

Table des matières

1	Résolution d'un système raide d'EDO	1
1.1	Schémas IMEX Runge-Kutta (IMEX-RK)	1
1.2	Les schémas SSP	2
1.2.1	SSP2	2
1.2.2	SSP3	2
2	Résolution de $\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(f(u)) = g(u)$	3
2.1	Discrétisation en espace	3
2.2	Calcul du flux	3
2.3	Discrétisation en temps pour $f(u) = \alpha u, \alpha > 0$: schémas IMEX-RK	4
2.4	Discrétisation en temps pour $f(u) = \alpha u, \alpha < 0$: schémas IMEX-RK	5
3	Résolution de $\frac{\partial u}{\partial t} - D\Delta(p(u)) = g(u) + f$	6
3.1	Transformation du problème	6
3.2	En dimension $d = 1$.	7
3.2.1	Discrétisation en espace	8
3.2.2	Discrétisation en temps : schémas IMEX-RK	9
3.3	En dimension $d = 2$.	12
3.3.1	Discrétisation en espace	13
3.3.2	Discrétisation en temps	13
4	Résultats dimension 2	16
4.1	PME et condition initiale Barenblatt : solution exacte	16
4.2	Condition initiale : densité loi normale	18
4.2.1	PME (PME_NORMAL_01.pde)	18
4.2.2	HEAT (HEAT_NORMAL_01.pde)	20
4.2.3	HEAV (HEAV_NORMAL_01.pde)	22
4.2.4	Comparaison : HEAT/HEAV	24
4.3	Condition initiale bimodale	26
4.3.1	PME (Init_PME_BIN0_02.pde)	26
4.3.2	HEAT (Init_HEAT_BIN0_02.pde)	28
4.3.3	HEAV (Init_HEAV_BIN0_02.pde)	30
4.3.4	Comparaison : HEAT/HEAV	32

1

1 Résolution d'un système raide d'EDO

Soit le système raide d'EDO

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t)) + \frac{1}{\varepsilon} \mathbf{g}(t, \mathbf{y}(t)) \quad (1.1)$$

où $\mathbf{y} : \mathbb{R}^+ \mapsto \mathbb{R}^d$, $\mathbf{f}, \mathbf{g} : \mathbb{R}^+ \times \mathbb{R}^d \mapsto \mathbb{R}^d$ et ε le paramètre de raideur.

1.1 Schémas IMEX Runge-Kutta (IMEX-RK)

Les schémas IMplicites-EXplicites de Runge-Kutta pour le système (1.1) sont de la forme

$$\mathbf{y}^{n+1} = \mathbf{y}^n + h \sum_{k=1}^{\nu} \tilde{b}_k \mathbf{f}(t^n + \tilde{c}_k h, \mathbf{Y}^{(k)}) + h \sum_{k=1}^{\nu} b_k \frac{1}{\varepsilon} \mathbf{g}(t^n + c_k h, \mathbf{Y}^{(k)}) \quad (1.2)$$

avec, pour $k \in \llbracket 1, \nu \rrbracket$

$$\mathbf{Y}^{(k)} = \mathbf{y}^n + h \sum_{l=1}^{k-1} \tilde{a}_{kl} \mathbf{f}(t^n + \tilde{c}_l h, \mathbf{Y}^{(l)}) + h \sum_{l=1}^{\nu} a_{k,l} \frac{1}{\varepsilon} \mathbf{g}(t^n + c_l h, \mathbf{Y}^{(l)}) \quad (1.3)$$

Les matrices $\tilde{\mathbb{A}}$ et \mathbb{A} sont des matrices $\nu \times \nu$ de composantes respectives \tilde{a}_{ij} et $a_{i,j}$. De plus $\tilde{a}_{ij} = 0$ si $j \geq i$. La notation de Butcher sous la forme d'un double tableau est la suivante

$$\begin{array}{c|c} \tilde{\mathbf{c}} & \tilde{\mathbb{A}} \\ \hline & \tilde{\mathbf{b}}^t \end{array} \quad \text{et} \quad \begin{array}{c|c} \mathbf{c} & \mathbb{A} \\ \hline & \mathbf{b}^t \end{array} \quad (1.4)$$

1.2 Les schémas SSP

Schéma SSP (Strong Stability Preserving)

1.2.1 SSP2

- Schéma SSP2(2,2,2)

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array} \quad \text{et} \quad \begin{array}{c|cc} \gamma & \gamma & 0 \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & 1/2 & 1/2 \end{array} \quad \text{avec } \gamma = 1 - \frac{1}{\sqrt{2}} \quad (1.5)$$

- Schéma SSP2(3,2,2)

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \hline & 0 & 1/2 & 1/2 \end{array} \quad \text{et} \quad \begin{array}{c|ccc} 1/2 & 1/2 & 0 & 0 \\ 0 & -1/2 & 1/2 & 0 \\ 1 & 0 & 1/2 & 1/2 \\ \hline & 0 & 1/2 & 1/2 \end{array} \quad (1.6)$$

- Schéma SSP2(3,3,2)

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 1/2 & 1/2 & 0 \\ \hline & 1/3 & 1/3 & 1/3 \end{array} \quad \text{et} \quad \begin{array}{c|ccc} 1/4 & 1/4 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 \\ 1 & 1/3 & 1/3 & 1/3 \\ \hline & 1/3 & 1/3 & 1/3 \end{array} \quad (1.7)$$

1.2.2 SSP3

- Schéma SSP3(3,3,2)

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ \hline & 1/6 & 1/6 & 2/3 \end{array} \quad \text{et} \quad \begin{array}{c|ccc} \gamma & \gamma & 0 & 0 \\ 1-\gamma & 1-2\gamma & \gamma & 0 \\ 1/2 & 1/2-\gamma & 0 & \gamma \\ \hline & 1/6 & 1/6 & 2/3 \end{array} \quad \text{avec } \gamma = 1 - \frac{1}{\sqrt{2}} \quad (1.8)$$

- Schéma SSP3(4,3,3)

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1/2 & 0 & 1/4 & 1/4 & 0 \\ \hline & 0 & 1/6 & 1/6 & 2/3 \end{array} \quad \text{et} \quad \begin{array}{c|cccc} \alpha & \alpha & 0 & 0 & 0 \\ 0 & -\alpha & \alpha & 0 & 0 \\ 1 & 0 & 1-\alpha & \alpha & 0 \\ 1/2 & \beta & \eta & 1/2-\beta-\eta-\alpha & \alpha \\ \hline & 0 & 1/6 & 1/6 & 2/3 \end{array} \quad (1.9)$$

avec $\alpha = 0.24169426078821$, $\beta = 0.06042356519705$ et $\eta = 0.12915286960590$.

2 Résolution de $\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} (f(u)) = g(u)$

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} (f(u)) = g(u), \quad \forall t \geq 0, \forall x \in [a; b] \\ u(t = 0, x) = u_0(x), \quad \forall x \in [a; b] \\ u(t, x = a) = u_a(t), \quad \forall t \geq 0 \\ u(t, x = b) = u_b(t), \quad \forall t \geq 0 \end{array} \right. \quad (2.1)$$

$$u(t = 0, x) = u_0(x), \quad \forall x \in [a; b] \quad (2.2)$$

$$u(t, x = a) = u_a(t), \quad \forall t \geq 0 \quad (2.3)$$

$$u(t, x = b) = u_b(t), \quad \forall t \geq 0 \quad (2.4)$$

Le flux $f : \mathbb{R} \mapsto \mathbb{R}$ est donné.

2.1 Discrétisation en espace

Soit $\Delta x = \frac{b-a}{N_x}$, $x_i = a + (i - 1/2)\Delta x$, $\forall i \in \llbracket 1, N_x \rrbracket$ et $x_{i+1/2} = x_i + \frac{\Delta x}{2}$, $\forall i \in \llbracket 0, N_x \rrbracket$.

Pour écrire un schéma conservatif, on introduit la fonction \hat{F} telle que

$$f(u(t, x)) = \frac{1}{\Delta x} \int_{x-\Delta x/2}^{x+\Delta x/2} \hat{F}(t, s) ds$$

On a alors

$$\frac{\partial f}{\partial x}(u(t, x_i)) = \frac{1}{\Delta x} (\hat{F}(t, x_{i+1/2}) - \hat{F}(t, x_{i-1/2}))$$

On note $F_{i+1/2}(t)$ un flux semi-discret approchant $\hat{F}(t, x_{i+1/2})$ et $u_i(t) = u(t, x_i)$.

Le schéma discrétisé en espace s'écrit donc sous la forme d'un système d'EDO :

$$\left\{ \begin{array}{l} \frac{du_i}{dt}(t) + \frac{1}{\Delta x} (F_{i+1/2}(t) - F_{i-1/2}(t)) = g(u_i(t)), \quad \forall t \geq 0, \forall i \in \llbracket 1, N_x \rrbracket \\ u(t = 0, x_i) = u_0(x_i), \quad \forall i \in \llbracket 1, N_x \rrbracket \\ u(t, x = a) = u_a(t), \quad \forall t \geq 0 \\ u(t, x = b) = u_b(t), \quad \forall t \geq 0 \end{array} \right. \quad (2.5)$$

$$u(t = 0, x_i) = u_0(x_i), \quad \forall i \in \llbracket 1, N_x \rrbracket \quad (2.6)$$

$$u(t, x = a) = u_a(t), \quad \forall t \geq 0 \quad (2.7)$$

$$u(t, x = b) = u_b(t), \quad \forall t \geq 0 \quad (2.8)$$

2.2 Calcul du flux

Pour calculer le flux $F_{i+1/2}(t)$ on introduit la fonction \mathcal{F} et on écrit

$$F_{i+1/2}(t) = \mathcal{F} \left[u_{i+1/2}^-(t), u_{i+1/2}^+(t) \right]$$

où les $u_{i+1/2}^+(t)$ sont calculés à partir du polynôme d'interpolation sur les intervalles $[x_{i+1/2}; x_{i+3/2}]$ avec un stencil de ENO et $u_{i+1/2}^-(t)$ sont calculés à partir du polynôme d'interpolation sur les intervalles $[x_{i-1/2}; x_{i+1/2}]$ avec un stencil de ENO.

Avec les notations de l'annexe ??,

$$\left\{ \begin{array}{l} u_{i+1/2}^+(t) = \sum_{j=0}^d \mathbf{C}_{\mathbf{r}(i)+1, j+1} u_{i-\mathbf{r}(i)+j}(t) \\ u_{i+1/2}^-(t) = \sum_{j=0}^d \mathbf{C}_{\mathbf{r}(i)+2, j+1} u_{i-\mathbf{r}(i)+j}(t) \end{array} \right. \quad (2.9)$$

$$u_{i+1/2}^-(t) = \sum_{j=0}^d \mathbf{C}_{\mathbf{r}(i)+2, j+1} u_{i-\mathbf{r}(i)+j}(t) \quad (2.10)$$

où $d - 1$ est le degré du polynôme de reconstruction et $\mathbf{r}(i)$, la valeur de r pour le stencil de ENO associé au point x_i .

Pour minimiser la viscosité numérique on choisit de prendre comme flux numérique, le flux de Godunov :

$$\mathcal{F}_G[u, v] = \begin{cases} \min_{u \leq \xi \leq v} f(\xi), & \text{si } u \leq v \\ \max_{v \leq \xi \leq u} f(\xi), & \text{si } v \leq u \end{cases} \quad (2.11)$$

On a donc le schéma semi discrétisé

$$\frac{du_i}{dt}(t) + \frac{1}{\Delta x} \left(\mathcal{F}_G \left[u_{i+1/2}^-(t), u_{i+1/2}^+(t) \right] - \mathcal{F}_G \left[u_{i-1/2}^-(t), u_{i-1/2}^+(t) \right] \right) = g(u_i(t)) \quad (2.12)$$

2.3 Discrétisation en temps pour $f(u) = \alpha u$, $\alpha > 0$: schémas IMEX-RK

Dans ce cas, le flux de Godunov devient

$$\mathcal{F}_G[u, v] = \alpha u$$

et le schéma semi-discret devient

$$\left\{ \begin{array}{l} \frac{du_i}{dt}(t) + \frac{\alpha}{\Delta x} \left(u_{i+1/2}^-(t) - u_{i-1/2}^-(t) \right) = g(u_i(t)), \quad \forall t \geq 0, \quad \forall i \in \llbracket 1, N_x \rrbracket \\ u_i(0) = u_0(x_i), \quad \forall i \in \llbracket 1, N_x \rrbracket \\ u(t, x = a) = u_a(t), \quad \forall t \geq 0 \\ u(t, x = b) = u_b(t), \quad \forall t \geq 0 \end{array} \right. \quad (2.13)$$

$$u_i(0) = u_0(x_i), \quad \forall i \in \llbracket 1, N_x \rrbracket \quad (2.14)$$

$$u(t, x = a) = u_a(t), \quad \forall t \geq 0 \quad (2.15)$$

$$u(t, x = b) = u_b(t), \quad \forall t \geq 0 \quad (2.16)$$

On utilise alors des schémas IMEX Runge-Kutta pour la discrétisation en temps. On note $\Delta t = \frac{T}{N_t}$, $t^n = n\Delta t$, $\forall n \in \llbracket 0, N_t \rrbracket$ et $u_i^n \approx u(t^n, x_i)$. A partir de (1.2-1.3), on a

$$u_i^{n+1} = u_i^n + \Delta t \sum_{k=1}^{\nu} \tilde{b}_k F_i^{(k)} \quad (2.17)$$

avec $\forall k \in \llbracket 1, \nu \rrbracket$

$$\left\{ \begin{array}{l} F_i^{(k)} = g(u_i^{(k)}) - \frac{\alpha}{\Delta x} (u_{i+1/2}^{(k)-} - u_{i-1/2}^{(k)-}) \\ u_i^{(k)} = u_i^n + \Delta t \sum_{l=1}^{k-1} \tilde{a}_{kl} F_i^{(l)} \end{array} \right. \quad (2.18)$$

$$u_i^{(k)} = u_i^n + \Delta t \sum_{l=1}^{k-1} \tilde{a}_{kl} F_i^{(l)} \quad (2.19)$$

et les $u_{i+1/2}^{(k)-}$ sont calculés en utilisant les polynômes d'interpolation de degré $d - 1$ (??) et les conditions aux limites :

$$\begin{aligned} u_{1/2}^{(k)-} &= u_a(t^n + \tilde{c}_k \Delta t) \\ u_{i+1/2}^{(k)-} &= \sum_{j=0}^d u_{i-\mathbf{r}(i)+j}^n \mathbf{C}_{\mathbf{r}(i)+2, j+1}, \quad \forall i \in \llbracket 1, N_x - 1 \rrbracket \\ u_{N_x+1/2}^{(k)-} &= u_b(t^n + \tilde{c}_k \Delta t) \end{aligned}$$

Algorithm 2.1 Résolution de $\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = g(u)$, $\alpha > 0$

Utilisation du symbolisme Matlab par soucis de simplification de l'écriture.

La fonction FONCLAGRANGEM est décrite par l'algorithme ??.

