Gorinskaia 3520807

Soualhi 3872967

4AN0: Méthodesnumériquesavancées

Résolution numérique de l'équation de Laplace

PARTIE 1 :

Écrire un programme qui construit et résout le problème discrétisé.

N=4

- On commence par construires les blocks de matrice finale qui sont tous de dimention N-1xN-1.
- Matrice A

```
A = 4*eye(N_dim-1);
A(2:N_dim-1,1:N_dim-2) = A(2:N_dim-1,1:N_dim-2) -
eye(N_dim-2);

A(1:N_dim-2,2:N_dim-1) = A(1:N_dim-2,2:N_dim-1) -
eye(N_dim-2);
A=A*(-1);
```

- Matrice Identite et Matrice nulle (dimentions N-1xN-1)

```
ZN = zeros(N_dim-1);
IN = eye(N_dim-1);
```

- On compose la matrice globale U (dimention NxN) a partir des matrices-lignes U1, U2, U3, U4.

```
U1= [A IN ZN ZN];
U2= [IN A IN ZN];
U3= [ZN IN A IN];
U4= [ZN ZN IN A/2];

U=[U1; U2; U3; U4]
```

- On veur construire le syseme global U*T=B avec B=-Tg-Th-Tb

```
Tg=zeros(N_dim*(N_dim-1),1);
Tb=Tg;
Th=[zeros(N_dim-2,1)' 1]';
Th=[Th; Th; Th; 0.5*Th]
```

```
B=-(Tg+Th+Tb)
```

- Finalement, on retrouve T:

```
T = U\B
```

Visualisation

- Pas d'espace en x et y est 1/N

```
N = N_dim;
pas_h = 1/N;
```

- On cree les vecteurs (dimention N+1) pour les valeurs de x et y, de 0 a 1 inclus

```
x_discretise = ([1:N+1]-1)*pas_h;
y_discretise = ([1:N+1]-1)*pas_h;
```

- Construction d'une grille de maillage
- X_grid : contient les abcisses de tous les points de la grille
- Y_grid : contient les ordonnees de tous les points de la grille

```
[X_grid,Y_grid] = meshgrid(x_discretise,y_discretise)
```

