

BE de Méthodes Numériques pour les EDP
Compléments d'information pour le code d'advection 2D

1 Codage du flux numérique 1D

On rappelle que le schéma de résolution de l'équation d'advection scalaire 1D peut se mettre sous la forme :

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x} (h_{i+\frac{1}{2}}^n - h_{i-\frac{1}{2}}^n) \quad (1)$$

où le flux numérique considéré est un flux dissipatif d'expression générale :

$$h_{i-\frac{1}{2}}^n = h(\rho_{i-1}^n, \rho_i^n) = \frac{1}{2}(f_i^n + f_{i-1}^n) - \frac{1}{2}q_{i-\frac{1}{2}}^n(\rho_i^n - \rho_{i-1}^n) \quad (2)$$

avec un coefficient de dissipation numérique $q_{i-\frac{1}{2}} = q(\rho_{i-1}^n, \rho_i^n)$.

La formule (1) est utilisée pour faire évoluer les états dits intérieurs (voir Fig. 1) du domaine de calcul. Les états aux frontières évoluent en fonction des conditions aux limites utilisées. On choisit

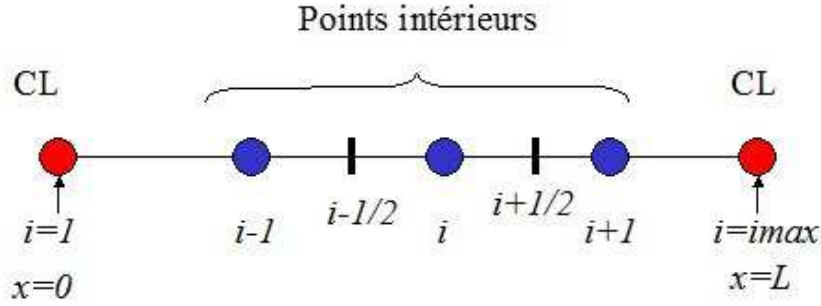


Fig. 1: Vue schématique du maillage pour le problème d'advection 1D. Les points intérieurs sont indiqués en bleu et les points frontières en rouge.

de coder dans une fonction f_phys l'expression du flux physique f en fonction de la variable ρ et de coder dans une fonction q_num l'expression du coefficient de dissipation numérique en $i - \frac{1}{2}$ (ou $i + \frac{1}{2}$) en fonction des états ρ_{i-1} , ρ_i (ou ρ_i , ρ_{i+1}).

On choisit de stocker le flux numérique au point $i + \frac{1}{2}$, $h_{i+\frac{1}{2}}$, dans le tableau $fnum(i)$. Puisque ρ_i^n est stocké dans le tableau $rho(i)$, la formule (2) se traduit par :

$$fnum(i) = \frac{1}{2}(f_phys(rho(i)) + f_phys(rho(i+1))) - \frac{1}{2}q_num(rho(i), rho(i+1))(rho(i+1) - rho(i))$$

Compte tenu de ce choix de stocker $h_{i+\frac{1}{2}}$ dans $fnum(i)$, le bilan de flux numériques (1) s'écrit aussi :

$$rho(i) = rho(i) - \frac{\Delta t}{\Delta x} (fnum(i) - fnum(i-1))$$

Cette formule est utilisée pour faire avancer les états intérieurs donc pour les indices $i = 2$ à $i = imax - 1$. On code ainsi dans *scalaire.sce* :

```
rho(2:imax-1) = rho(2:imax-1)-(dt/dx) * (fnum(2:imax-1) - fnum(1:imax-2));
```

ou bien, de façon équivalente mais moins efficace car n'exploitant pas le caractère vectoriel de *Scilab* :

```
for i=2:imax-1
rho(i) = rho(i)-(dt/dx) * (fnum(i) - fnum(i-1));
end
```

On constate immédiatement que $fnum(i)$ doit être calculé pour $i = 1$ à $i = imax - 1$ soit, dans le script *flux_numerique.sce* :

```
for i=1:imax-1
q(i)=q_num(rho(i),rho(i+1));
fnum(i) = 0.5*(f_phys(rho(i))+f_phys(rho(i+1)))-0.5*q(i)*(rho(i+1)-rho(i));
end
```