Données : α : $\alpha > 0$
 d : polynôme d'interpolation de degré $d - 1$.
 X_m : tableau des N_x points x_i

Résultat : U : matrice de dimension $N_x \times (Nt + 1)$
 $U(i, n) \approx u(t^n; x_i)$

```

1: C ← COEFSTENCILSFONC(d)
2: U(:, 1) ← u0(Xm)
3: F ← zeros(Nx, ν)
4: for n ← 1 to Nt do
5:   S0 ← 0 ▷ Vecteur de dimension Nx
6:   for k ← 1 to ν do
7:     S1 ← 0 ▷ Vecteur de dimension Nx
8:     for l ← 1 to k - 1 do
9:       S1 ← S1 +  $\tilde{a}_{k,l}$  * F(:, l)
10:    end for
11:    U(k) ← U(:, n) + Δt * S1
12:    [U(k)+, U(k)-] ← FONCLAGRANGEM(U(k), d, C)
13:    U(k)- ← [ua(tn +  $\tilde{c}_k$ Δt); U(k)-] ▷ Prise en compte de la C.L. en a
14:    U(k)-(Nx + 1) ← ub(tn +  $\tilde{c}_k$ Δt) ▷ Prise en compte de la C.L. en b
15:    F(:, k) ← g(U(k)) -  $\frac{\alpha}{\Delta x}$  (U(k)-(2 : Nx + 1) - U(k)-(1 : Nx))
16:    S0 ← S0 +  $\tilde{b}_k$  * F(:, k)
17:  end for
18:  U(:, n + 1) ← U(:, n) + Δt * S0
19: end for

```

2.4 Discrétisation en temps pour $f(u) = \alpha u$, $\alpha < 0$: schémas IMEX-RK

Dans ce cas, le flux de Godunov devient

$$\mathcal{F}_G[u, v] = \alpha v$$

et le schéma semi-discret devient

$$\left\{ \begin{array}{l} \frac{du_i}{dt}(t) + \frac{\alpha}{\Delta x} (u_{i+1/2}^+(t) - u_{i-1/2}^+(t)) = g(u_i(t)), \quad \forall t \geq 0, \forall i \in \llbracket 1, N_x \rrbracket \\ u_i(0) = u_0(x_i), \quad \forall i \in \llbracket 1, N_x \rrbracket \\ u(t, x = a) = u_a(t), \quad \forall t \geq 0 \\ u(t, x = b) = u_b(t), \quad \forall t \geq 0 \end{array} \right. \quad (2.20)$$

$$u_i(0) = u_0(x_i), \quad \forall i \in \llbracket 1, N_x \rrbracket \quad (2.21)$$

$$u(t, x = a) = u_a(t), \quad \forall t \geq 0 \quad (2.22)$$

$$u(t, x = b) = u_b(t), \quad \forall t \geq 0 \quad (2.23)$$

A partir de (1.2-1.3) et des notations de la section précédente, le schéma IMEX-RK s'écrit sous la forme

$$u_i^{n+1} = u_i^n + \Delta t \sum_{k=1}^{\nu} \tilde{b}_k F_i^{(k)} \quad (2.24)$$

avec $\forall k \in \llbracket 1, \nu \rrbracket$

$$\left\{ \begin{array}{l} F_i^{(k)} = g(u_i^{(k)}) - \frac{\alpha}{\Delta x} (u_{i+1/2}^{(k)+} - u_{i-1/2}^{(k)+}) \\ u_i^{(k)} = u_i^n + \Delta t \sum_{l=1}^{k-1} \tilde{a}_{kl} F_i^{(l)} \end{array} \right. \quad (2.25)$$

$$\left\{ \begin{array}{l} F_i^{(k)} = g(u_i^{(k)}) - \frac{\alpha}{\Delta x} (u_{i+1/2}^{(k)+} - u_{i-1/2}^{(k)+}) \\ u_i^{(k)} = u_i^n + \Delta t \sum_{l=1}^{k-1} \tilde{a}_{kl} F_i^{(l)} \end{array} \right. \quad (2.26)$$

et les $u_{i+1/2}^{(k)+}$ sont calculés en utilisant les polynômes d'interpolation de degré $d - 1$ (??) et les conditions aux limites :

$$\begin{aligned} u_{1/2}^{(k)+} &= u_a(t^n + \tilde{c}_k \Delta t) \\ u_{i+1/2}^{(k)+} &= \sum_{j=0}^d u_{i-\mathbf{r}(i)+j}^n \mathbf{C}_{\mathbf{r}(i)+1,j+1}, \quad \forall i \in \llbracket 1, N_x - 1 \rrbracket \\ u_{N_x+1/2}^{(k)+} &= u_b(t^n + \tilde{c}_k \Delta t) \end{aligned}$$

Algorithm 2.2 Résolution de $\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = g(u)$, $\alpha < 0$

Utilisation du symbolisme Matlab par soucis de simplification de l'écriture.

La fonction FONCLAGRANGEPM est décrite par l'algorithme ??.

Données : α : $\alpha < 0$
 d : polynôme d'interpolation de degré $d - 1$.
 X_m : tableau des N_x points x_i

Résultat : U : matrice de dimension $N_x \times (N_t + 1)$
 $U(i, n) \approx u(t^n; x_i)$

```

1: C ← COEFSTENCILSFONC(d)
2: U(:, 1) ← u_0(X_m)
3: F ← zeros(N_x, nu)
4: for n ← 1 to N_t do
5:   S_0 ← 0                                     ▷ Vecteur de dimension N_x
6:   for k ← 1 to nu do
7:     S_1 ← 0                                     ▷ Vecteur de dimension N_x
8:     for l ← 1 to k - 1 do
9:       S_1 ← S_1 + a_{k,l} * F(:, l)
10:    end for
11:    U^{(k)} ← U(:, n) + Δt * S_1
12:    [U^{(k)+}, U^{(k)-}] ← FONCLAGRANGEPM(U^{(k)}, d, C)
13:    U^{(k)+} ← [U^{(k)+}; u_b(t^n + c_k Δt)]
14:    U^{(k)+}(1) ← u_a(t^n + c_k Δt)             ▷ Prise en compte de la C.L. en a
15:    F(:, k) ← g(U^{(k)}) - α/Δx (U^{(k)+}(2 : N_x + 1) - U^{(k)+}(1 : N_x))
16:    S_0 ← S_0 + b_k * F(:, k)                 ▷ Prise en compte de la C.L. en b
17:  end for
18:  U(:, n + 1) ← U(:, n) + Δt * S_0
19: end for

```

3 Résolution de $\frac{\partial u}{\partial t} - D\Delta(p(u)) = g(u) + f$

On veut résoudre l'EDP suivante

$$\frac{\partial u}{\partial t} - D\Delta(p(u)) = g(u) + f, \quad x \in \mathbb{R}^d, t > 0, \quad (3.1)$$

avec la condition initiale $u(x, 0) = u_0(x)$, $u_0 \in L^1(\mathbb{R}^d)$, $D > 0$ qui est le coefficient de diffusion et $f : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}$ une fonction donnée.

On suppose que $p : \mathbb{R} \rightarrow \mathbb{R}$ est non décroissante et Lipschitz continuos.

3.1 Transformation du problème

Nous rappelons ici les grandes lignes (voir [?],[?])

On introduit une fonction additionnelle $\mathbf{v} : \mathbb{R}^+ \times \mathbb{R}^d \mapsto \mathbb{R}^d$ et un paramètre positif ε pour obtenir le système de relaxation suivant

$$\begin{cases} \frac{\partial u}{\partial t} + \operatorname{div} \mathbf{v} & = g(u) + f \\ \frac{\partial \mathbf{v}}{\partial t} + \frac{D}{\varepsilon} \nabla(p(u)) & = -\frac{1}{\varepsilon} \mathbf{v} \end{cases} \quad (3.2)$$

Le paramètre φ est introduit pour obtenir des vitesses caractéristiques finies et pour ne pas avoir un opérateur différentiel singulier quand $\varepsilon \rightarrow 0^+$.

$$\begin{cases} \frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{v}) & = g(u) + f \\ \frac{\partial \mathbf{v}}{\partial t} + \varphi^2 \nabla(p(u)) & = -\frac{1}{\varepsilon} \mathbf{v} + \left(\varphi^2 - \frac{D}{\varepsilon}\right) \nabla(p(u)) \end{cases} \quad (3.3)$$

Enfin, le terme non-linéaire dans le terme de convection est supprimé en utilisant une technique classique dans les schémas de relaxation : on introduit une fonction $w : \mathbb{R}^+ \times \mathbb{R}^d \mapsto \mathbb{R}$ et on obtient

$$\begin{cases} \frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{v}) = g(u) + f & (3.4) \\ \frac{\partial \mathbf{v}}{\partial t} + \varphi^2 \nabla(w) = -\frac{1}{\varepsilon} \mathbf{v} + \left(\varphi^2 - \frac{D}{\varepsilon}\right) \nabla(w) & (3.5) \\ \frac{\partial w}{\partial t} + \operatorname{div}(\mathbf{v}) = -\frac{1}{\varepsilon}(w - p(u)) & (3.6) \end{cases}$$

Formellement, quand $\varepsilon \rightarrow 0^+$, on a $w \rightarrow p(u)$ et $\mathbf{v} \rightarrow -D \nabla(p(u))$ et l'on retrouve (3.1).

3.2 En dimension $d = 1$.

Le problème modèle à résoudre est le suivant :

$$\begin{cases} \frac{\partial u}{\partial t}(t; x) - D \frac{\partial^2}{\partial x^2}(p(u(t; x))) = g(u(t; x)) + f(t; x), \quad \forall t > 0, \quad \forall x \in]a; b[& (3.7) \\ u(0; x) = u_0(x), \quad \forall x \in]a; b[& (3.8) \\ u(t; a) = u_a(t), \quad \forall t > 0 & (3.9) \\ u(t; b) = u_b(t), \quad \forall t > 0 & (3.10) \end{cases}$$

Après transformation de (3.7) avec les techniques décrites en section 3.1, on obtient

$$\frac{\partial \mathbf{z}}{\partial t} + \frac{\partial \mathcal{F}(\mathbf{z})}{\partial x} = \mathbf{g}(\mathbf{z}) + \frac{1}{\varepsilon} \mathbf{h}(\mathbf{z}) \quad (3.11)$$

avec

$$\mathbf{z} = \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \quad \mathcal{F}(\mathbf{z}) = \mathbb{A} \mathbf{z}, \quad \mathbf{h}(\mathbf{z}) = \begin{pmatrix} 0 \\ -v + (\varphi^2 \varepsilon - D) \frac{\partial w}{\partial x} \\ p(u) - w \end{pmatrix}, \quad \mathbf{g}(\mathbf{z}) = \begin{pmatrix} g(u) + f \\ 0 \\ 0 \end{pmatrix} \quad (3.12)$$

et

$$\mathbb{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & \varphi^2 \\ 0 & 1 & 0 \end{pmatrix}.$$

A l'aide d'un changement de variables (variables caractéristiques), on va montrer que (3.11) revient à la résolution de 3 problèmes d'advection avec vitesse d'advection positive, négative et nulle.

$$\det(\mathbb{A} - \lambda \mathbb{I}) = -\lambda(\lambda^2 - \varphi^2)$$

Les valeurs propres de \mathbb{A} sont $(\varphi, -\varphi, 0)$. Les vecteurs propres associés sont respectivement

$$\begin{pmatrix} 1 \\ \varphi \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -\varphi \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

On note

$$\mathbb{P} = \begin{pmatrix} 1 & 1 & 1 \\ \varphi & -\varphi & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

On a alors

$$\mathbb{P}^{-1} = \begin{pmatrix} 0 & \frac{1}{2\varphi} & \frac{1}{2} \\ 0 & -\frac{1}{2\varphi} & \frac{1}{2} \\ 1 & 0 & -1 \end{pmatrix}$$

et

$$\mathbb{P}^{-1}\mathbb{A}\mathbb{P} = \mathbb{D} = \begin{pmatrix} \varphi & 0 & 0 \\ 0 & -\varphi & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

On remarque immédiatement la propriété suivante de la matrice \mathbb{P}^{-1}

Propriété 1 soit $\mathbf{U} \in \mathbb{R}^3$ et $\mathbf{V} = \mathbb{P}^{-1}\mathbf{U}$ alors

$$\sum_{i=1}^3 V_i = U_1$$

On pose $\mathbf{Z} = \mathbb{P}^{-1}\mathbf{z}$ et on obtient

$$\mathbf{z} = \begin{pmatrix} \mathcal{U} \\ \mathcal{V} \\ \mathcal{W} \end{pmatrix} = \begin{pmatrix} \frac{v + \varphi w}{2\varphi} \\ -v + \varphi w \\ \frac{2\varphi}{u - w} \end{pmatrix}. \quad (3.13)$$

On remarque que l'on a

$$u = \mathcal{U} + \mathcal{V} + \mathcal{W}. \quad (3.14)$$

Comme $\mathbf{z} = \mathbb{P}\mathbf{Z}$, l'équation (??) devient

$$\mathbb{P} \frac{\partial \mathbf{Z}}{\partial t} + \mathbb{A}\mathbb{P} \frac{\partial \mathbf{Z}}{\partial x} = \mathbf{g}(\mathbb{P}\mathbf{Z}) + \frac{1}{\varepsilon} \mathbf{h}(\mathbb{P}\mathbf{Z})$$

et en multipliant à gauche par \mathbb{P}^{-1} , on obtient

$$\frac{\partial \mathbf{Z}}{\partial t} + \mathbb{D} \frac{\partial \mathbf{Z}}{\partial x} = \mathbb{P}^{-1} \mathbf{g}(\mathbb{P}\mathbf{Z}) + \frac{1}{\varepsilon} \mathbb{P}^{-1} \mathbf{h}(\mathbb{P}\mathbf{Z}) \quad (3.15)$$

ou encore

$$\begin{pmatrix} \frac{\partial \mathcal{U}}{\partial t} + \varphi \frac{\partial \mathcal{U}}{\partial x} \\ \frac{\partial \mathcal{V}}{\partial t} - \varphi \frac{\partial \mathcal{V}}{\partial x} \\ \frac{\partial \mathcal{W}}{\partial t} \end{pmatrix} = \mathbb{P}^{-1} \mathbf{g}(\mathbb{P}\mathbf{Z}) + \frac{1}{\varepsilon} \mathbb{P}^{-1} \mathbf{h}(\mathbb{P}\mathbf{Z}) \quad (3.16)$$

3.2.1 Discrétisation en espace

Nous devons discrétiser le système d'équation suivant :

$$\begin{cases} \frac{\partial \mathbf{Z}}{\partial t}(t; x) + \mathbb{D} \frac{\partial \mathbf{Z}}{\partial x}(t; x) = \mathbf{g}(\mathbf{Z}(t; x)), & \forall t > 0, \forall x \in]a; b[\\ \mathbf{Z}(0; x) = \mathbf{Z}_0(x) & \forall x \in]a; b[\\ + C.L. \end{cases} \quad (3.17)$$

$$(3.18)$$

avec

$$\mathbf{g}(\mathbf{Z}) = \mathbb{P}^{-1} \mathbf{g}(\mathbb{P}\mathbf{Z}) + \frac{1}{\varepsilon} \mathbb{P}^{-1} \mathbf{h}(\mathbb{P}\mathbf{Z})$$

$$\mathbf{Z}_0(x) = \begin{pmatrix} \frac{v_0(x) + \varphi w_0(x)}{2\varphi} \\ \frac{-v_0(x) + \varphi w_0(x)}{2\varphi} \\ u_0(x) - w_0(x) \end{pmatrix}, \quad v_0(x) = -D \frac{\partial}{\partial x} (p(u_0(x))) \quad \text{et} \quad w_0(x) = p(u_0(x))$$