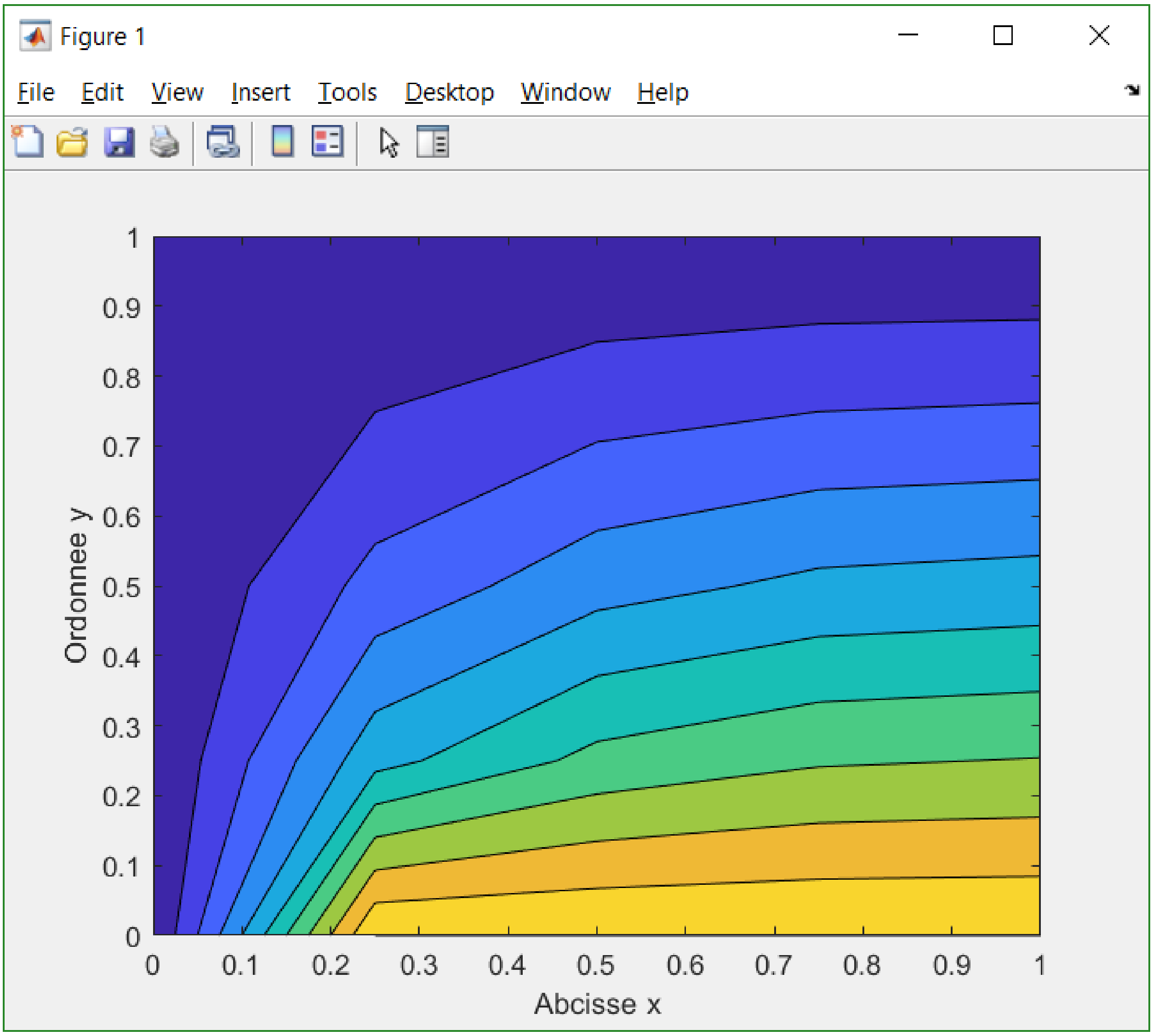
- On reorganize le vecteur T pour obtenir une matrice N-1xN

```
T_grid = reshape(T,N-1,N)
T_grid=[flipud(T_grid(:,1)),flipud(T_grid(:,2)),flipud(T_gr
id(:,3)),flipud(T_grid(:,4))]
```

- On construit la matrice globale O a partir des vecteurs V1 (qui represent le bord haut), V2 (bord gauche), V3 (bord bas) et le vecteur T_grid, qui contient les coefficients du milieu. On affiche le resultat grace a une fonction `contourf`.

```
V1=[0 ones(1,N_dim)]
V2=zeros(N_dim-1,1);
V3=zeros(1,N_dim+1);

O=[V1; V2 T_grid ;V3]
figure(1)
title('N=4')
contourf(X_grid,Y_grid,O)
xlabel('Abcisse x'), ylabel('Ordonnee y')
```

Sur la figure obtenu on reconnait bien l'allure typique des equations de propagation.



N=5

La resolution pour N=5 est tres semblant de solution avec N=4.

