)^2)$ , on obtient le schéma de Crank-Nicholson décrit par l'équation (2.68). Il s'agit d'un schéma implicite, car on estime exprimer le temps  $(j + 1)$  en fonction de  $j$ . En posant  $2 \times \Delta t = \Delta t'$ , le schéma de Crank-Nicholson se ramène à l'équation:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t'} = \frac{1}{2} \times \left( \frac{u_{i-1}^j - 2 \times u_i^j + u_{i+1}^j}{(\Delta x)^2} + \frac{u_{i-1}^{j+1} - 2 \times u_i^{j+1} + u_{i+1}^{j+1}}{(\Delta x)^2} \right) \quad (2.72)$$

**b) Ordre du schéma de Crank-Nicholson:**

Le schéma de Crank-Nicholson est d'ordre 2 en temps et d'ordre 2 en espace. En effet, d'après le développement du Taylor, on a:

$$\begin{cases} u(x, t_{k+1}) &= u(x, t_k) + \Delta t \times u'(x, t_k) + \frac{\Delta t^2}{2} u''(x, t_k) + r_1(\Delta t) \times \frac{\Delta t^3}{3!} \\ u(x, t_{k-1}) &= u(x, t_k) - \Delta t \times u'(x, t_k) + \frac{\Delta t^2}{2} u''(x, t_k) - \frac{\Delta t^3}{3!} u^{(3)}(x, t_k) + r_2(\Delta t) \times \frac{\Delta t^4}{4!} \end{cases} \quad (2.73)$$

On soustrait (2.73.2) de (2.73.1), on obtient:

$$\frac{u(x, t_{k+1}) - u(x, t_{k-1})}{2 \times \Delta t} - u'(x, t_k) = \frac{\Delta t^2}{3!} u^{(3)}(x, t_k) + r(\Delta t) \times \frac{\Delta t^4}{4!} \quad (2.74)$$

Ainsi:

$$\left| \frac{u(x, t_{k+1}) - u(x, t_{k-1})}{2 \times \Delta t} - u'(x, t_k) \right| = \Delta t^2 \left| \frac{1}{3!} u^{(3)}(x, t_k) + r(\Delta t) \times \frac{\Delta t^2}{4!} \right| \quad (2.75)$$

Comme il s'agit d'une fonction régulière sur un compact, donc bornée. On en déduit que:

$$\left| \frac{u(x, t_{k+1}) - u(x, t_{k-1})}{2 \times \Delta t} - u'(x, t_k) \right| \leq \Delta t^2 \times M \quad (2.76)$$

On conclut que le schéma de Crank-Nicholson est d'ordre 2 en temps. Il est aussi d'ordre 2 en espace, car il s'agit de la moyenne de deux schémas (implicite et explicite) d'ordre 2 en espace.

**c) Stabilité de schéma de Crank-Nicholson:**

Pour ne pas confondre dans l'analyse de stabilité  $i$  par celle du nombre complexe, on considère l'indice  $j$  pour l'espace et l'indice  $m$  pour le temps.

On considère l'équation (2.68) et on remplace  $u_j^m$  par son expression dans l'équation (2.42).

Soit:

$$\frac{a_k - 1}{2 \Delta t} = \frac{1}{2} \times \left( \frac{e^{ik\pi\Delta x} - 2 + e^{-ik\pi\Delta x}}{(\Delta x)^2} + a_k \times \frac{e^{ik\pi\Delta x} - 2 + e^{-ik\pi\Delta x}}{(\Delta x)^2} \right) \quad (2.77)$$

Tout développement fait, on obtient l'expression  $a_k$  suivante:

$$a_k = \frac{1 - 4 \times \sin^2\left(\frac{k\pi\Delta x}{2}\right)}{1 + 4 \times \sin^2\left(\frac{k\pi\Delta x}{2}\right)} \quad (2.78)$$