Soient $\Delta x = \frac{b-a}{N_x}$ et $x_i = a + (i - \frac{1}{2})\Delta x$, $\forall i \in \llbracket 1, N_x \rrbracket$. On note $\mathbf{Z}_i(t) = \mathbf{Z}(t; x_i)$,
 Pour l'instant les conditions aux limites ne sont pas présent en compte (A FAIRE)
 Du système d'équations (3.17), on obtient

$$\frac{\partial \mathbf{Z}}{\partial t}(t; x_i) + \mathbb{D} \frac{\partial \mathbf{Z}}{\partial x}(t; x_i) = \mathbf{g}(\mathbf{Z}(t; x_i)), \quad \forall t > 0, \quad \forall i \in \llbracket 1, N_x \rrbracket \quad (3.19)$$

ou encore, $\forall i \in \llbracket 1, N_x \rrbracket$,

$$\begin{cases} \frac{\partial \mathcal{U}}{\partial t}(t; x_i) + \varphi \frac{\partial \mathcal{U}}{\partial x}(t; x_i) = \mathcal{G}_1(\mathbf{Z}(t; x_i)), \end{cases} \quad (3.20)$$

$$\begin{cases} \frac{\partial \mathcal{V}}{\partial t}(t; x_i) - \varphi \frac{\partial \mathcal{V}}{\partial x}(t; x_i) = \mathcal{G}_2(\mathbf{Z}(t; x_i)), \end{cases} \quad (3.21)$$

$$\begin{cases} \frac{\partial \mathcal{W}}{\partial t}(t; x_i) = \mathcal{G}_3(\mathbf{Z}(t; x_i)). \end{cases} \quad (3.22)$$

On peut maintenant utiliser les techniques décrites en ???. Comme $\varphi > 0$, on utilise le schéma (??) pour l'équation (3.20) et le schéma (??) pour l'équation (3.21). On obtient donc le système d'EDO suivant

$$\begin{cases} \frac{d\mathcal{U}_i}{dt}(t) + \frac{\varphi}{\Delta x} (\mathcal{U}_{i+1/2}^-(t) - \mathcal{U}_{i-1/2}^-(t)) = \mathcal{G}_1(\mathbf{Z}_i(t)), \end{cases} \quad (3.23)$$

$$\begin{cases} \frac{d\mathcal{V}_i}{dt}(t) - \frac{\varphi}{\Delta x} (\mathcal{V}_{i+1/2}^+(t) - \mathcal{V}_{i-1/2}^+(t)) = \mathcal{G}_2(\mathbf{Z}_i(t)), \end{cases} \quad (3.24)$$

$$\begin{cases} \frac{d\mathcal{W}_i}{dt}(t) = \mathcal{G}_3(\mathbf{Z}_i(t)), \end{cases} \quad (3.25)$$

En sommant ces trois équations et en utilisant (3.14), on a

$$\frac{du_i}{dt}(t) + \frac{\varphi}{\Delta x} (\mathcal{U}_{i+1/2}^-(t) - \mathcal{U}_{i-1/2}^-(t) - (\mathcal{V}_{i+1/2}^+(t) - \mathcal{V}_{i-1/2}^+(t))) = \sum_{k=1}^3 \mathcal{G}_k(\mathbf{Z}_i(t))$$

De part la propriété 1, on a

$$\sum_{k=1}^3 \mathcal{G}_k(\mathbf{Z}_i(t)) = g(u_i(t)) + f(t; x_i)$$

Nous sommes donc ramené à résoudre le système d'EDO, $\forall i \in \llbracket 1, N_x \rrbracket$

$$\begin{cases} \frac{du_i}{dt}(t) + \frac{\varphi}{\Delta x} \mathcal{F}_i(t) = g(u_i(t)) + f(t; x_i), \quad \forall t > 0 \end{cases} \quad (3.26)$$

$$u_i(0) = u_0(x_i) \quad (3.27)$$

avec

$$\mathcal{F}_i(t) = \mathcal{U}_{i+1/2}^-(t) - \mathcal{U}_{i-1/2}^-(t) - (\mathcal{V}_{i+1/2}^+(t) - \mathcal{V}_{i-1/2}^+(t))$$

3.2.2 Discrétisation en temps : schémas IMEX-RK

On note $u_i^n \approx u_i(t^n)$. Les schémas IMEX Runge-Kutta associés à (3.26-3.27) peuvent s'écrire sous la forme

$$u_i^{n+1} = u_i^n + \Delta t \sum_{k=1}^{\nu} \tilde{b}_k F_i^{(k)}, \quad \forall i \in \llbracket 1, N_x \rrbracket \quad (3.28)$$

avec $\forall k \in \llbracket 1, \nu \rrbracket$

$$\begin{cases} F_i^{(k)} = -\frac{\varphi}{\Delta x} \mathcal{F}_i^{(k)} + g(u_i^{(k)}) + f(t^n + \tilde{c}_k \Delta t; x_i) & (3.29) \\ u_i^{(k)} = u_i^n + \Delta t \sum_{l=1}^{k-1} \tilde{a}_{kl} F_i^{(l)} & (3.30) \\ \mathcal{F}_i^{(k)} = \mathcal{U}_{i+1/2}^{(k)-} - \mathcal{U}_{i-1/2}^{(k)-} - (\mathcal{V}_{i+1/2}^{(k)+} - \mathcal{V}_{i-1/2}^{(k)+}) & (3.31) \end{cases}$$

Il reste à décrire le calcul de $\mathcal{U}_{i+1/2}^{(k)-}$ et $\mathcal{V}_{i+1/2}^{(k)+}$ pour $i \in \llbracket 0, N_x \rrbracket$ à partir des $u_i^{(k)}$. On pose $w_i^{(k)} = p(u_i^{(k)})$ et $v_i^{(k)} = -D \hat{\nabla}_{\Delta x}^d w_i^{(k)}$ où $\hat{\nabla}_{\Delta x}^d$ note l'opérateur des différences centrées à d points. Par (3.13), on a

$$\mathcal{U}_i^{(k)} = \frac{v_i^{(k)} + \varphi w_i^{(k)}}{2\varphi} \text{ et } \mathcal{V}_i^{(k)} = \frac{-v_i^{(k)} + \varphi w_i^{(k)}}{2\varphi}.$$

On utilise alors des polynômes de reconstructions d'ordre s de type ENO ou WENO pour déterminer $\mathcal{U}_{i+1/2}^{(k)-}$, $\mathcal{U}_{i-1/2}^{(k)+}$ (resp. $\mathcal{V}_{i+1/2}^{(k)-}$, $\mathcal{V}_{i-1/2}^{(k)+}$) en fonction de $u_i^{(k)}$ (resp. $v_i^{(k)}$).

On écrit maintenant la fonction CALCULFLUX1D (vectorisée) permettant de calculer le vecteur $\mathcal{F}^{(k)} \in \mathbb{R}^{N_x}$ de composantes $\mathcal{F}_i^{(k)} = \mathcal{U}_{i+1/2}^{(k)-} - \mathcal{U}_{i-1/2}^{(k)-} - (\mathcal{V}_{i+1/2}^{(k)+} - \mathcal{V}_{i-1/2}^{(k)+})$ par la méthode décrite précédemment.

Algorithm 3.1 Calcul de $\mathcal{F}^{(k)}$ défini en (3.31)

Données : \mathbf{u} : vecteur de \mathbb{R}^{N_x} ,
 \mathbf{x} : vecteur de \mathbb{R}^{N_x} , (discrétisation)
 h : un réel strictement positif, (pas de discrétisation)
 u_a, u_b : condition aux limites,
 φ : un réel
 p : fonction
 D :

Résultat : \mathcal{F} : vecteur de \mathbb{R}^{N_x} ,

```

1: function  $\mathcal{F} \leftarrow$  CALCULFLUX1D( $\mathbf{u}, \mathbf{x}, \Delta x, h, u_a, u_b, \varphi, p, D$ )
2:    $\mathbf{w} \leftarrow p(\mathbf{u})$ 
3:    $\mathbf{v} \leftarrow -D * \text{CENTERDIFF}(\mathbf{w}, \mathbf{x}, h, p(u_a), p(u_b), 3)$ 
4:    $\mathcal{U} \leftarrow (\mathbf{v} + \varphi \mathbf{w}) / (2\varphi)$ 
5:    $\mathcal{V} \leftarrow (-\mathbf{v} + \varphi \mathbf{w}) / (2\varphi)$ 
6:    $[\mathcal{U}^+, \mathcal{U}^-] \leftarrow \text{WENO5}(\mathcal{U})$ 
7:    $[\mathcal{V}^+, \mathcal{V}^-] \leftarrow \text{WENO5}(\mathcal{V})$ 
8:    $\mathcal{U}^- \leftarrow [\mathcal{U}^+(1); \mathcal{U}^-]$ 
9:    $\mathcal{V}^+ \leftarrow [\mathcal{V}^+; \mathcal{V}^-(N)]$ 
10:   $Im \leftarrow 1 : N, Ip \leftarrow 2 : N + 1$ 
11:   $\mathcal{F} \leftarrow \mathcal{U}^-(Ip) - \mathcal{U}^-(Im) - (\mathcal{V}^+(Ip) - \mathcal{V}^+(Im))$ 
12: end function

```

Algorithm 3.2 Résolution de (??) par un schema IMEX Runge-Kutta de dimension ν

Données : Schéma IMEX : les vecteurs \mathbf{b} , \mathbf{c} , $\tilde{\mathbf{b}}$ et $\tilde{\mathbf{c}}$ de \mathbb{R}^ν ,
les matrices \mathbb{A} et $\tilde{\mathbb{A}}$ de $\mathcal{M}_{\nu,\nu}(\mathbb{R})$
 d : polynôme de reconstruction de fonction de degré $d - 1$.
 \tilde{d} : polynôme de reconstruction de dérivée de fonction de degré $\tilde{d} - 1$.

Résultat : $(\mathbf{u}^n)_{n=0}^{N_t}$: les $N_t + 1$ vecteurs de \mathbb{R}^{N_x} ,
avec $\mathbf{u}^n(x_i) \approx u(t^n, x_i)$. du moins on l'espère!

```

1:  $\mathbf{u}^0 \leftarrow u_0(\mathbf{x}_m)$ 
2: for  $n \leftarrow 0$  to  $N_t - 1$  do
3:    $\mathbf{u}^{n+1} \leftarrow \mathbf{u}^n$ 
4:   for  $k \leftarrow 1$  to  $\nu$  do
5:      $t^{(k)} \leftarrow t^n + \tilde{c}_k * \Delta t$ 
6:      $\mathbf{f}^{(k)} \leftarrow f(t^{(k)}, \mathbf{x}_m)$ 
7:      $\mathbf{F}^{(k)} \leftarrow \text{CALCULF1D}(\mathbf{F}^{(1:k-1)}, \mathbf{u}^n, \mathbf{f}^{(k)}, \Delta t, \Delta x, \mathbb{C}, \mathbb{D}, \tilde{\mathbb{A}})$ 
8:      $\mathbf{u}^{n+1} \leftarrow \mathbf{u}^{n+1} + \Delta t * b_k * \mathbf{F}^{(k)}$ 
9:   end for
10: end for

```

Algorithm 3.3 Calcul de $\mathbf{F}^{(k)}$ défini en (3.29)

Données : $\mathbf{F}^{(1:k-1)}$: $k - 1$ vecteurs de \mathbb{R}^{N_x} ,
 \mathbf{u}^n : vecteur de \mathbb{R}^{N_x} ,
 $\mathbf{f}^{(k)}$: vecteur de \mathbb{R}^{N_x} ,
 Δt : un réel strictement positif,
 Δx : un réel strictement positif,
 \mathbb{C} : la matrice de $\mathcal{M}_{d+1,d}(\mathbb{R})$,
 \mathbb{D} : la matrice de $\mathcal{M}_{\tilde{d},\tilde{d}}(\mathbb{R})$,
 $\tilde{\mathbb{A}}$: matrice de $\mathcal{M}_{\nu,\nu}(\mathbb{R})$,

Résultat : $\mathbf{F}^{(k)}$: vecteur de \mathbb{R}^{N_x} ,

```

1: function  $\mathbf{F}^{(k)} \leftarrow \text{CALCULF1D}(\mathbf{F}^{(1:k-1)}, \mathbf{u}^n, \mathbf{f}^{(k)}, \Delta t, \Delta x, \mathbb{C}, \mathbb{D}, \tilde{\mathbb{A}})$ 
2:    $\mathbf{u}^{(k)} \leftarrow \mathbf{u}^n$ 
3:   for  $l \leftarrow 1$  to  $k - 1$  do
4:      $\mathbf{u}^{(k)} \leftarrow \mathbf{u}^{(k)} + \Delta t * \tilde{\mathbb{A}}(k, l) * \mathbf{F}^{(l)}$ 
5:   end for
6:    $\mathbf{F}^{(k)} \leftarrow -\frac{\varphi}{\Delta x} \text{CALCULFLUX1D}(\mathbf{u}^{(k)}, \mathbf{x}, \dots)$ 
7:    $\mathbf{F}^{(k)} \leftarrow \mathbf{F}^{(k)} + g(\mathbf{u}^{(k)}) + \mathbf{f}^{(k)}$ 
8: end function

```

3.3 En dimension $d = 2$.

Soit Ω un ouvert borné de \mathbb{R}^2 de frontière Γ . Le problème modèle à résoudre est le suivant :

$$\begin{cases} \frac{\partial u}{\partial t} - D\Delta(p(u)) = g(u) + f, & \text{dans } \mathbb{R}^+ \times \Omega \\ u(0; \mathbf{x}) = u_0(\mathbf{x}), & \forall \mathbf{x} \in \Omega \\ u(t; \mathbf{x}) = u_\Gamma(\mathbf{x}), & \forall \mathbf{x} \in \Gamma \end{cases} \quad (3.32)$$

$$u(0; \mathbf{x}) = u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega \quad (3.33)$$

$$u(t; \mathbf{x}) = u_\Gamma(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma \quad (3.34)$$