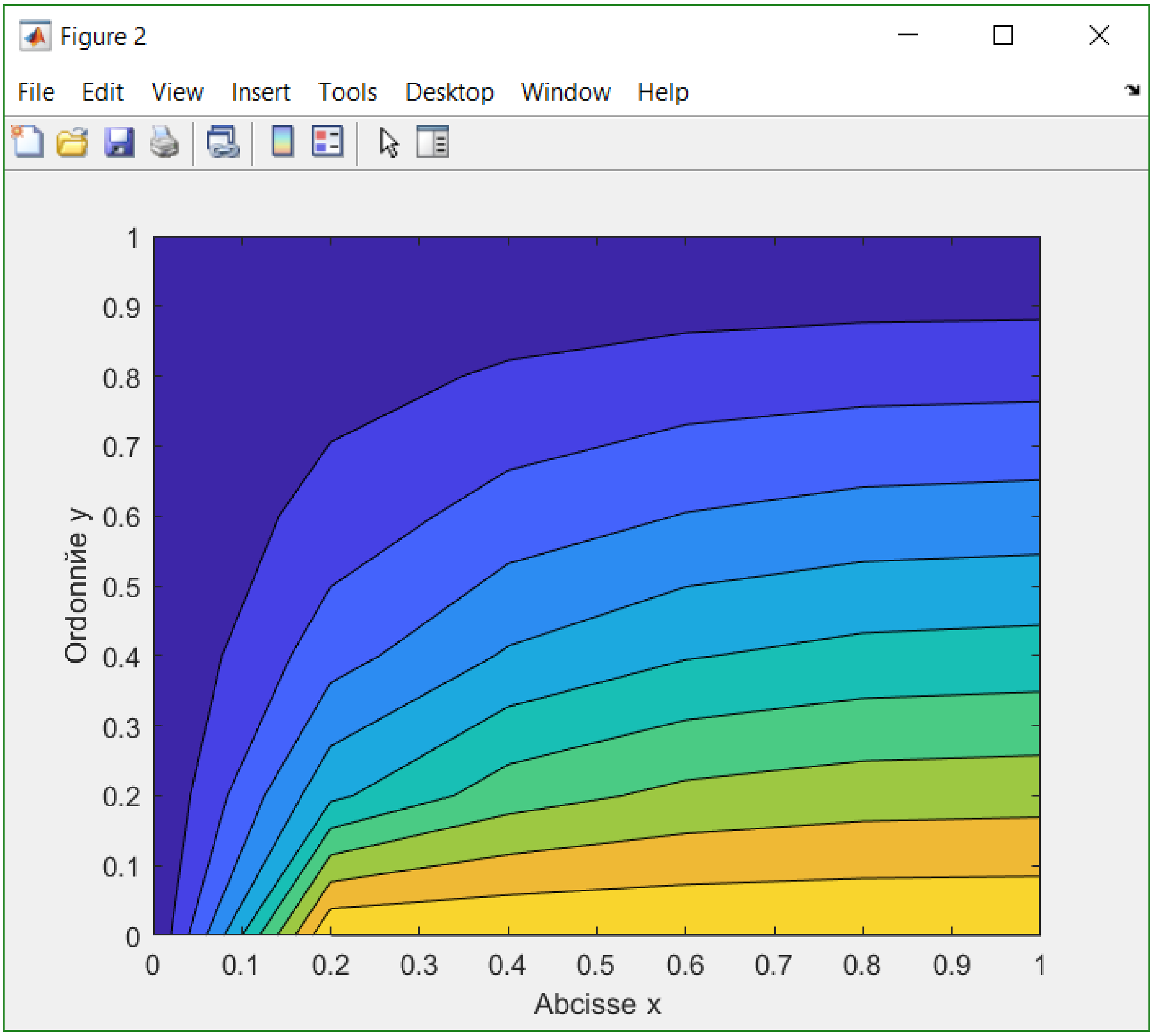
- Notre vecteur U va etre compose de 5 lignes U1-U5 :

```
U1= [A IN ZN ZN ZN];
U2= [IN A IN ZN ZN];
U3= [ZN IN A IN ZN];
U4= [ZN ZN IN A IN];
U5= [ZN ZN ZN IN A/2];

U=[U1; U2; U3; U4 ; U5]
```

- Le T_grid aura, de meme maniere, les dimentions N-1xN

```
T_grid = reshape(T,N-1,N)
T_grid=[flipud(T_grid(:,1)),flipud(T_grid(:,2)),flipud(T_grid(:,3)),flipud(T_grid(:,4)),flipud(T_grid(:,5))]
```

On remarque que avec l’augmenation de dimention de calcul l’allure de la figure devient plus proche de la realite physique.



N quelconque

- On doit construire les matrices A , Identite et Nulle (toutes de dim N-1xN-1)

```
N=50;

N_dim = N-1;
A = 4*eye(N_dim);
A(2:N_dim,1:N_dim-1) = A(2:N_dim,1:N_dim-1) - eye(N_dim-1);
A(1:N_dim-1,2:N_dim) = A(1:N_dim-1,2:N_dim) - eye(N_dim-1);
A=-A;
```

