TP 4 : Problèmes dépendant du temps (solutions)

Pierre.Saramito@imag.fr
http://www-lmc.imag.fr/lmc-edp/Pierre.Saramito

20 novembre 2000

Première partie

Actions 1 et 2

La commande de lancement sous matlab est :

```
resoudre_temps(0.1,100,20)
```

correspondant à $\Delta t = 0.1$, $n = 100$ et kfin=20.

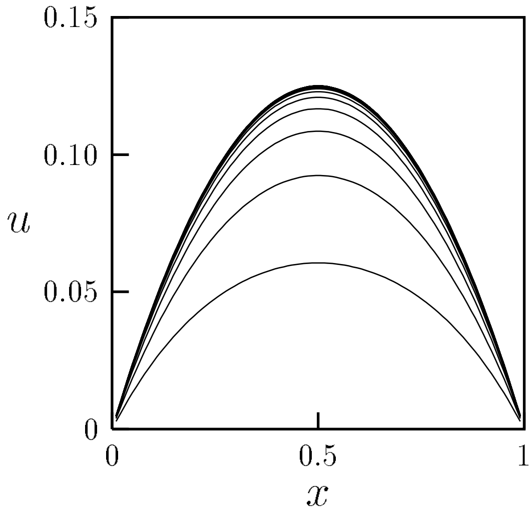

FIG. 1 – *Évolution au cours du temps de la solution depuis une position au repos et vers un état stationnaire.*

Il y a plusieurs manière de programmer la résolution du schéma. Le code proposé n'utilise qu'un vecteur `u` de taille n et réutilise directement la fonction `initialise_matrice(n)` du TP 1.

Fichier ‘`resoudre_temps.m`’

```
function u = resoudre_temps(dt,n,kfin)

   h = 1/(n+1);
   for i = 1:n
     u(i) = 0;
     x(i) = i*h;
   end
   a = init_matrice(n) + (h*h/dt)*eye(n,n);
   hold on;
   for k = 1:kfin
     for i = 1:n
       b(i,1) = h*h + (h*h/dt)*u(i);
     end
     u = a\b;
     plot (x,u);
     pause(1);
   end
   hold off;
```

Action 3

Le système tends vers un état stationnaire. La différence entre deux courbes n'est plus graphiquement perceptible sur la figure 1 à partir du huitième itéré, ce qui correspond à un temps $t_f = 0.8$, le pas de temps étant $\Delta t = 0.1$.

Deuxième partie

Action 4

Le problème approché par différences finies s'écrit :

$(P)_{\Delta t,h}$: trouver $(u_i^k)_{0\leq i\leq n+1, k\geq 0}$ tel que

$$\begin{aligned}
-\alpha_{i+\frac{1}{2}}u_{i+1}^k + \left(\alpha_{i+\frac{1}{2}} + \alpha_{i-\frac{1}{2}} + \frac{h^2}{\Delta t}\right) u_i^k - \alpha_{i-\frac{1}{2}}u_{i-1}^k &= h^2 f(t_k, x_i) + \frac{h^2}{\Delta t}u_i^{k-1} \text{ pour } 1 \leq i \leq n \text{ et } k \geq 1 \\
u_0^k &= u_{n+1}^k = 0 \text{ pour } k \geq 1 \\
u_i^0 &= u_0(x_i) \text{ pour } 0 \leq i \leq n+1
\end{aligned}$$

Actions 5 et 6

La commande de lancement est :

```
resoudre_t_temps(10,100,20,0.001);
```

correspondant à $\Delta t = 10$, $n = 100$, `kfin=20` et $\varepsilon = 10^{-3}$.

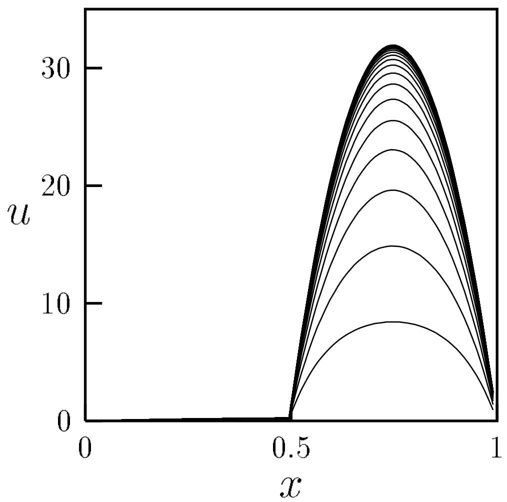

FIG. 2 – *Évolution dans le cas du problème de transmission.*

La programmation est très voisine de celle de la première partie. Seule diffère l'initialisation de la matrice a.

Fichier 'resoudre_t_temps.m'

```
function u = resoudre_t_temps(dt,n,kfin, epsilon)

   h = 1/(n+1);
   for i = 1:n
     u(i) = 0;
     x(i) = i*h;
   end
   a = init_matrice_t(n,epsilon) + (h*h/dt)*eye(n,n);
   hold on;
   for k = 1:kfin
     for i = 1:n
       b(i,1) = h*h + (h*h/dt)*u(i);
     end
     u = a\b;
     plot (x,u);
   end
   hold off;
```

Action 7

Les courbes se superposent sur le graphique de la figure 2 à partir de la quatorzième itération, ce qui correspond à un temps $t_f = 140$, le pas de temps étant $\Delta t = 10$.