La condition de stabilité de Von-Neumann implique que:

$$\|a_k\| = \left\| \frac{1 - 4 \times \sin^2\left(\frac{k\pi\Delta x}{2}\right)}{1 + 4 \times \sin^2\left(\frac{k\pi\Delta x}{2}\right)} \right\| \leq 1 \quad (2.79)$$

Cette inégalité est toujours vérifiée car  $(4 \times \sin^2(\frac{k\pi\Delta x}{2}))$  est un entier positif. D'où  $(1 + 4 \times \sin^2(\frac{k\pi\Delta x}{2})) \geq 1 - 4 \times \sin^2(\frac{k\pi\Delta x}{2})$ . On en déduit que le schéma de Crank-Nicholson est inconditionnellement stable.

d) **Schéma de Crank-Nicholson sous forme matricielle:**

On reprend l'équation (2.72) et on regroupe les termes au temps  $j + 1$  d'un côté et les termes en  $j$  de l'autre côté. On obtient:

$$\frac{-1}{\Delta x^2} \times u_{i-1}^{j+1} + \left(\frac{1}{\Delta t} + \frac{2}{\Delta x^2}\right) \times u_i^{j+1} - \frac{1}{\Delta x^2} \times u_{i+1}^{j+1} = \frac{1}{\Delta x^2} \times u_{i-1}^j + \left(\frac{1}{\Delta t} - \frac{2}{\Delta x^2}\right) \times u_i^j + \frac{1}{\Delta x^2} \times u_{i+1}^j \quad (2.80)$$

On multiplie l'équation (2.80) par  $\Delta t$  pour faire apparaître la variable  $r = \frac{\Delta t}{\Delta x^2}$ .

$$-r \times u_{i-1}^{j+1} + (1 + 2 \times r) \times u_i^{j+1} - r \times u_{i+1}^{j+1} = r \times u_{i-1}^j + (1 - 2 \times r) \times u_i^j + r \times u_{i+1}^j \quad (2.81)$$

Soit sous forme matricielle:

$$A_1 \times (u_i)^{j+1} = A_2 \times (u_i)^j \quad (2.82)$$

avec:

$$\begin{cases} (u_i)^{j+1} & : \text{vecteur dans le temps futur} \\ A_1 \text{ et } A_2 & : \text{les matrices à déterminer} \\ (u_i)^j & : \text{vecteur dans le temps présent} \end{cases} \quad (2.83)$$

Soit la forme matricielle donnée par le système suivant:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -r & (1+2r) & -r & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -r & (1+2r) & -r \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_1} \underbrace{\begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \end{bmatrix}}_{(\mathbf{u}_i)^{j+1}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ r & -(1+2r) & r & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & r & -(1+2r) & r \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_2} \underbrace{\begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \end{bmatrix}}_{(\mathbf{u}_i)^j} \quad (2.84)$$

d) **Implémentation:**

A titre de comparaison, on reprend les valeurs du pas du temps et d'espace définis dans l'équation (2.53). Soit l'algorithme du schéma explicite donné dans listing (2.7).

Listing 2.7: Schéma de Crank-Nicholson.

```
function [u]=Crank_Nicholson(N,M)

    A1=zeros(N,N); %initialisation du matrice A1
    A2=zeros(N,N); %initialisation du matrice A2
    u=zeros(N,1); %initialisation du vecteur u

    %Donnees de x
    Dx= 1/N; %pas en x

    %Donnees de t
    Tob = 1; %temps d' observation
    Dt = 1/M; %pas en t

    r=Dt/(Dx)^2; %variable r

    %Conditions aux bord
    u(1) = 0; %temperature fixe pour x = 0
    u(N) = 0; %temperature fixe pour x = N

    %Condition initiale: Temperature de toute la barre a t = 0
    for i = 2:(N-1)
        [f]= func(i*Dt); %fonction definit dans un autre script
        u(i) = f;
    end

    %Matrices A1 et A2
    A1 = diag((1+2*r)*ones(1,N)) + diag(-r * ones (1,N-1),1) + +diag(-r * ones
(1,N-1),-1);
    A2 = diag((-1-2*r)*ones(1,N)) + diag(r * ones (1,N-1),1) + +diag(r * ones
(1,N-1),-1);

    %Expression de u en temps et en espace
    for j = 1:M %En temps
        u=A1\ (A2*u); %resolution du systeme (2.84)
    end
endfunction
```

Les expressions des fonctions  $f_1$  et  $f_2$  sont données dans listings (2.2) et (2.3), respectivement.