2 Codage du flux numérique 2D

On souhaite étendre la démarche précédente en deux dimensions d'espace. L'état au point de maillage (i, j) évolue suivant la formule d'approximation de l'EDP d'advection 2D :

$$\rho_{i,j}^{n+1} = \rho_{i,j}^n - \Delta t \left[\frac{h_{i+\frac{1}{2},j}^1 - h_{i-\frac{1}{2},j}^1}{\Delta x} + \frac{h_{i,j+\frac{1}{2}}^2 - h_{i,j-\frac{1}{2}}^2}{\Delta y} \right] \quad (3)$$

avec des flux numériques suivant les directions x et y du maillage donnés par :

$$\begin{cases} h_{i+\frac{1}{2},j}^1 = \frac{1}{2} (f_{i+1,j}^1 + f_{i,j}^1) - \frac{1}{2} q_{i+\frac{1}{2},j}^1 (\rho_{i+1,j} - \rho_{i,j}) \\ h_{i,j+\frac{1}{2}}^2 = \frac{1}{2} (f_{i,j+1}^2 + f_{i,j}^2) - \frac{1}{2} q_{i,j+\frac{1}{2}}^2 (\rho_{i,j+1} - \rho_{i,j}) \end{cases} \quad (4)$$

où les coefficients de dissipation numérique associés à chaque direction spatiale sont tels que $q_{i+\frac{1}{2},j}^1 = q^1(\rho_{i,j}, \rho_{i+1,j})$, $q_{i,j+\frac{1}{2}}^2 = q^2(\rho_{i,j}, \rho_{i,j+1})$.

Le schéma d'évolution (3) s'applique aux points intérieurs du maillage (voir Fig.2). On choisit de coder (par exemple) :

- dans une fonction *f1_phys* l'expression du flux physique f^1 (ou f , flux physique associé à la première direction d'espace) en fonction de la variable ρ ,
- dans une fonction *f2_phys* l'expression du flux physique f^2 (ou g , flux physique associé à la seconde direction d'espace) en fonction de la variable ρ ,
- dans une fonction *q1_num* l'expression du coefficient de dissipation numérique en $i - \frac{1}{2}, j$ (ou $i + \frac{1}{2}, j$) en fonction des états $\rho_{i-1,j}$, $\rho_{i,j}$ (ou $\rho_{i,j}$, $\rho_{i+1,j}$),
- dans une fonction *q2_num* l'expression du coefficient de dissipation numérique en $i, j - \frac{1}{2}$ (ou $i, j + \frac{1}{2}$) en fonction des états $\rho_{i,j-1}$, $\rho_{i,j}$ (ou $\rho_{i,j}$, $\rho_{i,j+1}$).

On choisit de stocker le flux numérique au point $i + \frac{1}{2}, j$, $h_{i+\frac{1}{2},j}^1$, dans le tableau $f1num(i, j)$. Puisque $\rho_{i,j}^n$ est stocké dans le tableau $rho(i, j)$, la formule (4) - pour la direction x - se traduit par :

$$f1num(i) = \frac{1}{2}(f1_phys(rho(i, j)) + f1_phys(rho(i+1, j))) - \frac{1}{2}q1_num(rho(i, j), rho(i+1, j))(rho(i+1, j) - rho(i, j))$$