- On cree la matrive U initiale et commence a la remplir afin d'oblenir le notif -1 4 1 sur la diagonale (dim N-1*N x N-1*N)

```
U = zeros((N-1)*N);

for  i=1:N
    M = U(i*N_dim-(N_dim-1): i*N_dim , i*N_dim-(N_dim-1):
i*N_dim  ) + A;
    U(i*N_dim-(N_dim-1): i*N_dim , i*N_dim-(N_dim-1):
i*N_dim  ) = M;
end

M = U(N*N_dim-(N_dim-1): N*N_dim , N*N_dim-(N_dim-1):
N*N_dim  ) - (0.5*A);
U(N*N_dim-(N_dim-1): N*N_dim , N*N_dim-(N_dim-1):
N*N_dim  ) = M;

I3 = eye(N_dim);

for  i=1:N_dim

   M = U(((i+1)*N_dim)-(N_dim-1): (i+1)*N_dim , (i)*N_dim-
(N_dim-1): i*N_dim) + I3;
    U(((i+1)*N_dim)-(N_dim-1): (i+1)*N_dim , (i)*N_dim-
(N_dim-1): i*N_dim) = M;

   M = U(((i)*N_dim)-(N_dim-1): (i)*N_dim , (i+1)*N_dim-
(N_dim-1): (i+1)*N_dim) + I3;
    U(((i)*N_dim)-(N_dim-1): (i)*N_dim , (i+1)*N_dim-
(N_dim-1): (i+1)*N_dim) = M;

end
```

- Resolution des matrices Tg, Th, Tb. Ub, Ug et Uh – variables temporelles pour les conditions citiques.

```
 Ub=0;
```

```
 Ug=0;
 Uh=1;

B=zeros(N*(N-1),1);
Tb=zeros(N*(N-1),1);
Th=zeros(N*(N-1),1);
Tg=zeros(N*(N-1),1);


for  i=1:N

    Th(i*N_dim) = Tg(i*N_dim) + Uh ;

end
```

- Resolution de B, le dernier coefficient de matrice etant 0.5 (cf preparation theorique)
-

```
B = -Th -Tg -Tb;
B(end)=-0.5;
```