La simulation est ainsi donnée dans listing (2.8), en variant le nombre de points  $N$  et  $M$ , pour varier le pas d'espace  $\Delta x$  et de temps  $\Delta t$  (on se place dans les mêmes conditions que précédemment).

Listing 2.8: Simulation en utilisant le schéma de Crank-Nicholson.

```

N = 100; %Nombre de points selon x
M = 10^(4); %Nombre de points selon y

[u]=Crank_Nicholson(N,M); %resultat en utilisant le schema de Crank-Nicholson
disp(u) %resultat u obtenu avec le schema de Crank-Nicholson

```

Variation dans le cas de  $f = f_1$  (Equation (2.17)):

La variation de  $u(x, t)$  en fonction de  $x$  dans le cas du schéma de Crank-Nicholson est illustrée dans la figure (2.10), pour la fonction  $f_1(x)$  considérée.

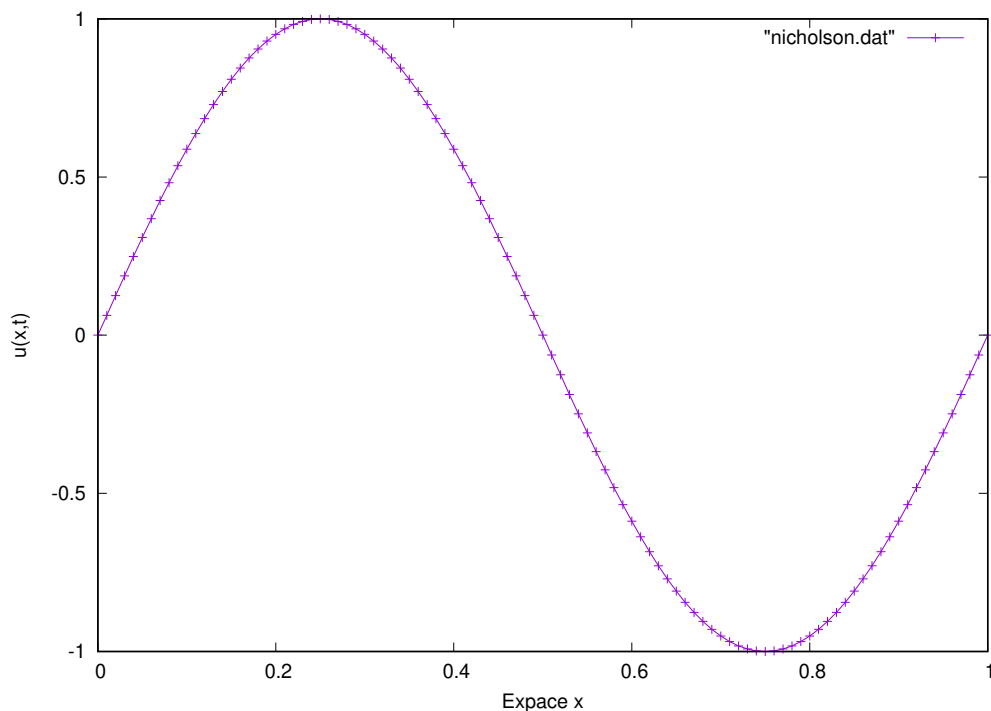


Figure 2.10: Variation de  $u(x, t)$  en fonction de  $x$  dans le cas du schéma de Crank-Nicholson,  $f = f_1$  et  $t$  fixe.