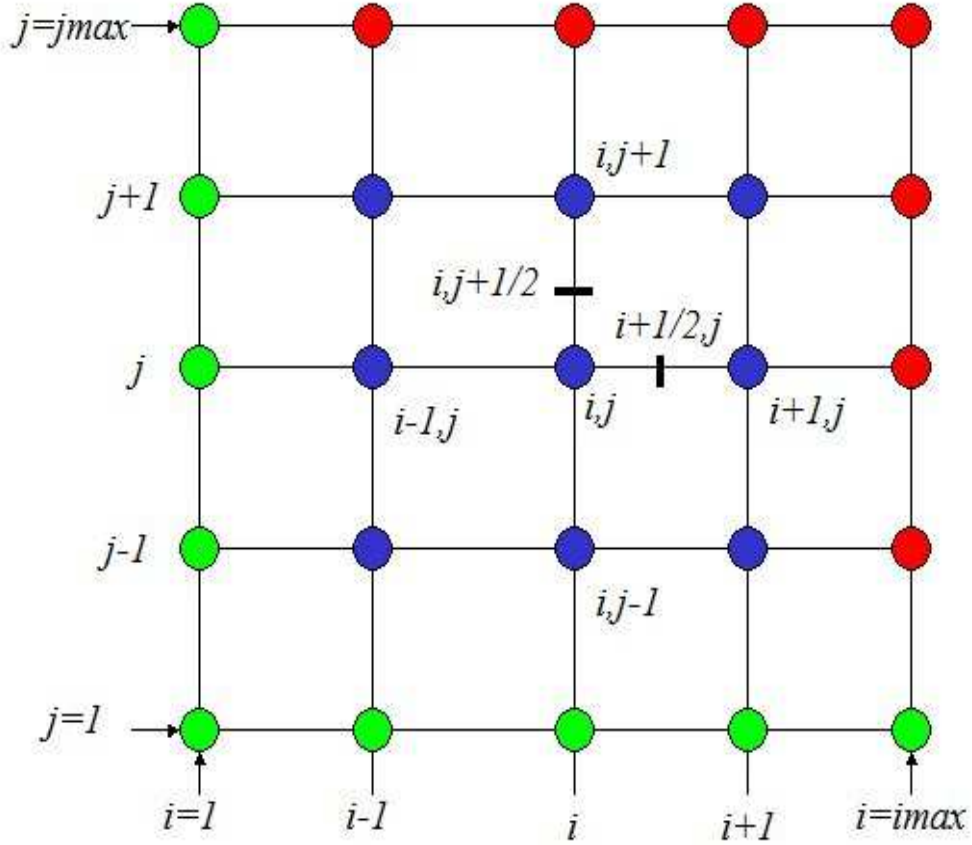


Fig. 2: Vue schématique du maillage pour le problème d'advection 2D. Les points intérieurs, auxquels s'appliquent le schéma d'évolution, sont en bleu ; les points frontières où s'appliquent les conditions aux limites d'entrée (état imposé) sont en vert ; les points frontières où s'appliquent les conditions aux limites de sortie (extrapolation de l'état physique depuis l'intérieur du domaine) sont en rouge.

De façon similaire, le flux numérique au point $i, j + \frac{1}{2}$, $h_{i,j+\frac{1}{2}}^2$, est stocké dans le tableau $f2num(i, j)$ et on a :

$$f2num(i) = \frac{1}{2}(f2_phys(rho(i, j)) + f2_phys(rho(i, j + 1))) - \frac{1}{2}q2_num(rho(i, j), rho(i, j + 1))(rho(i, j + 1) - rho(i, j))$$

Compte tenu de ce choix de stocker $h_{i+\frac{1}{2},j}^1$ dans $f1num(i)$ et $h_{i,j+\frac{1}{2}}^2$ dans $f2num(i)$, le bilan de flux numériques (3) s'écrit aussi :

$$rho(i, j) = rho(i, j) - \frac{\Delta t}{\Delta x}(f1num(i, j) - f1num(i - 1, j)) - \frac{\Delta t}{\Delta y}(f2num(i, j) - f2num(i, j - 1))$$

Cette formule est utilisée pour faire avancer les états intérieurs donc pour les indices $i = 2, imax - 1$ et $j = 2, jmax - 1$ (voir Fig. 2). On code ainsi dans *scalaire.sce* :

```
for i=2:imax-1
for j=2:jmax-1
rho(i, j)=rho(i, j)-(dt/dx)*(f1num(i, j)-f1num(i-1, j))-(dt/dy)*(f2num(i, j)-f2num(i, j-1));
end
end
```

On constate immédiatement que $f1num(i, j)$ doit être calculé pour $i = 1, imax - 1$ et $j = 2, jmax - 1$ tandis que $f2num(i, j)$ doit être calculé pour $i = 2, imax - 1$ et $j = 1, jmax - 1$. On écrit donc dans le script *flux_numerique.sce* :

```
for i=1:imax-1
for j=2:jmax-1
q1(i, j)=q1_num(rho(i, j), rho(i+1, j));
f1num(i, j)=0.5*(f1_phys(rho(i, j))+f1_phys(rho(i+1, j)))
-0.5*q1(i, j).*(rho(i+1, j)-rho(i, j));
end
end

for i=2:imax-1
for j=1:jmax-1
q2(i, j)=q2_num(rho(i, j), rho(i, j+1));
f2num(i, j)=0.5*(f2_phys(rho(i, j))+f2_phys(rho(i, j+1)))
-0.5*q2(i, j).*(rho(i, j+1)-rho(i, j));
end
end
```