- On retrouve la matrice T, comme dans les exemples precedents

```
T = U\B;
```

<u>Visualisation</u>

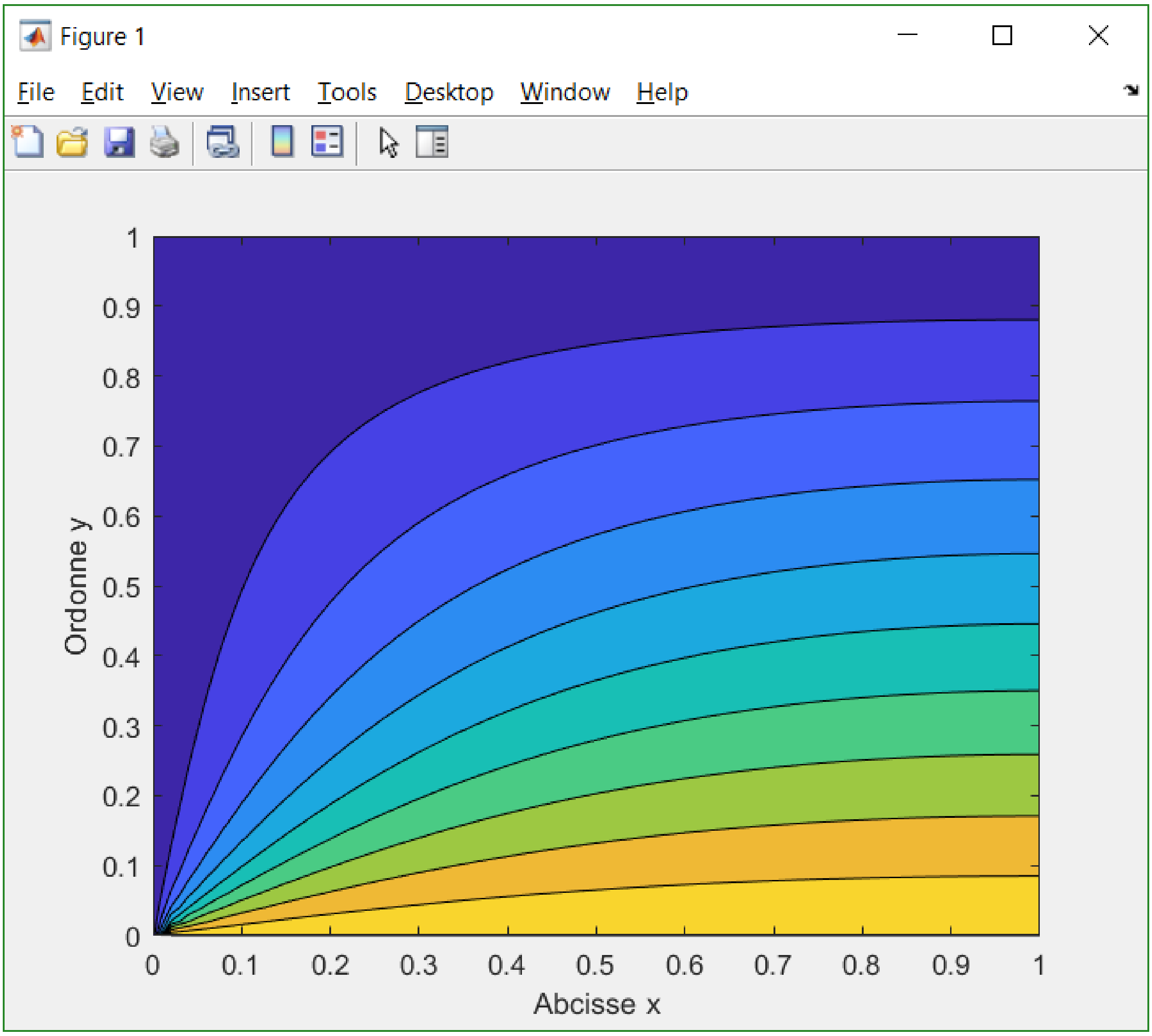
Visualisation se passe de la meme maniere que pour N=4. Pour reconstruire T_grid on utilise la fonction `flipud`

```
pas_h = 1/N ;
x_discretise = ([1:N+1]-1)*pas_h;
y_discretise = ([1:N+1]-1)*pas_h;

[X_grid,Y_grid] = meshgrid(x_discretise,y_discretise);

T_grid = reshape(T,N-1,N);
```

- Pour reconstruire T_grid on utilise la fonction `flipud`

```
T_grid=flipud(T_grid);

V1=[0 ones(1,N)];
V2=zeros(N_dim,1);
V3=zeros(1,N+1);
O=[V1; V2 T_grid ;V3];
```

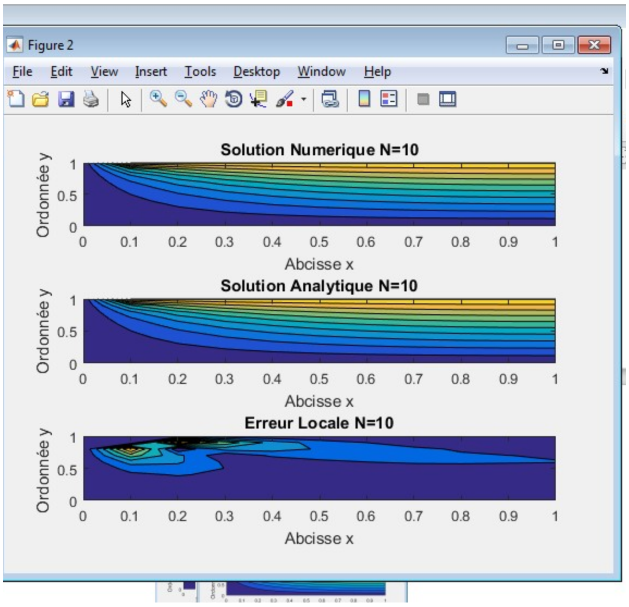
```
figure(1)
contourf(X_grid,Y_grid,O)
xlabel('Abcisse x'), ylabel('Ordonne y')
```