D'après la figure (2.10), on remarque que l'allure de la courbe suit la fonction  $\sin(2\pi x)$ . Il y a ainsi deux points importants:

- Un **maximum**  $u = 1$  atteint dans le cas ou  $x = \frac{1}{4}$ ,
- Un **minimum**  $u = -1$  atteint dans le cas ou  $x = \frac{3}{4}$ .

La figure (2.11) montre la variation de la fonction scalaire  $u(x, t)$  en fonction du temps  $t$  et d'espace  $x$  dans le cas du schéma de Crank-Nicholson. Ainsi, on obtient le même comportement observé dans la figure (2.10) pour des valeurs de temps différentes.

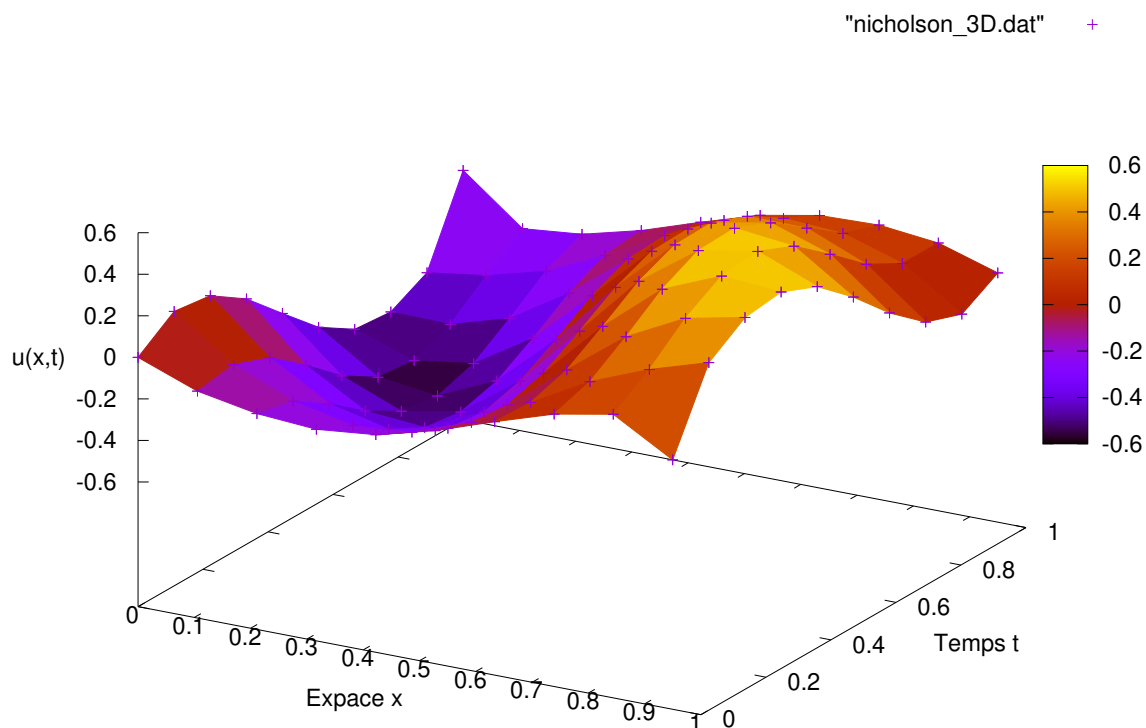


Figure 2.11: Variation de  $u(x,t)$  en fonction d'espace  $x$  et du temps  $t$  dans le cas du schéma de Crank-Nicholson,  $f = f_1$ .

Variation dans le cas de  $f = f_2$  (Equation (2.23):

On se propose dans cette partie de tracer les 2 graphes suivants:

- La variation de  $u(x,t)$  en fonction de  $x$ : Figure (2.12),
- La variation de  $u(x,t)$  en fonction de  $x$  et de  $t$ : Figure (2.13).