3 Codage des conditions aux limites

Comme indiqué sur la figure 2, les états $rho(i, j)$ avec $j = 1$ et $i = 1, imax$ (le long de $y = 0$ donc) d'une part et avec $i = 1$ et $j = 1, jmax$ (le long de $x = 0$ donc) d'autre part sont imposés par les conditions aux limites :

$$\begin{cases} \rho(x, 0) = \rho_0(x) \\ \rho(0, y) = \rho_0(0) \end{cases} \quad (5)$$

En projection dans le maillage on a donc :

$$\begin{cases} \rho(i, 1) = \rho_inlet(x(i)) & \text{pour } i = 1, imax \\ \rho(1, j) = \rho_inlet(x(1)) & \text{pour } j = 1, jmax \end{cases}$$

où on a introduit une fonction ρ_inlet que l'on pourra définir dans *model.sce* et qui permet à l'utilisateur de choisir sa condition d'entrée en modifiant simplement cette fonction, sans toucher à la routine de conditions aux limites. On pourra ainsi coder :

```
function y=rho_inlet(x)

if iinlet==0 then

if x > 0.2 & x < 0.5 then
y=1
else
y=0
end

end

endfunction
```

Pour les conditions de sortie, la théorie des caractéristiques nous suggère de prendre pour l'état frontière l'état au point intérieur le plus proche. On écrira ainsi le long de $x = 2$ ($i = imax$, $j = 1, jmax$) et de $y = 1$ ($j = jmax$, $i = 1, imax$) :

$$\begin{cases} \rho(imax, j) = \rho(imax - 1, j) & \text{pour } j = 1, jmax \\ \rho(i, jmax) = \rho(i, jmax - 1) & \text{pour } i = 1, imax \end{cases}$$

4 Codage de la norme du résidu

Pour calculer la norme du résidu on doit disposer de $\rho_{i,j}^{n+1}$ et $\rho_{i,j}^n$ en tout point du maillage. On a noté précédemment que la formule d'évolution pour ρ :

```
for i=2:imax-1
for j=2:jmax-1
rho(i,j)=rho(i,j)-(dt/dx)*(f1num(i,j)-f1num(i-1,j))-(dt/dy)*(f2num(i,j)-f2num(i,j-1));
end
end
```

est telle que l'on affecte la nouvelle valeur $\rho_{i,j}^{n+1}$ au tableau $\rho(i, j)$ qui stockait précédemment $\rho_{i,j}^n$. Pour être en mesure de calculer aisément le résidu, on introduit donc le tableau *rhoold* dans lequel on stockera l'état ρ , c'est-à-dire $\rho_{i,j}^n$ avant d'appliquer le schéma d'évolution ci-dessus. La norme du résidu est alors simplement calculée en écrivant :

```
norm_res=0;
for i=1:imax
for j=1:jmax
```

```

norm_res=norm_res+(rho(i,j)-rhoold(i,j))**2;
end
end
norm_res=sqrt(norm_res/(imax*jmax));

```

5 Rappels sur les commandes de visualisation

Pour tracer une vue en élévation de la solution dans le plan de calcul on utilisera :

```

xset("window",1)
plot3d(x,y,rho);

```

Pour tracer les isovaleurs de la solution dans le plan de calcul on utilisera :

```

xset("window",2)
contour2d(x,y,rho,20);

```

Pour visualiser par exemple la distribution calculée et exacte de la solution le long de la frontière supérieure du domaine de calcul (en superposant sur une même courbe ces distributions) on utilisera :

```

xset ("window",4)
plot2d(x(1:imax),[rho(1:imax,jmax),rhoex(1:imax,jmax)], [1,2]);

```

sous réserve bien sûr d'avoir défini au préalable le tableau *rhoex* qui stocke la solution stationnaire exacte en tout point du maillage.