Après transformation de (3.32) avec les techniques décrites en section 3.1, on obtient

$$\frac{\partial \mathbf{z}}{\partial t} + \frac{\partial \mathcal{F}^{(1)}(\mathbf{z})}{\partial x_1} + \frac{\partial \mathcal{F}^{(2)}(\mathbf{z})}{\partial x_2} = \mathbf{g}(\mathbf{z}) + \frac{1}{\varepsilon} \mathbf{h}(\mathbf{z}) \quad (3.35)$$

avec

$$\mathbf{z} = \begin{pmatrix} u \\ v^{(1)} \\ v^{(2)} \\ w \end{pmatrix}, \quad \mathcal{F}^{(\alpha)}(\mathbf{z}) = \mathbb{A}_\alpha \mathbf{z}, \quad \forall \alpha \in \llbracket 1, 2 \rrbracket, \quad \mathbf{h}(\mathbf{z}) = \begin{pmatrix} 0 \\ -v^{(1)} + (\varphi^2 \varepsilon - D) \frac{\partial w}{\partial x_1} \\ -v^{(2)} + (\varphi^2 \varepsilon - D) \frac{\partial w}{\partial x_2} \\ p(u) - w \end{pmatrix}, \quad \mathbf{g}(\mathbf{z}) = \begin{pmatrix} g(u) + f \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.36)$$

et

$$\mathbb{A}_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \varphi^2 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mathbb{A}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varphi^2 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Les valeurs propres des matrices \mathbb{A}_1 et \mathbb{A}_2 sont identiques : $(\varphi, -\varphi, 0, 0)$. On note \mathbb{P}_1 et \mathbb{P}_2 les matrices

$$\mathbb{P}_1 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ \varphi & -\varphi & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}, \quad \mathbb{P}_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \varphi & -\varphi & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

vérifiant $\mathbb{P}_\alpha^{-1} \mathbb{A}_\alpha \mathbb{P}_\alpha = \mathbb{D}$, $\forall \alpha \in \llbracket 1, 2 \rrbracket$, avec $\mathbb{D} = \text{diag}(\varphi, -\varphi, 0, 0)$. On a

$$\mathbb{P}_1^{-1} = \begin{pmatrix} 0 & \frac{1}{2\varphi} & 0 & \frac{1}{2} \\ 0 & -\frac{1}{2\varphi} & 0 & \frac{1}{2} \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbb{P}_2^{-1} = \begin{pmatrix} 0 & 0 & \frac{1}{2\varphi} & \frac{1}{2} \\ 0 & 0 & -\frac{1}{2\varphi} & \frac{1}{2} \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

On veut calculer $\frac{\partial \mathcal{F}^{(\alpha)}(\mathbf{z})}{\partial x_\alpha}$, $\forall \alpha \in \llbracket 1, 2 \rrbracket$. On note

$$\mathbf{Z}^{(\alpha)} = \begin{pmatrix} Z^{(\alpha,1)} \\ Z^{(\alpha,2)} \\ Z^{(\alpha,3)} \\ Z^{(\alpha,4)} \end{pmatrix} = \mathbb{P}_\alpha^{-1} \mathbf{z} = \begin{pmatrix} \frac{1}{2\varphi}(v^{(\alpha)} + \varphi w) \\ \frac{1}{2\varphi}(-v^{(\alpha)} + \varphi w) \\ u - w \\ v^{(\alpha \bmod 2+1)} \end{pmatrix}$$

On a alors

$$\mathbb{P}_\alpha^{-1} \mathcal{F}^{(\alpha)}(\mathbf{z}) = \mathbb{P}_\alpha^{-1} \mathbb{A}_\alpha \mathbb{P}_\alpha \mathbf{Z}^{(\alpha)} = \mathbb{D} \mathbf{Z}^{(\alpha)} = \begin{pmatrix} \varphi Z^{(\alpha,1)} \\ -\varphi Z^{(\alpha,2)} \\ 0 \\ 0 \end{pmatrix}.$$

3.3.1 Discrétisation en espace

On note $\mathbf{Z}_{i,j}^{(\alpha)} = \mathbf{Z}^{(\alpha)}(x_i, y_j)$. En utilisant le flux de Godunov, on a

$$\frac{\partial}{\partial x_1} (\mathbb{D}\mathbf{Z}_{i,j}^{(1)}) \approx \frac{\varphi}{\Delta_1} \begin{pmatrix} Z_{i+1/2,j}^{(1,1)-} - Z_{i-1/2,j}^{(1,1)-} \\ -(Z_{i+1/2,j}^{(1,2)+} - Z_{i-1/2,j}^{(1,2)+}) \\ 0 \\ 0 \end{pmatrix}$$

et

$$\frac{\partial}{\partial x_2} (\mathbb{D}\mathbf{Z}_{i,j}^{(2)}) \approx \frac{\varphi}{\Delta_2} \begin{pmatrix} Z_{i,j+1/2}^{(2,1)-} - Z_{i,j-1/2}^{(2,1)-} \\ -(Z_{i,j+1/2}^{(2,2)+} - Z_{i,j-1/2}^{(2,2)+}) \\ 0 \\ 0 \end{pmatrix}.$$

Comme $\mathcal{F}^{(\alpha)}(\mathbf{z}) = \mathbb{A}_\alpha \mathbf{z}$ et $\mathbf{z} = \mathbb{P}_\alpha Z^{(\alpha)}$, on en déduit

$$\mathbb{P}_\alpha^{-1} \mathcal{F}^{(\alpha)}(\mathbf{z}) = \mathbb{P}_\alpha^{-1} \mathbb{A}_\alpha \mathbb{P}_\alpha Z^{(\alpha)} = \mathbb{D}Z^{(\alpha)}.$$

Ce qui donne

$$\frac{\partial}{\partial x_\alpha} (\mathcal{F}^{(\alpha)}(\mathbf{z}_{i,j})) \approx \mathbb{P}_\alpha \frac{\partial}{\partial x_\alpha} (\mathbb{D}\mathbf{Z}_{i,j}^{(\alpha)}).$$

et donc

$$\frac{\partial}{\partial x_1} (\mathcal{F}^{(1)}(\mathbf{z}_{i,j})) + \frac{\partial}{\partial x_2} (\mathcal{F}^{(2)}(\mathbf{z}_{i,j})) \approx \frac{\varphi}{\Delta_1} \mathbb{P}_1 \begin{pmatrix} Z_{i+1/2,j}^{(1,1)-} - Z_{i-1/2,j}^{(1,1)-} \\ -(Z_{i+1/2,j}^{(1,2)+} - Z_{i-1/2,j}^{(1,2)+}) \\ 0 \\ 0 \end{pmatrix} + \frac{\varphi}{\Delta_2} \mathbb{P}_2 \begin{pmatrix} Z_{i,j+1/2}^{(2,1)-} - Z_{i,j-1/2}^{(2,1)-} \\ -(Z_{i,j+1/2}^{(2,2)+} - Z_{i,j-1/2}^{(2,2)+}) \\ 0 \\ 0 \end{pmatrix}$$

On obtient en prenant la première équation de (3.35)

$$\frac{\partial u_{i,j}}{\partial t}(t) + \frac{\varphi}{\Delta_1} \mathcal{F}_{i,j}^{(1)}(t) + \frac{\varphi}{\Delta_2} \mathcal{F}_{i,j}^{(2)}(t) = g(u_{i,j}(t)) + f_{i,j}(t) \quad (3.37)$$

avec

$$\mathcal{F}_{i,j}^{(1)}(t) = Z_{i+1/2,j}^{(1,1)-}(t) - Z_{i-1/2,j}^{(1,1)-}(t) - (Z_{i+1/2,j}^{(1,2)+}(t) - Z_{i-1/2,j}^{(1,2)+}(t)) \quad (3.38)$$

$$\mathcal{F}_{i,j}^{(2)}(t) = Z_{i,j+1/2}^{(2,1)-}(t) - Z_{i,j-1/2}^{(2,1)-}(t) - (Z_{i,j+1/2}^{(2,2)+}(t) - Z_{i,j-1/2}^{(2,2)+}(t)) \quad (3.39)$$

Pour simplifier les notations, on pose

$$\mathcal{U}_{i+1/2,j}^- = Z_{i+1/2,j}^{(1,1)-}(t), \quad \mathcal{V}_{i+1/2,j}^+ = Z_{i+1/2,j}^{(1,2)+}(t), \quad \mathcal{U}_{i,j+1/2}^- = Z_{i,j+1/2}^{(2,1)-}(t), \quad \mathcal{V}_{i,j+1/2}^+ = Z_{i,j+1/2}^{(2,2)+}(t).$$

3.3.2 Discrétisation en temps

On note $u_{i,j}^n \approx u_{i,j}(t^n)$. Les schémas IMEX Runge-Kutta associés à (3.26-3.27) peuvent s'écrire sous la forme

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t \sum_{k=1}^{\nu} \tilde{b}_k F_{i,j}^{(k)}, \quad \forall i \in \llbracket 1, N_x \rrbracket, \quad \forall j \in \llbracket 1, N_y \rrbracket \quad (3.40)$$

avec $\forall k \in \llbracket 1, \nu \rrbracket$

$$\begin{cases} u_{i,j}^{(k)} = u_{i,j}^n + \Delta t \sum_{l=1}^{k-1} \tilde{a}_{kl} F_{i,j}^{(l)} & (3.41) \\ F_{i,j}^{(k)} = -\frac{\varphi}{\Delta x_1} \mathcal{F}_{i,j}^{(1,k)} - \frac{\varphi}{\Delta x_2} \mathcal{F}_{i,j}^{(2,k)} + g(u_{i,j}^{(k)}) + f(t^n + \tilde{c}_k \Delta t; x_i, y_j) & (3.42) \\ \mathcal{F}_{i,j}^{(1,k)} = \mathcal{U}_{i+1/2,j}^{(k)-} - \mathcal{U}_{i-1/2,j}^{(k)-} - (\mathcal{V}_{i+1/2,j}^{(k)+} - \mathcal{V}_{i-1/2,j}^{(k)+}) & (3.43) \\ \mathcal{F}_{i,j}^{(2,k)} = \mathcal{U}_{i,j+1/2}^{(k)-} - \mathcal{U}_{i,j-1/2}^{(k)-} - (\mathcal{V}_{i,j+1/2}^{(k)+} - \mathcal{V}_{i,j-1/2}^{(k)+}) & (3.44) \end{cases}$$

A j fixé, le flux $\mathcal{F}_{:,j}^{(1,k)}$ revient au calcul 1D et on a

$$\mathcal{F}_{:,j}^{(1,k)} \leftarrow \text{CALCULFLUX1D}(\mathbf{u}_{:,j}^{(k)}; \mathbf{x}, \dots).$$

A i fixé, le flux $\mathcal{F}_{i,:}^{(2,k)}$ revient au calcul 1D et on a

$$\mathcal{F}_{i,:}^{(2,k)} \leftarrow \text{CALCULFLUX1D}(\mathbf{u}_{i,:}^{(k)}; \mathbf{y}, \dots).$$

Algorithm 3.4 Calcul de $\mathcal{F}_{i,j}^{(k)} = \frac{\varphi}{\Delta x_1} \mathcal{F}_{i,j}^{(1,k)} - \frac{\varphi}{\Delta x_2} \mathcal{F}_{i,j}^{(2,k)}$

Données : \mathbf{u} : matrice $N_x \times N_y$,
 \mathbf{x}_m : vecteur de \mathbb{R}^{N_x} , (discrétisation)
 h_x : un réel strictement positif, (pas de discrétisation)
 \mathbf{y}_m : vecteur de \mathbb{R}^{N_y} , (discrétisation)
 h_y : un réel strictement positif, (pas de discrétisation)
 $\mathbf{u}_{xa}, \mathbf{u}_{xb}$: vecteurs de \mathbb{R}^{N_x} (condition aux limites),
 $\mathbf{u}_{ya}, \mathbf{u}_{yb}$: vecteurs de \mathbb{R}^{N_y} (condition aux limites),
 φ : un réel
 p : fonction
 D :

Résultat : \mathcal{F} : matrice $N_x \times N_y$.

```

1: function  $\mathcal{F} \leftarrow \text{CALCULFLUX2D}(\mathbf{u}, \mathbf{u}_{xa}, \mathbf{u}_{xb}, \mathbf{u}_{ya}, \mathbf{u}_{yb}, \mathbf{x}, \Delta x, \mathbf{y}, \Delta y)$ 
2:   for  $j \leftarrow 1$  to  $N_y$  do
3:      $\mathcal{F}(:, j) \leftarrow -\frac{\varphi}{\Delta x} \text{CALCULFLUX1D}(\mathbf{u}_{:,j}, \mathbf{x}, \dots)$ 
4:   end for
5:   for  $i \leftarrow 1$  to  $N_x$  do
6:      $\mathcal{F}(i, :) \leftarrow \mathcal{F}(i, :) - \frac{\varphi}{\Delta y} \text{CALCULFLUX1D}(\mathbf{u}_{i,:}, \mathbf{y}, \dots)$ 
7:   end for
8: end function

```

Algorithm 3.5 Calcul de la matrice $\mathbf{F}^{(k)}$ défini en (3.42)

Données : $\mathbf{F}^{(1:k-1)}$: $k-1$ matrices de $N_x \times N_y$,
 \mathbf{u}^n : matrice de $N_x \times N_y$,
 $\mathbf{f}^{(k)}$: vecteur de $N_x \times N_y$,
 Δt : un réel strictement positif,
 Δx : un réel strictement positif,
 Δy : un réel strictement positif,
 \mathbb{C} : la matrice de $\mathcal{M}_{d+1,d}(\mathbb{R})$,
 \mathbb{D} : la matrice de $\mathcal{M}_{\bar{d},\bar{d}}(\mathbb{R})$,
 $\tilde{\mathbb{A}}$: matrice de $\mathcal{M}_{\nu,\nu}(\mathbb{R})$,

Résultat : $\mathbf{F}^{(k)}$: matrice de $N_x \times N_y$,

```

1: function  $\mathbf{F}^{(k)} \leftarrow \text{CALCULF2D}(\mathbf{F}^{(1:k-1)}, \mathbf{u}^n, \mathbf{f}^{(k)}, \Delta t, \Delta x, \Delta y, \mathbb{C}, \mathbb{D}, \tilde{\mathbb{A}})$ 
2:    $\mathbf{u}^{(k)} \leftarrow \mathbf{u}^n$ 
3:   for  $l \leftarrow 1$  to  $k-1$  do
4:      $\mathbf{u}^{(k)} \leftarrow \mathbf{u}^{(k)} + \Delta t * \tilde{\mathbb{A}}(k, l) * \mathbf{F}^{(l)}$ 
5:   end for
6:    $\mathbf{F}^{(k)} \leftarrow \text{CALCULFLUX2D}(\mathbf{u}^{(k)}, \dots) + g(\mathbf{u}^{(k)}) + \mathbf{f}^{(k)}$ 
7: end function

```

Algorithm 3.6 Résolution de (3.37) par un schema IMEX Runge-Kutta de dimension ν

Données : Schéma IMEX : les vecteurs \mathbf{b} , \mathbf{c} , $\tilde{\mathbf{b}}$ et $\tilde{\mathbf{c}}$ de \mathbb{R}^ν ,
les matrices \mathbb{A} et $\tilde{\mathbb{A}}$ de $\mathcal{M}_{\nu,\nu}(\mathbb{R})$
 d : polynôme de reconstruction de fonction de degré $d - 1$.
 \bar{d} : polynôme de reconstruction de dérivée de fonction de degré $\bar{d} - 1$.