Comme la fonction est répartie sur 2 intervalles, elle est alors:

- Croissante sur  $[0, \frac{1}{2}]$ ,
- Décroissante sur  $[\frac{1}{2}, 1]$ .

Elle a comme maximum  $u(x = \frac{1}{2}, t) = 1$  et comme minimum  $u(x = 0, t) = u(x = 1, t) = 0$ . En variant le temps, on obtient également cette allure répartie sur l'intervalle  $[0, T_{obs} = 1]$ .

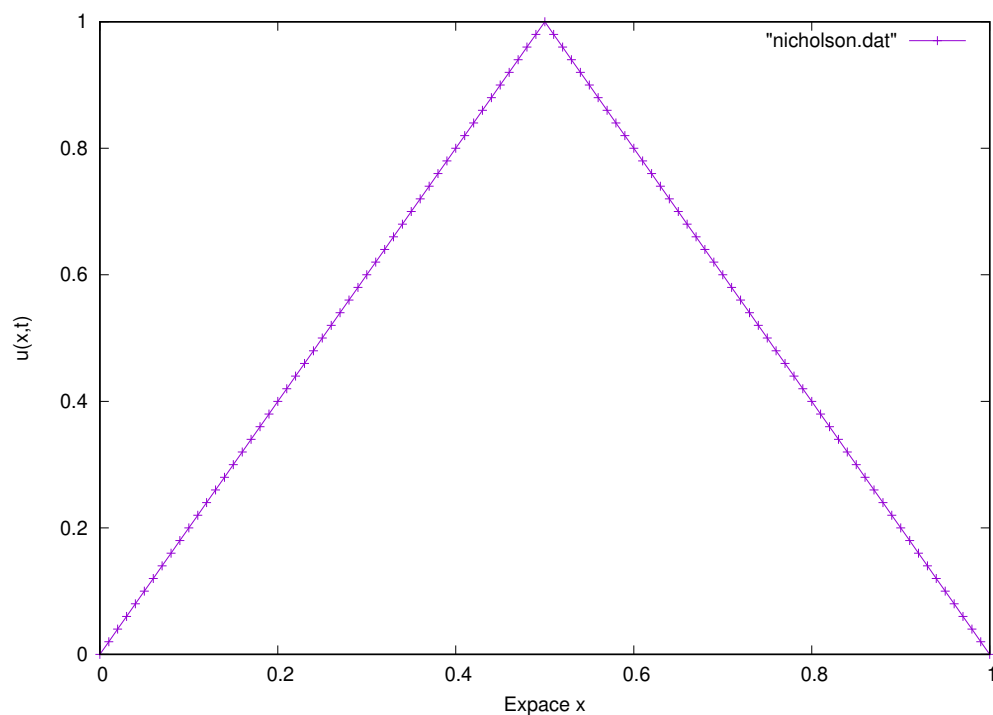


Figure 2.12: Variation de  $u(x,t)$  en fonction de  $x$  dans le cas du schéma de Crank-Nicholson,  $f = f_2$  et  $t$  fixe.