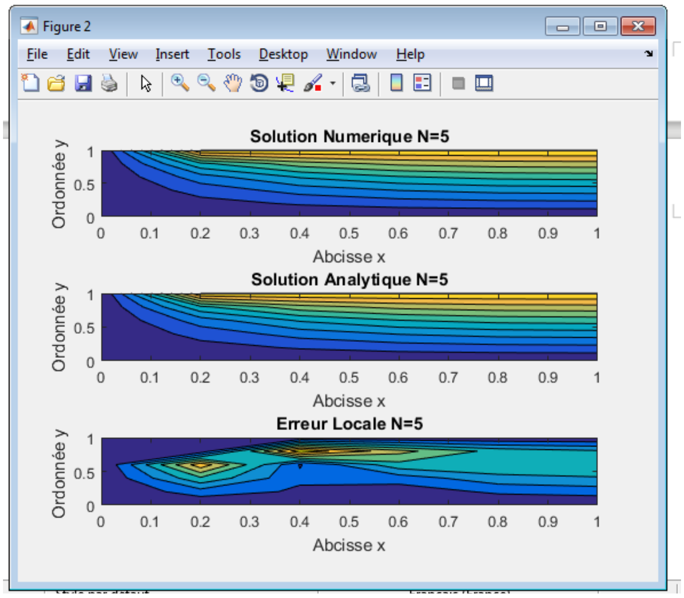
Utilisation de N important:

+ Precision
+ Plus proche de la realite physique
- Augmentation de cout de calcul

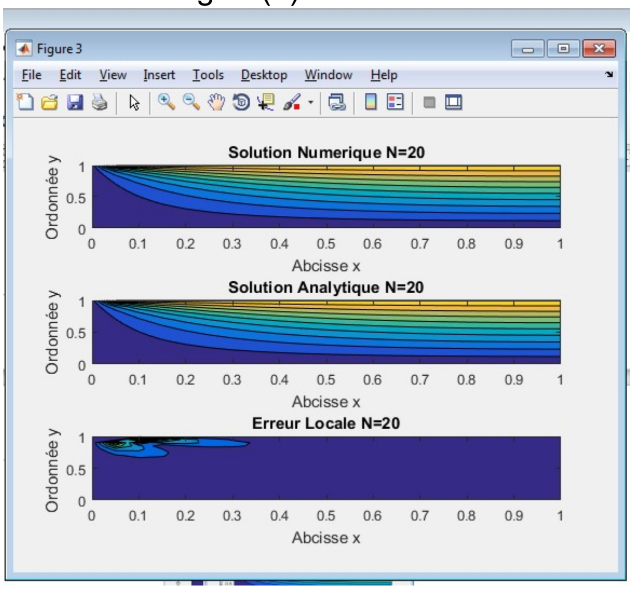
PARTIE 2 :



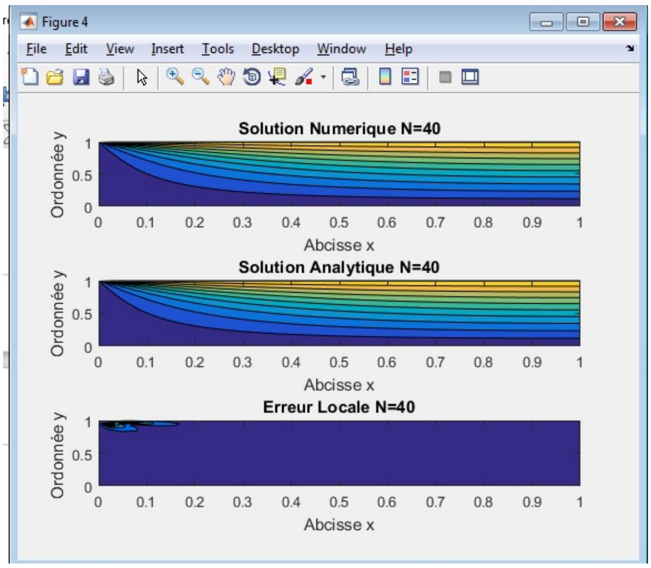
figure(1)