Résultat : $(\mathbf{u}^n)_{n=0}^{N_t}$: les $N_t + 1$ matrices $N_x \times N_y$,
avec $\mathbf{u}_{i,j}^n \approx u(t^n, x_i, y_j)$.

```
1:  $\mathbf{u}^0 \leftarrow u_0(\mathbf{x}_m, \mathbf{y}_m)$ 
2: for  $n \leftarrow 0$  to  $N_t - 1$  do
3:    $\mathbf{u}^{n+1} \leftarrow \mathbf{u}^n$ 
4:   for  $k \leftarrow 1$  to  $\nu$  do
5:      $t^{(k)} \leftarrow t^n + \tilde{c}_k * \Delta t$ 
6:      $\mathbf{f}^{(k)} \leftarrow f(t^{(k)}, \mathbf{x}_m, \mathbf{y}_m)$ 
7:      $\mathbf{F}^{(k)} \leftarrow \text{CALCULF2D}(\mathbf{F}^{(1:k-1)}, \mathbf{u}^n, \mathbf{f}^{(k)}, \Delta t, \Delta x, \Delta y, \mathbb{C}, \mathbb{D}, \tilde{\mathbb{A}})$ 
8:      $\mathbf{u}^{n+1} \leftarrow \mathbf{u}^{n+1} + \Delta t * \tilde{b}_k * \mathbf{F}^{(k)}$ 
9:   end for
10: end for
```

4 Résultats dimension 2

PME : Porous Medium Equation,

HEAV : Heaviside,

HEAT : Chaleur.

4.1 PME et condition initiale Barenblatt : solution exacte

On compare la solution du problème avec solution exacte (donnée par Nadia avec décalage temporel de +1.) avec celle calculée par l'algorithme déterministe précédant.

```
[EDP]
NAME=PME;
p=@(u) u.^3;
D=0.5;
I=[-2.5 ,2.5, -2.5, 2.5];
Ti=0;
Tf=1.5;

u0=@(x,y) SolPME_2D(x,y,1)
uex=@(t,x,y) PME_Ex_2D(t,x,y,1)
ug=@(t,x,y) 0;
g=@(u) 0*u;
f=@(t,x,y) 0;
phi=1;
```

```
[SCHEMA]
N=[100,100];
Nt=400;
Ns=100;
IMEXname=IMEX_SSP3;
k=3;
```

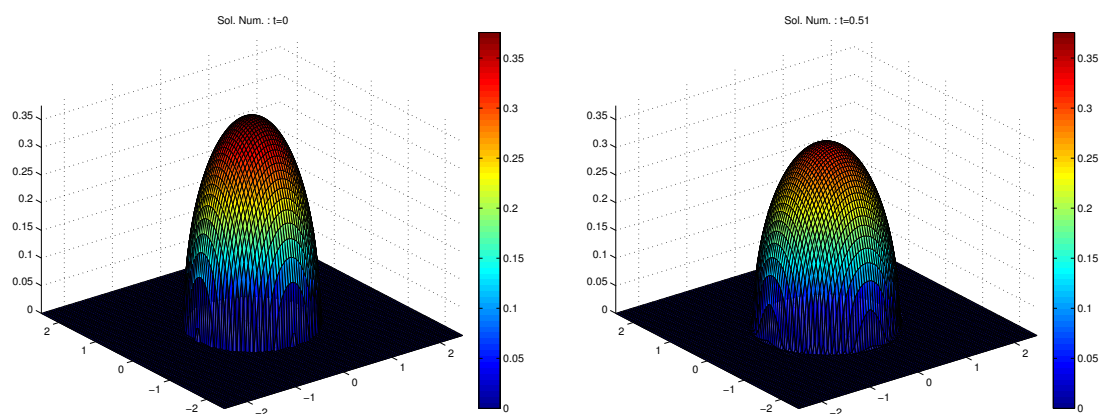


FIGURE 1 – Solutions numériques au temps $t = 0$ et $t = 0.51$

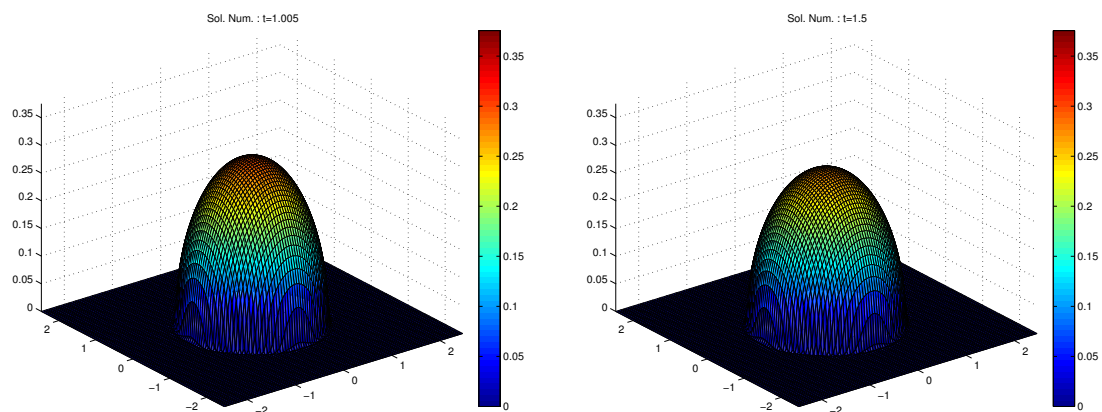


FIGURE 2 – Solutions numériques au temps $t = 1.0005$ et $t = 1.5$

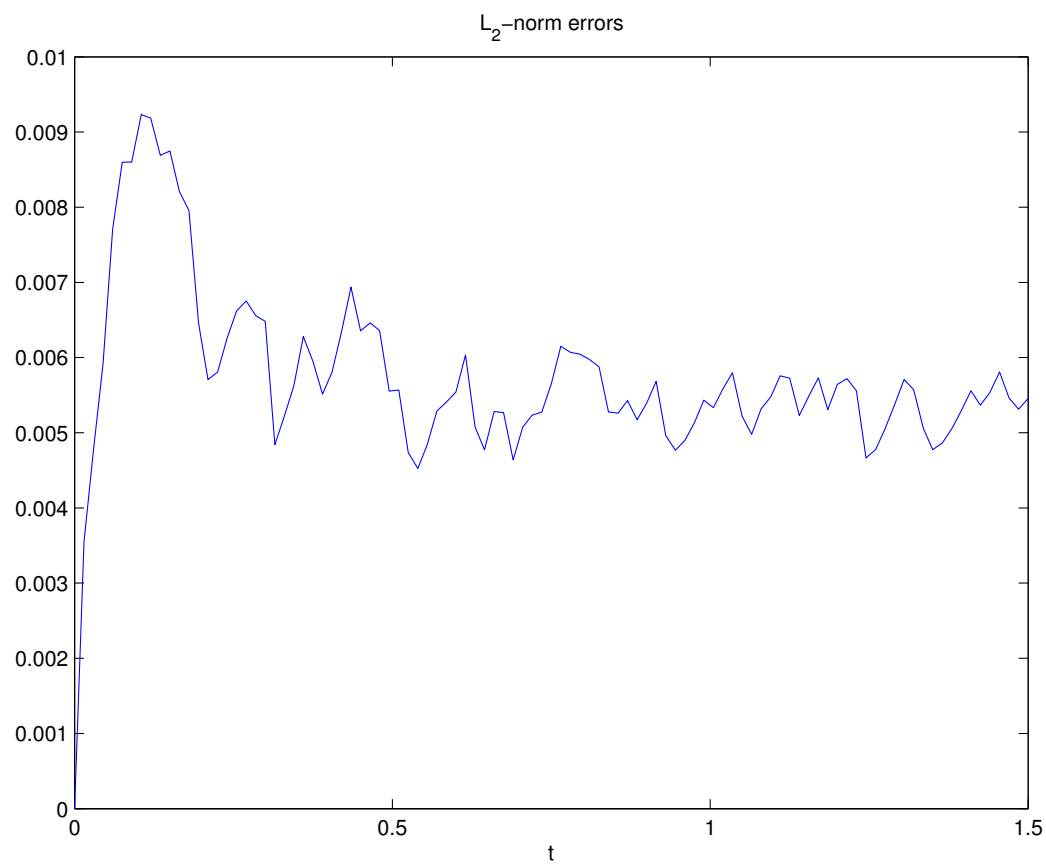


FIGURE 3 – Erreur en norme L_2

4.2 Condition initiale : densité loi normale

4.2.1 PME (PME_NORMAL_01.pde)

On résoud par l'algorithme déterministe le problème avec donnée initiale normale

```
[EDP]
NAME=PME;

p=@(u) u.^3;
D=0.5;
I=[-4 ,4, -4 ,4];
Ti=0;
Tf=20;

u0=@(x,y) NormalDensity2Dv2([0, 0],[1, 0;0, 1],x,y)
ug=@(t,x,y) zeros(length(t),length(x),length(y));
g=@(u) 0*u;
f=@(t,x,y) 0;
phi=1;

[SCHEMA]
N=[100,100];
Nt=1000;
Ns=100;
IMEXname=IMEX_SSP3;
matlabpoolNum=4;

k=3;
```

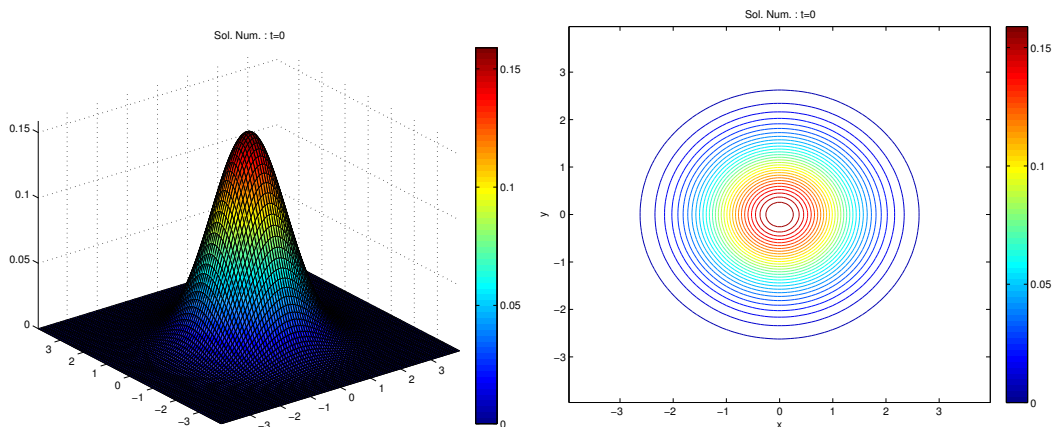


FIGURE 4 – Solution numérique au temps $t = 0$

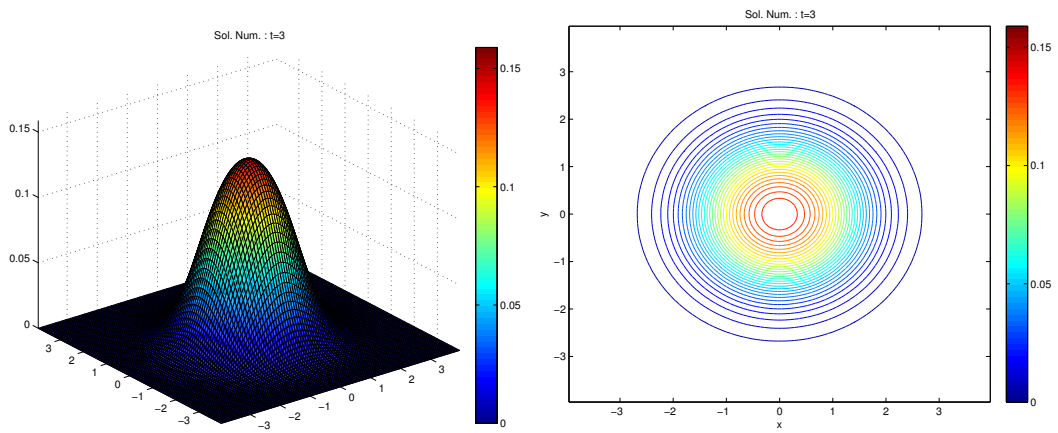


FIGURE 5 – Solution numérique au temps $t = 3$

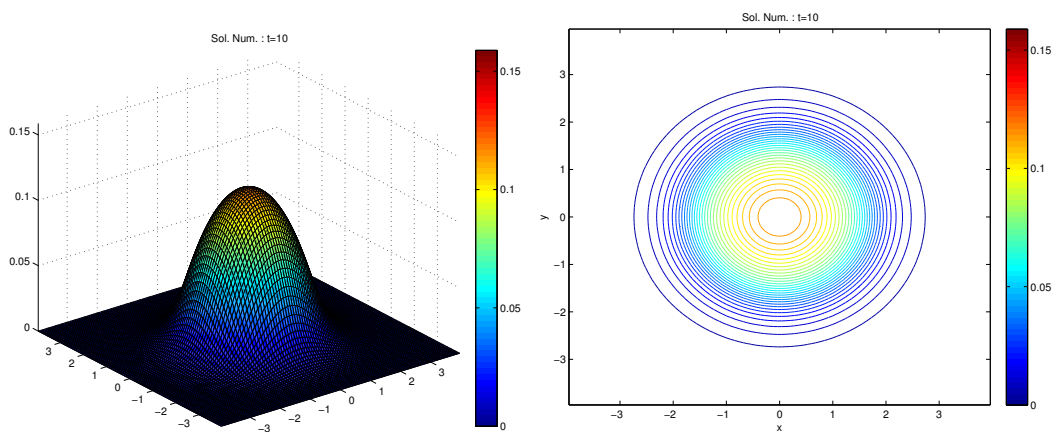


FIGURE 6 – Solution numérique au temps $t = 10$

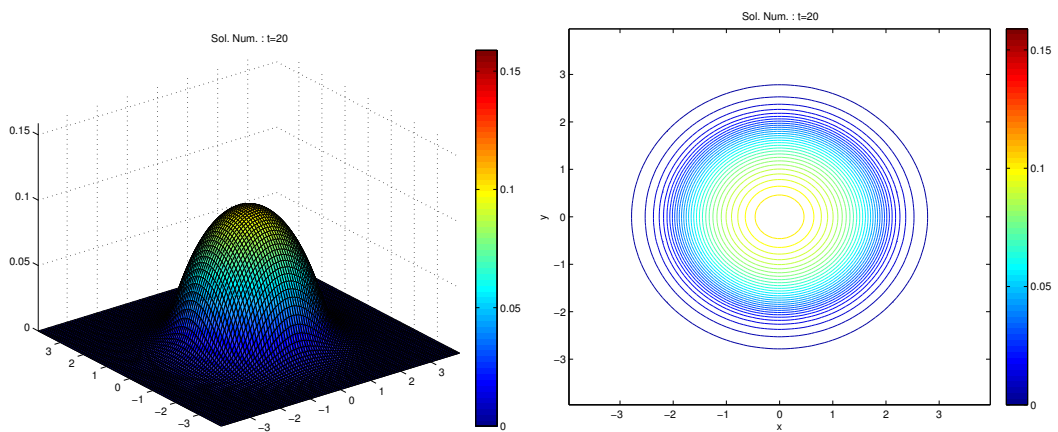


FIGURE 7 – Solution numérique au temps $t = 20$

4.2.2 HEAT (HEAT_NORMAL_01.pde)

On résoud par l'algorithme déterministe le problème avec donnée initiale normale

```
[EDP]
NAME=HEAT;

p=@(u) u;
D=0.5;
I=[-4 ,4, -4 ,4];
Ti=0;
Tf=0.7;

u0=@(x,y) NormalDensity2Dv2([0, 0],[1, 0;0, 1],x,y)
ug=@(t,x,y) zeros(length(t),length(x),length(y));
g=@(u) 0*u;
f=@(t,x,y) 0;
phi=1;

[SCHEMA]
N=[100,100];
Nt=1000;
Ns=100;
IMEXname=IMEX_SSP3;
matlabpoolNum=4;

k=3;
```