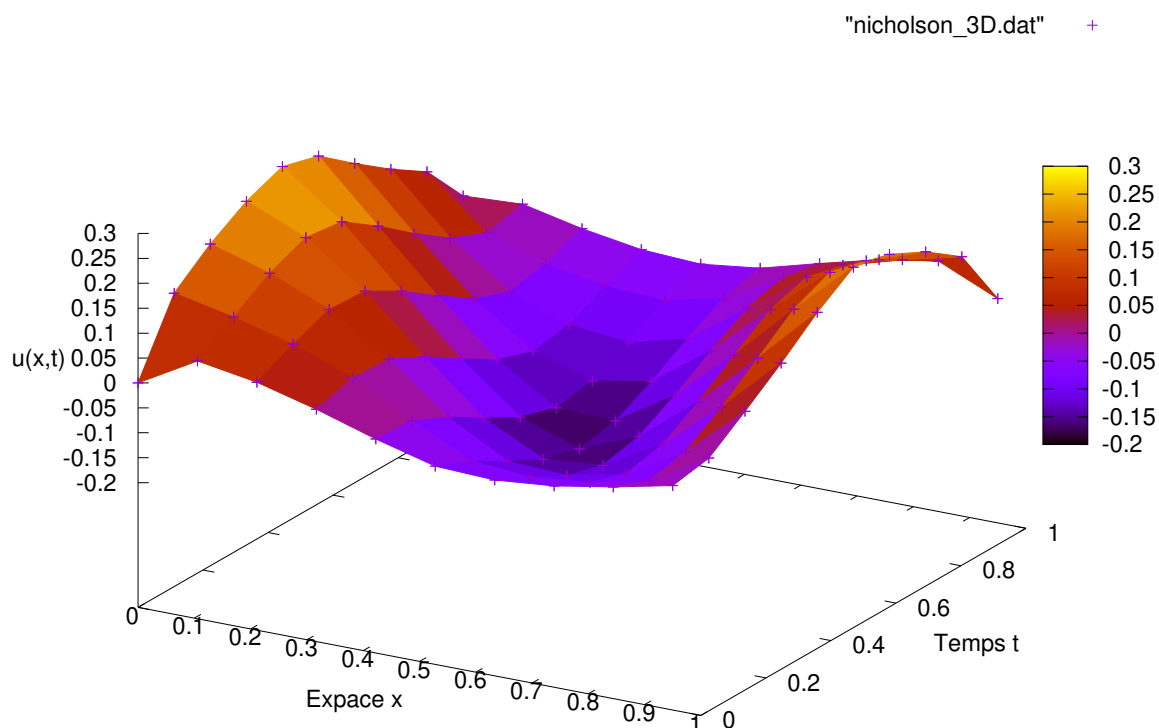


Figure 2.13: Variation de  $u(x,t)$  en fonction de  $x$  et  $t$  dans le cas du schéma de Crank-Nicholson,  $f = f_2$ .



**5) Résultats et comparaison:**

Les résultats pour les trois schémas numériques dans le cas de chacune des conditions aux bords  $f_1$  et  $f_2$  sont analysées dans les sections précédentes.

On conclut que:

- Le maillage en espace avec un pas  $\Delta x$  et la discrétisation en temps avec un pas  $\Delta t$  sont utiles pour calculer la solution approchée  $u_i^j$  de la solution exacte  $u(x, t)$ . Le but de ce TP était de calculer la solution approchée en utilisant la méthode de différences finies moyennant trois schémas numériques (explicite, implicite, Crank-Nicklson).
- Le schéma explicite est conditionnellement stable. D'où le choix judicieux des pas du temps et d'espace  $\Delta t$  et  $\Delta x$  dans la rubrique simulation pour obéir à la condition de stabilité associée,
- Les schémas implicite et de Crank-Nicholson sont inconditionnellement stables,
- Les schémas implicite et explicite sont d'ordre 1 en temps et 2 en espace,
- Le schéma de Crank-Nicholson est d'ordre 2 en temps et d'ordre 2 en espace,
- Les trois schémas donnent des résultats presque identiques.

# References

"*Différences finies*" (2009). Paris Saclay. URL: <https://www.imo.universite-paris-saclay.fr/mobile/~fdubois/cours/idef/idef-ch2.pdf>.

## Appendix A

# Codes des Schémas Numériques en C

Listing A.1: Schéma Explicite en C.