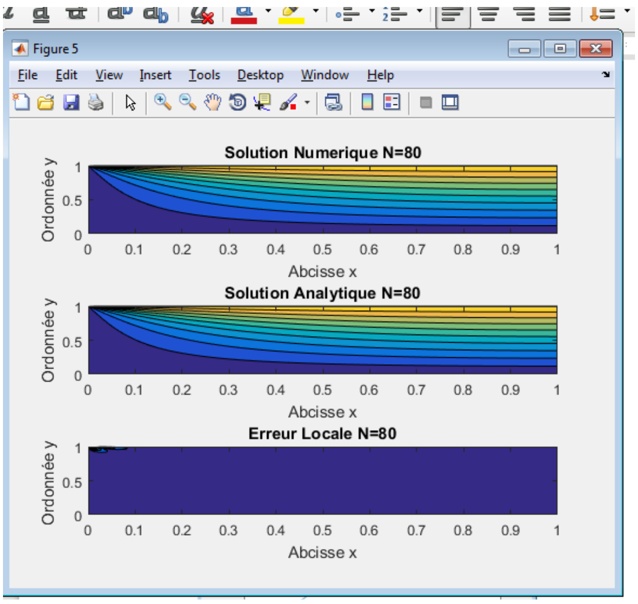
figure(2)



figure(3)



figure(4)



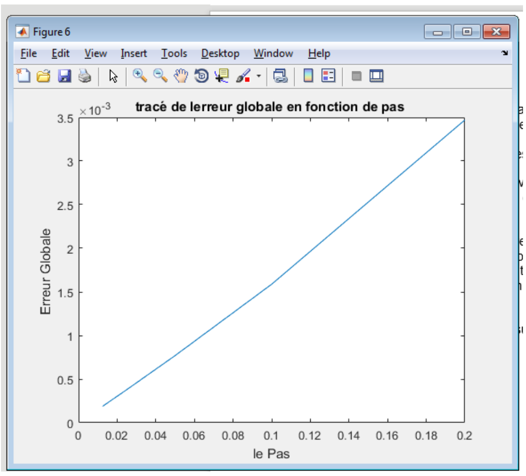
figure(5)

Commentaires :
plusieurs remarques sont été vus :
1- sur figure(1) et figure(2) Pour des faibles valeurs de N, la l'erreur locale de la solution numérique existe pour presque toute les valeurs, ce qui a résulter en une erreur globale importante
aussi c'est remarquable sur les courbes qu'elles sont déformée a cause du pas grand

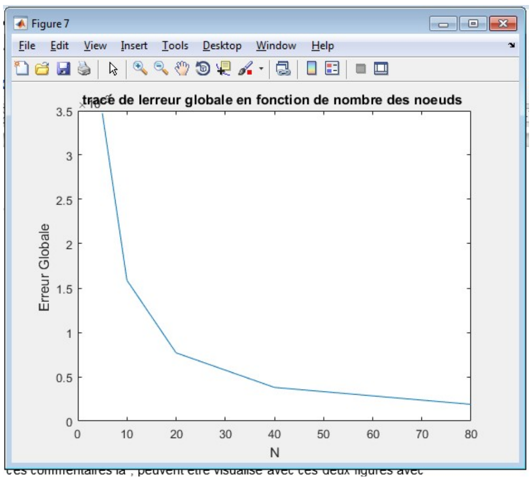
2- figure(3) : a partir de N=20 on peut voir que l'erreur locale est nuls sur quelques neouds donc le calcul devient de plus en plus précis, et l'erreur globale a diminué par rapport aux cas précédents

3- figure(4) et figure(5) : pour ces figures la on peut voir que l'erreur locale est nuls partout sauf quelques point ,, ce qui nous a donné une erreur globale d'un ordre très petit, au niveau des courbes, elles sont maintenant plus précises et presque identiques a cause du pas très élevé (le nombre important des nœuds)

ces commentaires la , peuvent être visualisé avec ces deux figures ci dessous :



figure(6)



figure(7)

sur la figure(6) on peut remarqué que plus que le pas est plus petit plus que l'erreur globale est plus petite mais elle est jamais nul
sur la figure(7) : on peut voir que plus le nombre des neouds est important, plus l'erreur globale est petite

Conclusion :
on peut conclure que :
- une solution analytique est toujours plus précise qu'une solution numérique
- la solution numérique est plus précise pour un nombre important des nœuds, mais c'est très coûteux au niveau des calculs (temps et énergie)
- plus le nombre des neouds est important plus le la solution numérique est proche a la solution Analytique