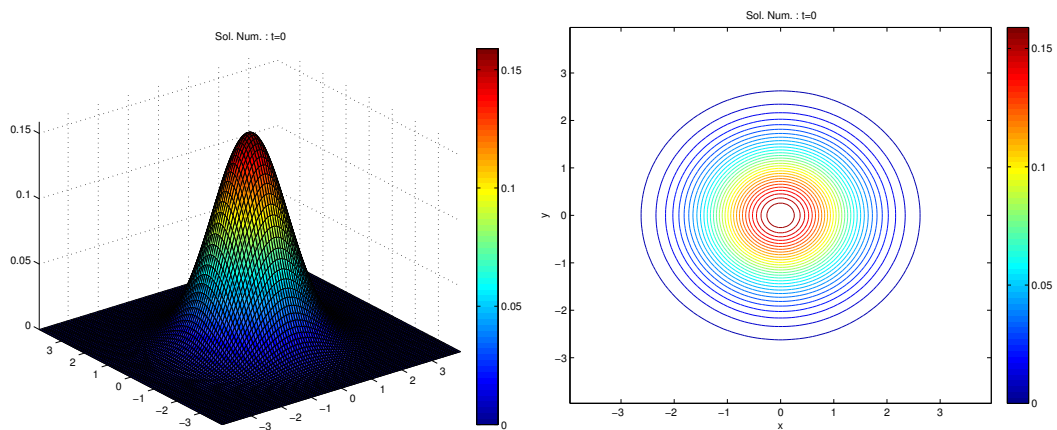


FIGURE 8 – Solution numérique au temps $t = 0$

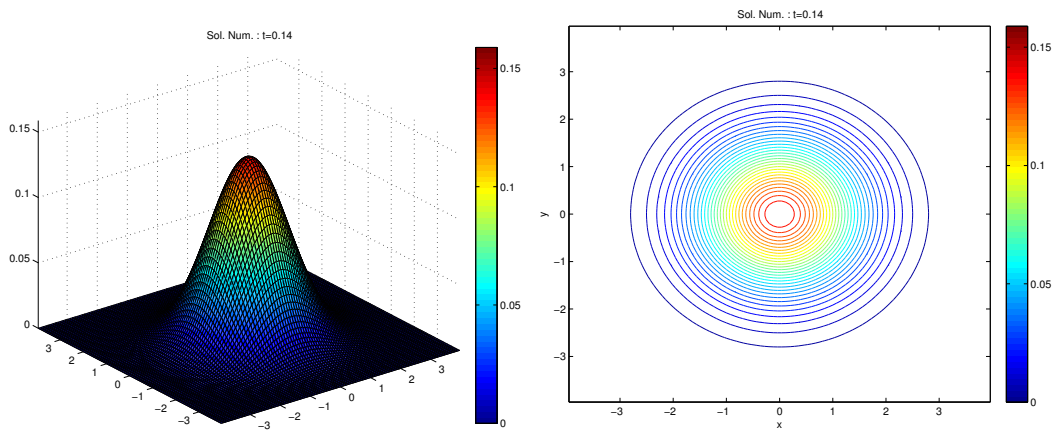


FIGURE 9 – Solution numérique au temps $t = 0.14$

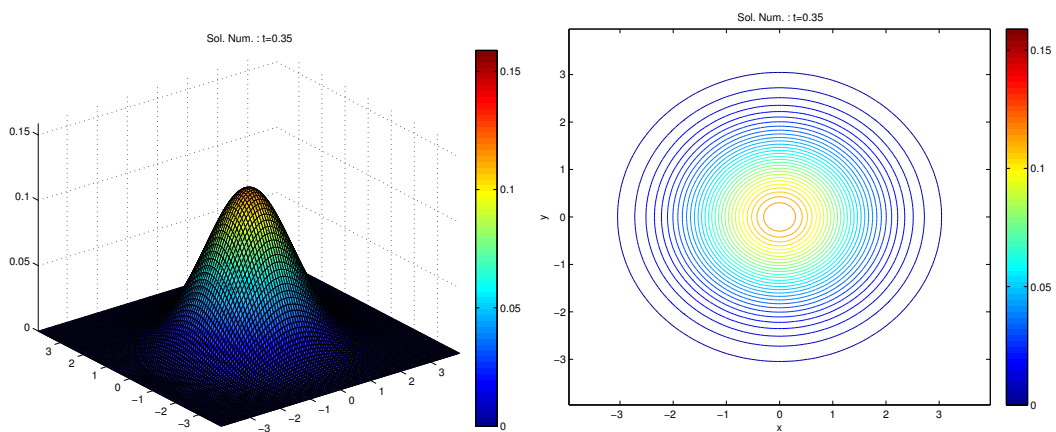


FIGURE 10 – Solution numérique au temps $t = 0.35$

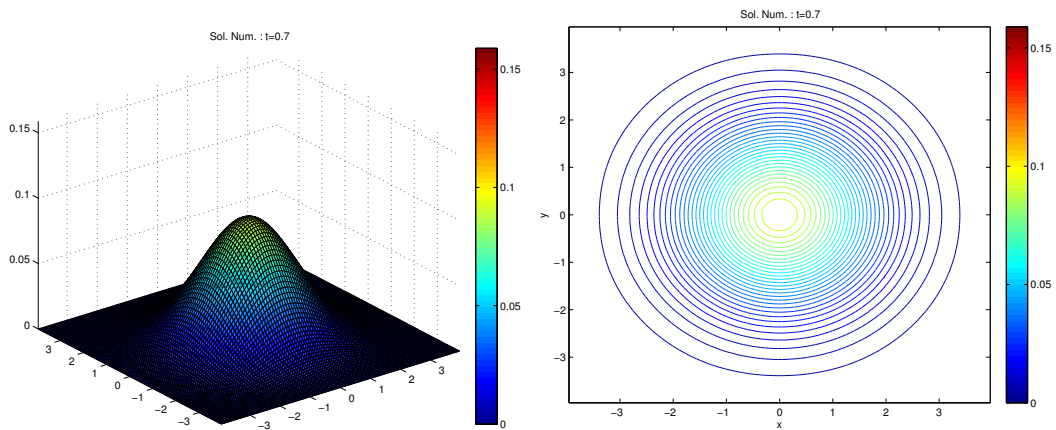


FIGURE 11 – Solution numérique au temps $t = 0.70$

4.2.3 HEAV (HEAV_NORMAL_01.pde)

On résoud par l'algorithme déterministe le problème avec donnée initiale normale

```
[EDP]
NAME=HEAV;

p=@(u) u.*(u-0.07)>0;
D=0.5;
I=[-4 ,4, -4 ,4];
Ti=0;
Tf=0.7;

u0=@(x,y) NormalDensity2Dv2([0, 0],[1, 0;0, 1],x,y)
ug=@(t,x,y) zeros(length(t),length(x),length(y));
g=@(u) 0*u;
f=@(t,x,y) 0;
phi=1;

[SCHEMA]
N=[100,100];
Nt=1000;
Ns=100;
IMEXname=IMEX_SSP3;
matlabpoolNum=4;

k=3;
```

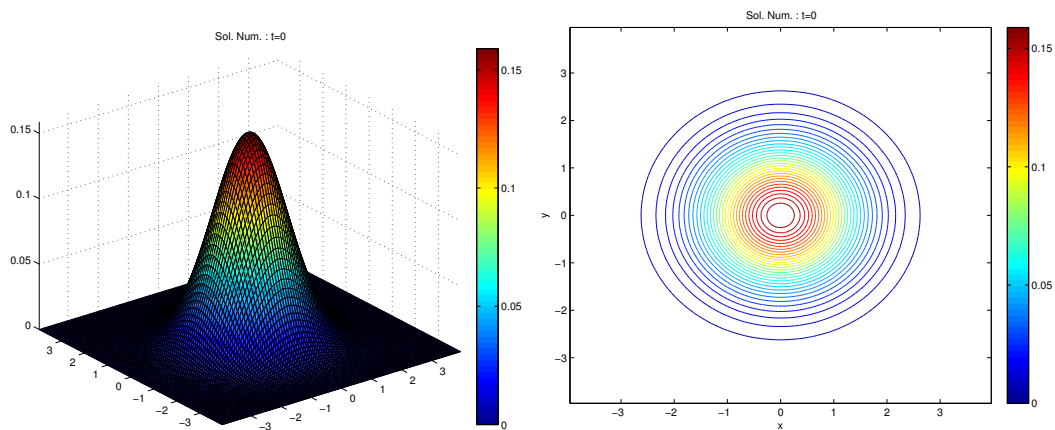


FIGURE 12 – Solution numérique au temps $t = 0$

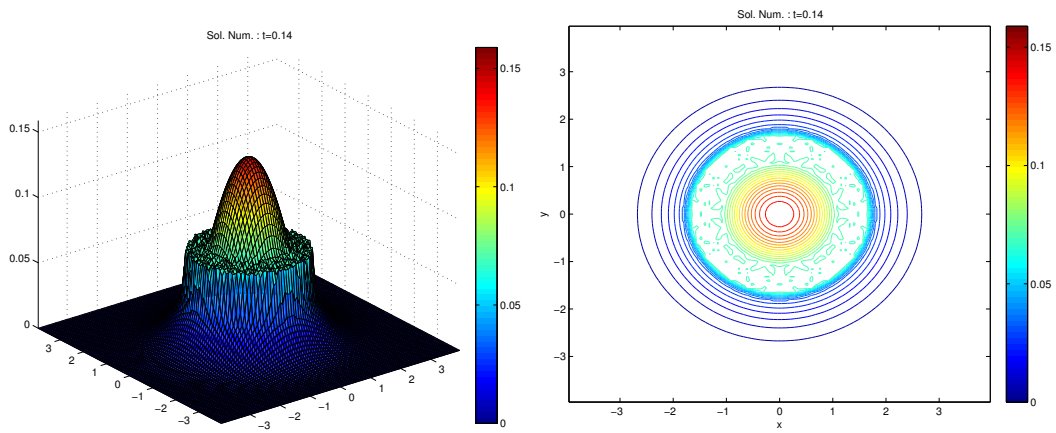


FIGURE 13 – Solution numérique au temps $t = 0.14$

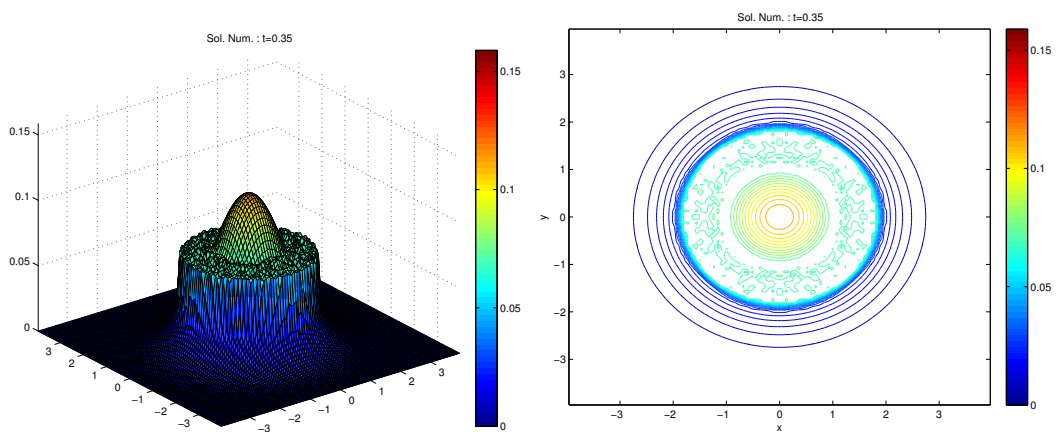


FIGURE 14 – Solution numérique au temps $t = 0.35$

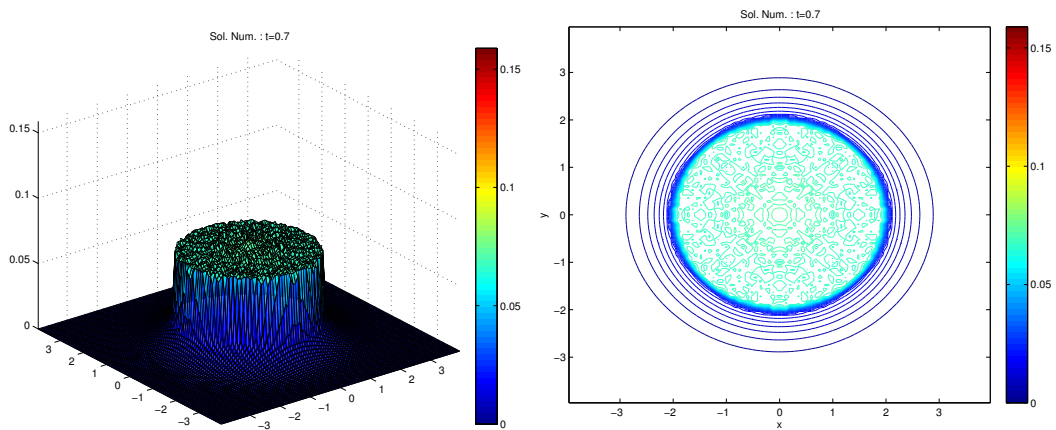


FIGURE 15 – Solution numérique au temps $t = 0.70$

4.2.4 Comparaison : HEAT/HEAV

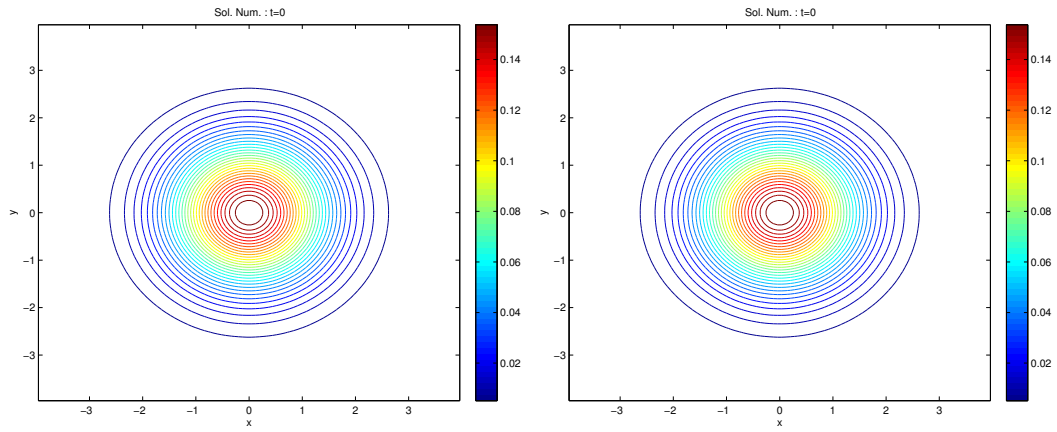


FIGURE 16 – Solutions numériques au temps $t = 0$ HEAT (left) et HEAV (right)