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

%fonction 1
double f_1(double x){
    return sin(2*M_PI*x);
}

%fonction 2
double f_2(double x){
    double exper;

    if ((x > 0) && (x <= 0.5)){
        exper = 2*x; %Expression de la fonction pour x in ]0,1/2]
    }
    else if ((x > 0.5) && (x < 1)){
        exper = 2*(1-x); %Expression de la fonction pour x in ]1/2,1[
    }

    return exper;
}

%Programme principal
int main(){

    FILE *fichier;
    fichier = fopen ("ftcs.dat", "w");

    % ***Declaration des variables***
    int N = 100, M = 10000; %nombre des points selon l espace et le temps

    double dx = 1./((double) N); %pas dans l espace
```

```

double Tobs = 1., dt = Tobs/(double) M; %temps et pas du temps
double r=dt/(dx*dx);
double u[N+1]; %initialisation tab de dim N

% ***conditions du probleme***

%Conditions aux bords
u[0] = 0.; %condition au 1er bord

%Condition initiale
for (int i=1; i<N; i++){
    u[i] = f_1(i*dx); %fonction f_i
}

int k=0; %ecriture des points dans fichier
for (int i = 0; i<=N; i++){
    fprintf(fichier, "%.6lf %.6lf %.6lf \n",k*dt, i*dx, u[i]);
}

for (int j = 1; j<M; j++){
    for (int i = 1; i<N-1; i++){
        u[i] = (1.-2.*r)*u[i] + r*(u[i+1]+u[i-1]); %schema explicite
    }
    if(j%(M/20)==0){ %affichage 3D
        for (int i = 0; i<=N; i++){
            fprintf(fichier, "%.6lf %.6lf %.6lf \n",j*dt, i*dx, u[i]);
        }
    }
}

%affichage 2D
%for (int i=0; i<=N; i++){
    %u[N] = 0.; %condition au dernier bord
    %printf("%.6lf %.6lf \n",i*dx, u[i]); %u en fonction de l'espace
    %fprintf(fichier, "%.6lf %.6lf \n",i*dx, u[i]); %u en fonction de l'espace
    %dans le fichier
    %}

return 0;
}

```

Listing A.2: Schéma Implicite en C.

```

#include <stdlib.h>
#include <stdio.h>
#include <math.h>

%fonction 1
double f_1(double x){
    return sin(2*M_PI*x);
}

%fonction 2
double f_2(double x){

double exper;

    if ((x > 0) && (x <= 0.5)){
        exper = 2*x; %Expression de la fonction pour x in ]0,1/2]
    }
}

```

```

    }
    else if ((x > 0.5) && (x < 1)){
        exper = 2*(1-x); %Expression de la fonction pour x in ]1/2,1[
    }

    return exper;
}

%Programme principal
int main(){

    FILE *fichier;
    fichier = fopen ("implicite.dat", "w"); %fichier 1D

    FILE *fichier2;
    fichier2 = fopen ("implicite_3D.dat", "w"); %fichier 3D

    % ***Declaration des variables***
    int N = 100, M = 10000; %nombre des points selon l espace et le temps

    double dx = 1./((double) N); %pas dans l espace
    double Tob = 1., dt = Tob/((double) M); %temps et pas du temps
    double r=dt/(dx*dx);

    %Initialisation du vecteur u
    double u[N+1][M+1]; %initialisation tab de dim N+1 * M+1

    % ***conditions du probleme***

    % ***Conditions aux bords***
    for (int j=0; j<=M; j++){
        u[0][j]=0.; %1er bord
        u[N][j]=0.; %2eme bord
    }

    % ***Condition initiale***
    for (int i=1; i<N; i++){
        u[i][0] = f_2(i*dx); %fonction f_i
    }

    %initialisation des u[i][j] a l'interieur, except conditions aux bords
    for (int i=1; i<N; i++){
        for (int j=1; j<M; j++){
            u[i][j] = 0.;
        }
    }

    %Schema implicite
    for (int j = 1; j<M; j++){
        for (int i = 1; i<N; i++){

            u[i][j] = (-1.*r)*u[i+1][j+1] + (1+2.*r)*u[i][j+1] + (-1.*r)*u[i-1][j+1];
        }
    }

    %affichage 2D
    for (int i=0; i<=N; i++){
        printf("%.6lf %.6lf \n",i*dx, u[i][0]); %u en fonction de l'espace
        fprintf(fichier, "%.6lf %.6lf \n",i*dx, u[i][0]); %u en fonction de l'
        espace dans le fichier
    }
}

```