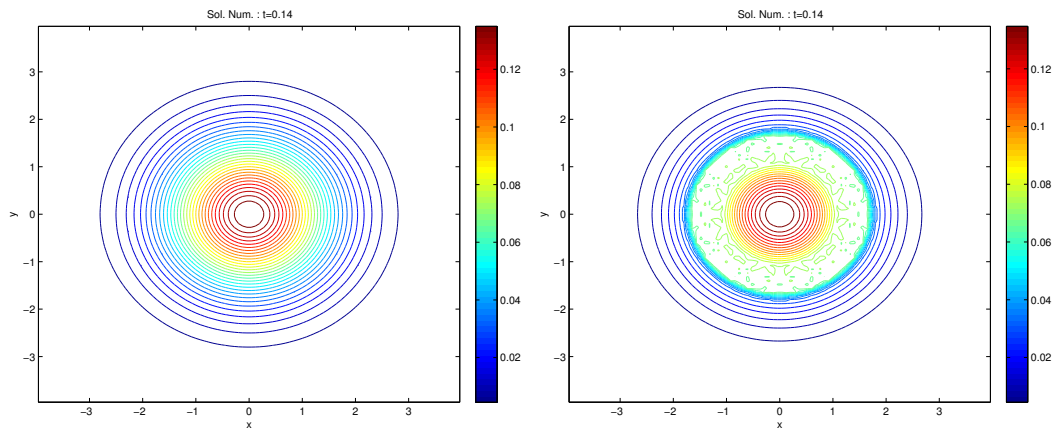


FIGURE 17 – Solutions numériques au temps $t = 0.14$ HEAT (left) et HEAV (right)

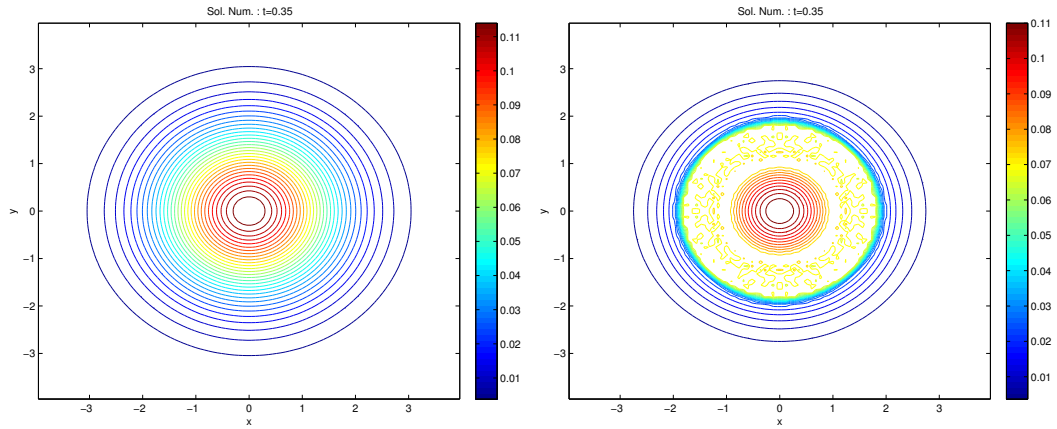


FIGURE 18 – Solutions numériques au temps $t = 0.35$ HEAT (left) et HEAV (right)

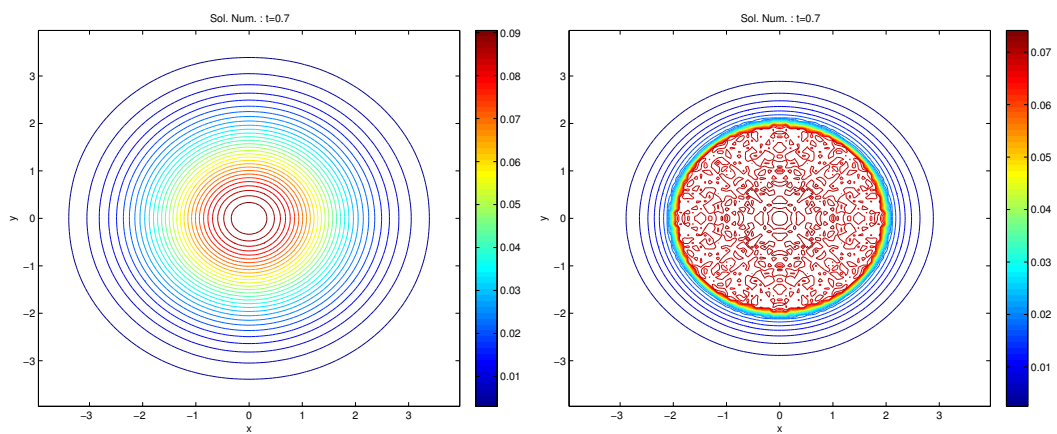


FIGURE 19 – Solutions numériques au temps $t = 0.7$ HEAT (left) et HEAV (right)

4.3 Condition initiale bimodale

4.3.1 PME (Init_PME_BIN0_02.pde)

On résoud par l'algorithme déterministe le problème avec donnée initiale bimodale

```
[EDP]
NAME=PME;

p=@(u) u.^3;
D=0.5;
I=[-5 ,5, -5 ,5];
Ti=0;
Tf=10.5;

u0=@(x,y) (1/2)*( NormalDensity2Dv2([0, 1],[.1, 0;0, .1],x,y) + NormalDensity2Dv2([0,-1],[.2, 0;0, .2],x,y));

ug=@(t,x,y) zeros(length(t),length(x),length(y));
g=@(u) 0*u;
f=@(t,x,y) 0;
phi=0.5;

[SCHEMA]
N=[120,120];
Nt=800;
Ns=100;
IMEXname=IMEX_SSP3;
matlabpoolNum=4;

k=3;
```

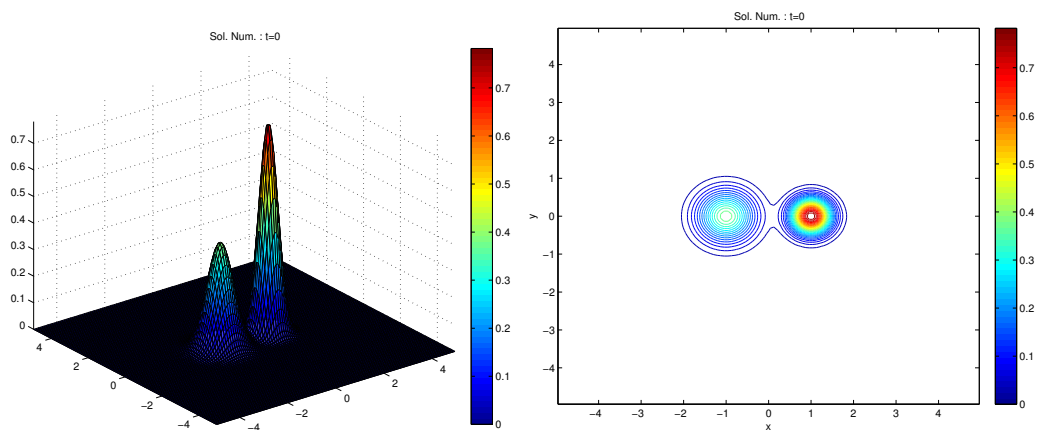


FIGURE 20 – Solution numérique au temps $t = 0$

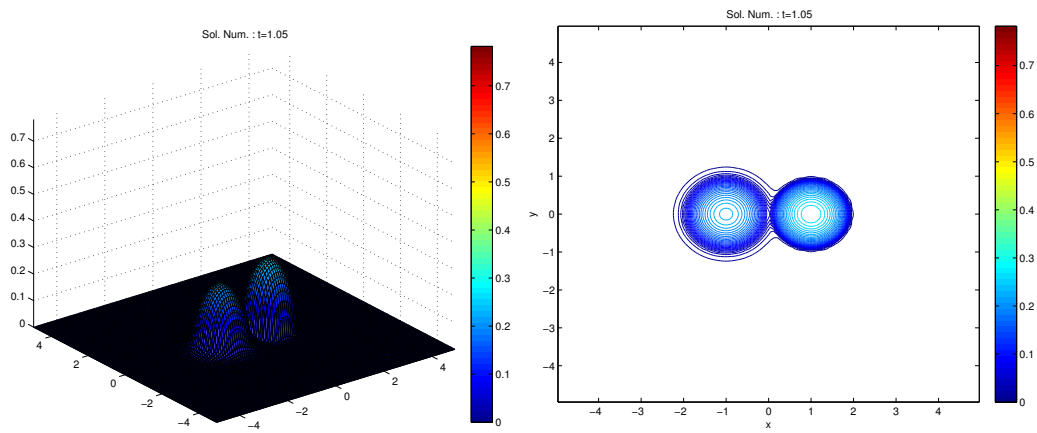


FIGURE 21 – Solution numérique au temps $t = 1.5$

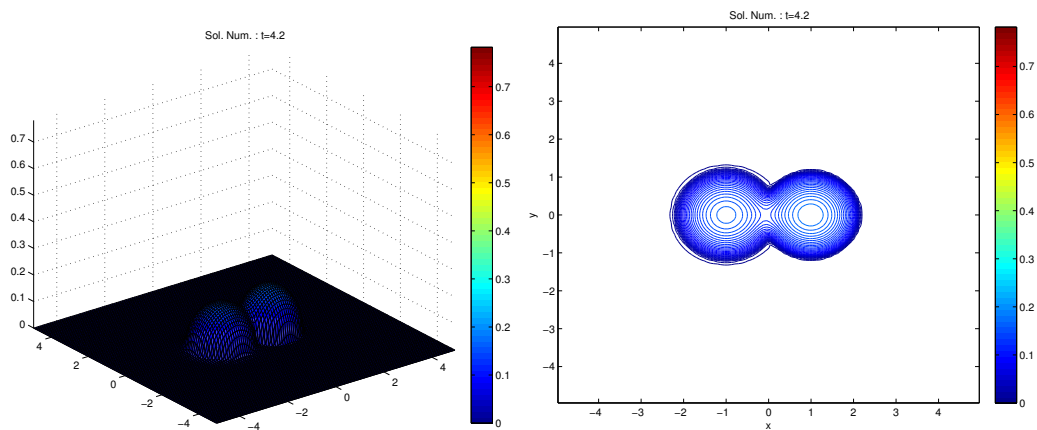


FIGURE 22 – Solution numérique au temps $t = 4.2$

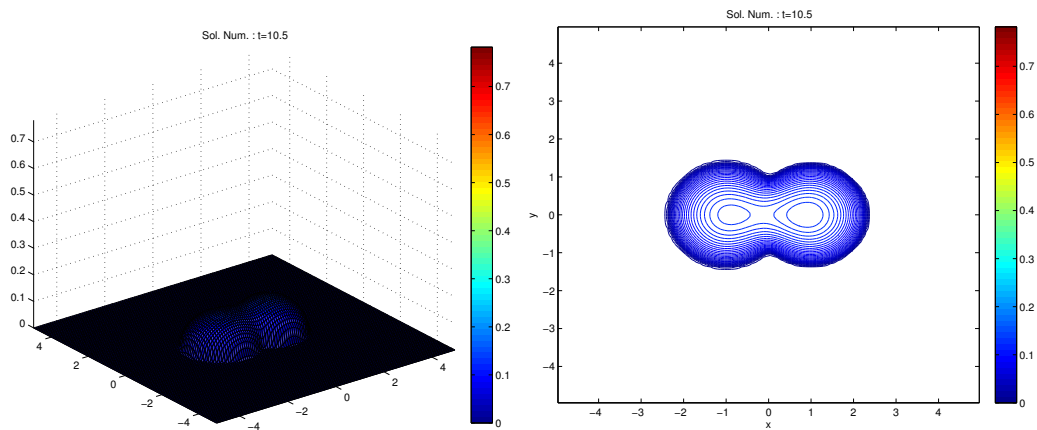


FIGURE 23 – Solution numérique au temps $t = 10.5$

4.3.2 HEAT (Init_HEAT_BIN0_02.pde)

On résoud par l'algorithme déterministe le problème avec donnée initiale bimodale

```
[EDP]
NAME=HEAT;

p=@(u) u;
D=0.5;
I=[-3.5 ,3.5, -3.5 ,3.5];
Ti=0;
Tf=0.75;

u0=@(x,y)(1/2)*(NormalDensity2D(x,y,[0,1],sqrt([.1,.1]))+NormalDensity2D(x,y,[0,-1],sqrt([.2,.2])));

ug=@(t,x,y) zeros(length(t),length(x),length(y));
g=@(u) 0*u;
f=@(t,x,y) 0;
phi=0.5;

[SCHEMA]
N=[100,100];
Nt=800;
Ns=100;
IMEXname=IMEX_SSP3;

k=3;
```