```

%affichage 3D
for (int i=0; i<=N; i++){
    for (int j=0; j<=M; j+=100){
        printf("%.6lf %.6lf %.6lf \n",i*dx, j*dt, u[i][j]); %u en fonction de
        l'espace
        fprintf(fichier2, " %.6lf %.6lf %.6lf \n",i*dx, j*dt, u[i][j]); %u en
        fonction de l'espace dans le fichier2
    }
}

return 0;
}

```

Listing A.3: Schéma Crank-Nicholson en C.

```

#include <stdlib.h>
#include <stdio.h>
#include <math.h>

%fonction 1
double f_1(double x){
    return sin(2*M_PI*x);
}

%fonction 2
double f_2(double x){

double exper;

    if ((x > 0) && (x <= 0.5)){
        exper = 2*x; %Expression de la fonction pour x in ]0,1/2]
    }
    else if ((x > 0.5) && (x < 1)){
        exper = 2*(1-x); %Expression de la fonction pour x in ]1/2,1[
    }

    return exper;
}

%Programme principal
int main(){

    FILE *fichier;
    fichier = fopen ("nicholson.dat", "w"); %fichier 1D

    FILE *fichier2;
    fichier2 = fopen ("nicholson_3D.dat", "w"); %fichier 3D

    % ***Declaration des variables***
    int N = 100, M = 10000; %nombre des points selon l'espace et le temps

    double dx = 1./((double) N); %pas dans l'espace
    double Tob = 1., dt = Tob/((double) M); %temps et pas du temps
    double r=dt/(dx*dx);

    %Initialisation du vecteur u
    double u[N+1][M+1]; %initialisation tab de dim N+1 * M+1

```

```
% ***conditions du probleme***

%Conditions aux bords
for (int j=0; j<=M;j++){
    u[0][j]=0.; %1er bord
    u[N][j]=0.; %2eme bord
}

%Condition initiale
for (int i=1; i<N; i++){
    u[i][0] = f_2(i*dx); %fonction f_i
}

%initialisation des u[i][j] a l'interieur, except conditions aux bords
for (int i=1; i<N; i++){
    for (int j=1; j<M; j++){
        u[i][j] = 0.;
    }
}

%Schema de Crank-Nicholson
for (int j = 1; j<M; j++){
    for (int i = 1; i<N; i++){
        u[i][j] = (1./(1.-2.*r)) * ( (-1*r)*u[i-1][j+1] + (1.+2.*r)*u[i][j+1] +
        (-1*r)*u[i+1][j+1] + (-1*r)*u[i-1][j] + (-1*r)*u[i+1][j] );
    }
}

%affichage 2D
for (int i=0; i<=N; i++){
    printf("%.6lf %.6lf \n",i*dx, u[i][0]); %u en fonction de l'espace
    fprintf(fichier, "%.6lf %.6lf \n",i*dx, u[i][0]); %u en fonction de l'
    espace dans le fichier
}

%affichage 3D
for (int i=0; i<=N; i++){
    for (int j=0; j<=M; j+=100){
        printf("%.6lf %.6lf %.6lf \n",i*dx, j*dt, u[i][j]); %u en fonction de
        l'espace
        fprintf(fichier2, " %.6lf %.6lf %.6lf \n",i*dx, j*dt, u[i][j]); %u en
        fonction de l'espace dans le fichier2
    }
}

return 0;
}
```

## Appendix A

# Dépôt Github

Les codes associé à ce rapport sont déposés dans le dépôt git **"TP\_Methode\_Numerique"**.

Le code SSH de ce dépôt est le suivant:

**git@github.com:Chaichas/TP\_Methode\_Numerique.git**

Lien Github: [https://github.com/Chaichas/TP\\_Methode\\_Numerique](https://github.com/Chaichas/TP_Methode_Numerique)