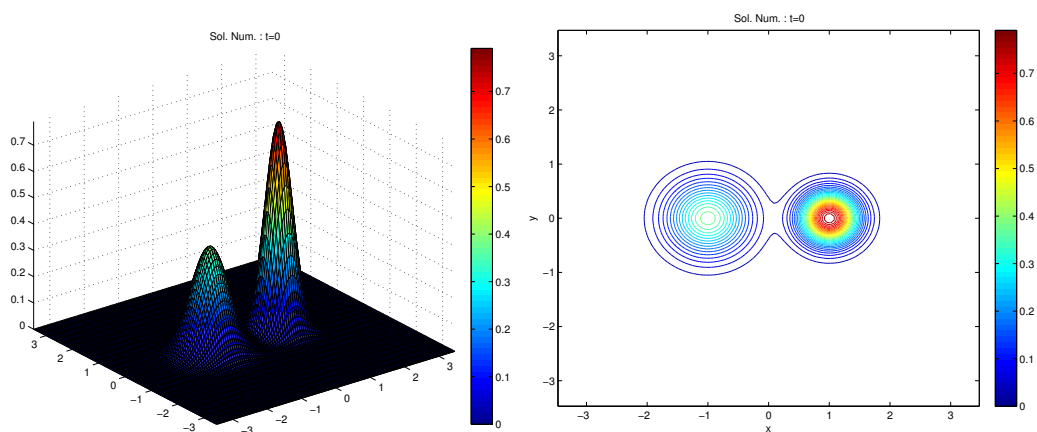


FIGURE 24 – Solution numérique au temps $t = 0$

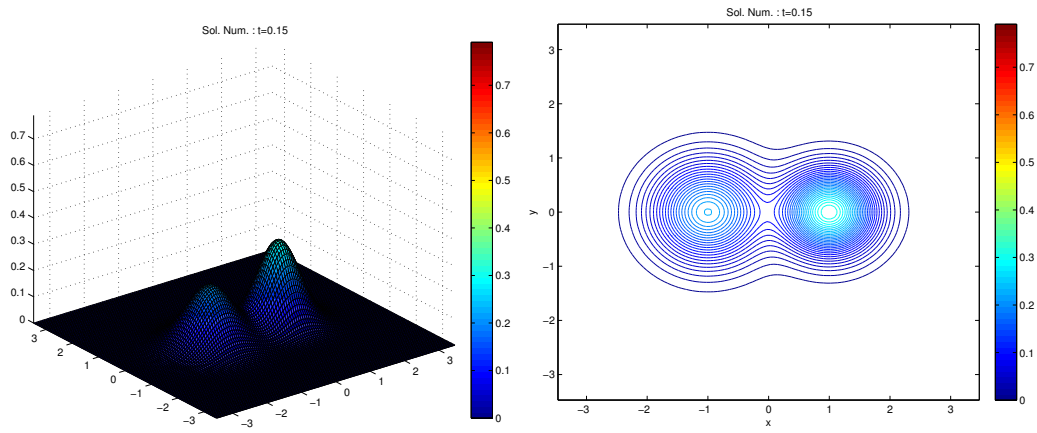


FIGURE 25 – Solution numérique au temps $t = 0.15$

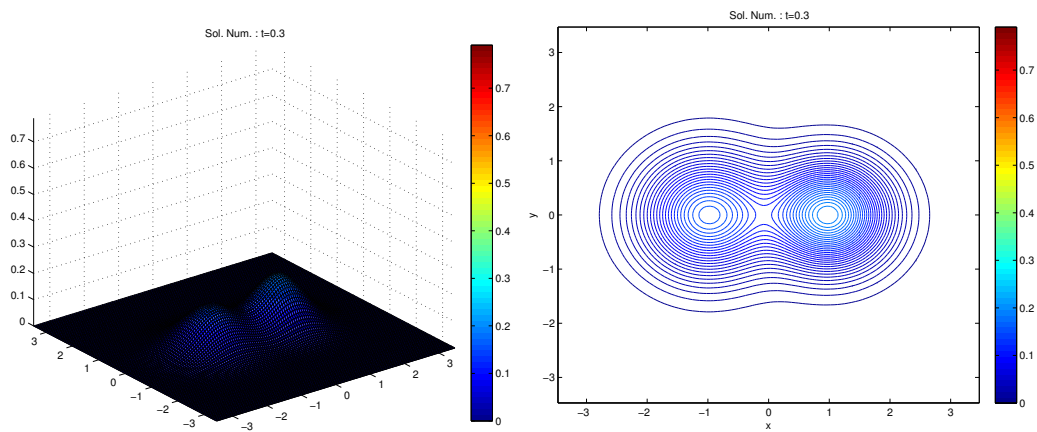


FIGURE 26 – Solution numérique au temps $t = 0.30$

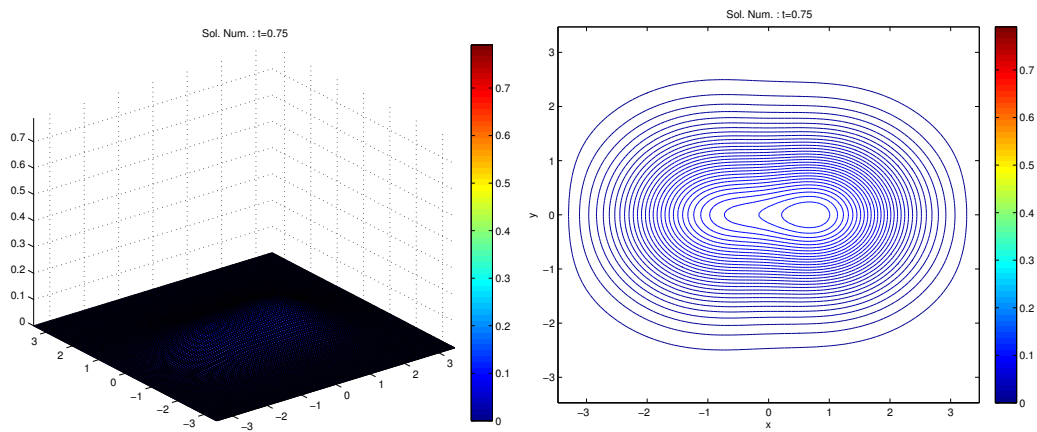


FIGURE 27 – Solution numérique au temps $t = 0.75$

4.3.3 HEAV (Init_HEAV_BIN0_02.pde)

On résoud par l'algorithme déterministe le problème avec donnée initiale bimodale

```
[EDP]
NAME=HEAV;

p=@(u) u.*(u-0.1)>0);;
D=0.5;
I=[-3.5 ,3.5, -3.5 ,3.5];
Ti=0;
Tf=0.75;

u0=@(x,y) (1/2)*( NormalDensity2Dv2([0, 1],[.1, 0;0, .1],x,y) + NormalDensity2Dv2([0,-1],[.2, 0;0, .2],
x,y));

ug=@(t,x,y) zeros(length(t),length(x),length(y));
g=@(u) 0*u;
f=@(t,x,y) 0;
phi=0.5;

[SCHEMA]
N=[120,120];
Nt=800;
Ns=100;
IMEXname=IMEX_SSP3;
matlabpoolNum=4;

k=3;
```

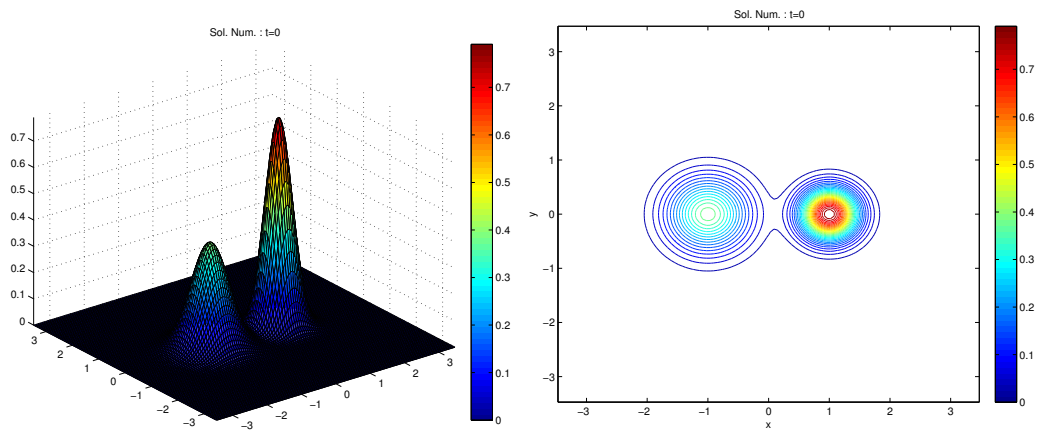


FIGURE 28 – Solution numérique au temps $t = 0$

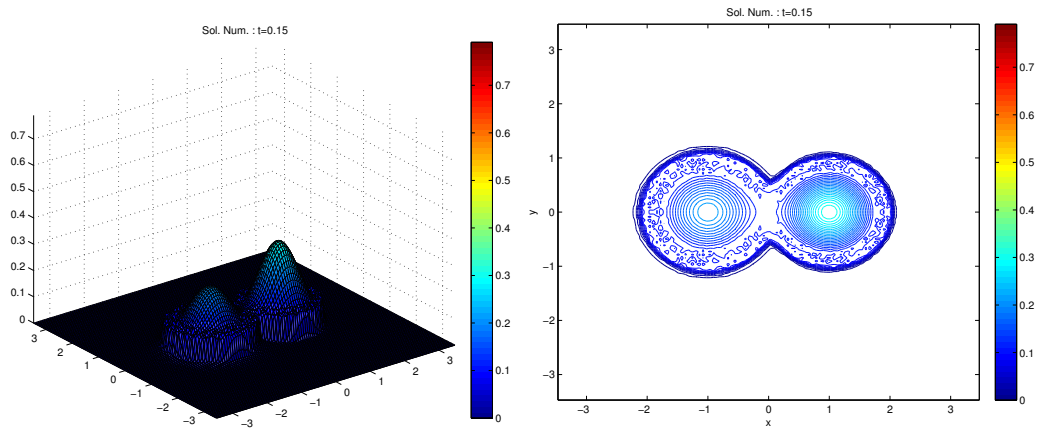


FIGURE 29 – Solution numérique au temps $t = 0.15$

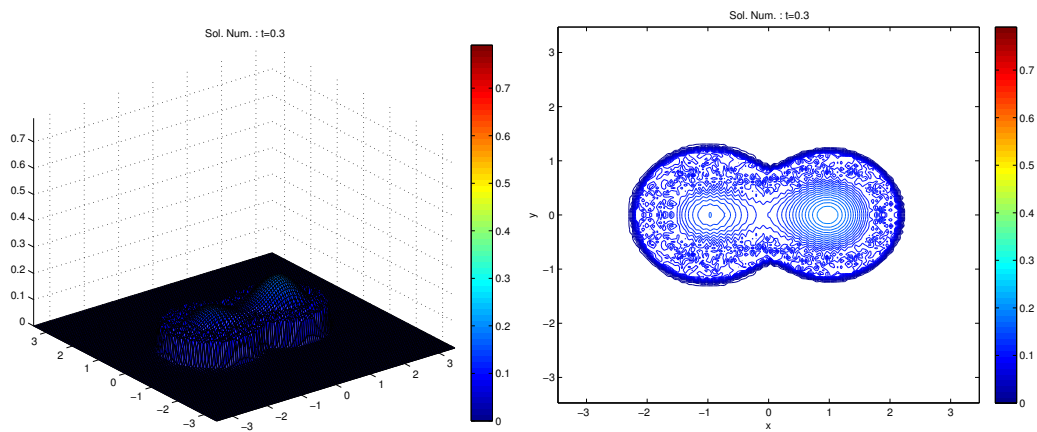


FIGURE 30 – Solution numérique au temps $t = 0.30$

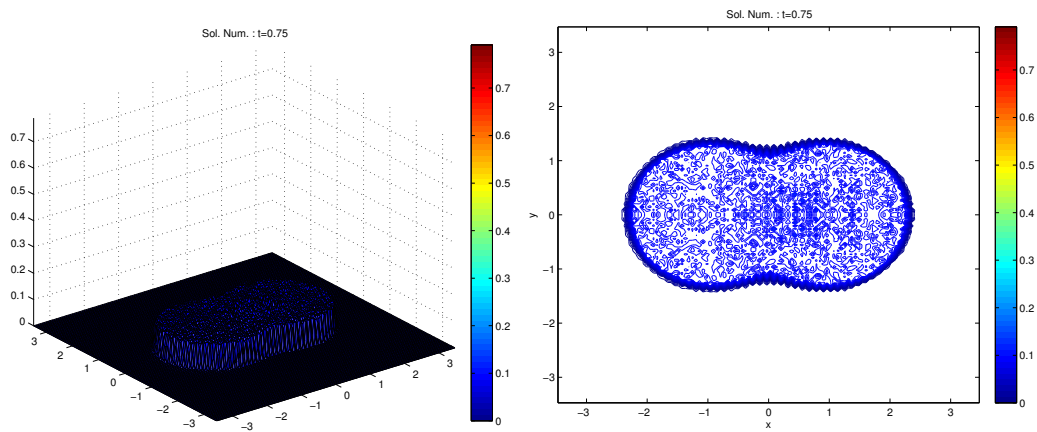


FIGURE 31 – Solution numérique au temps $t = 0.75$

4.3.4 Comparaison : HEAT/HEAV

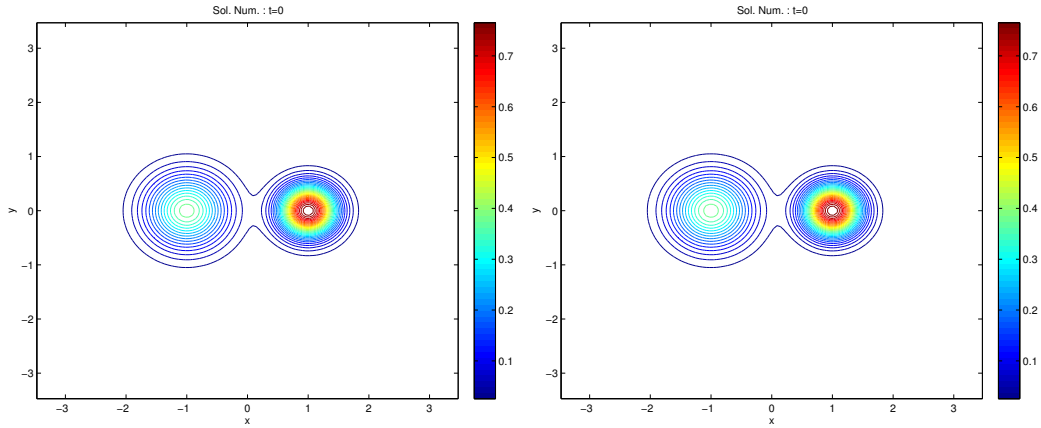


FIGURE 32 – Solutions numériques au temps $t = 0$ HEAT (left) et HEAV (right)

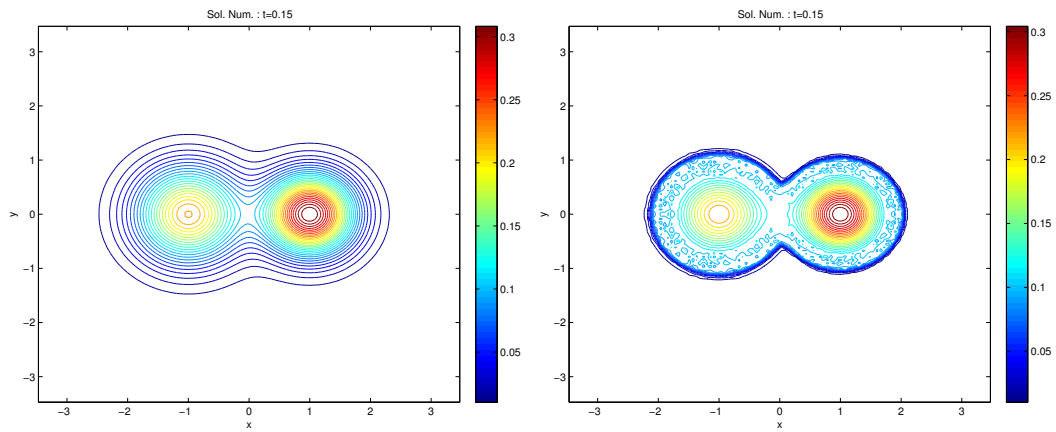


FIGURE 33 – Solutions numériques au temps $t = 0.15$ HEAT (left) et HEAV (right)

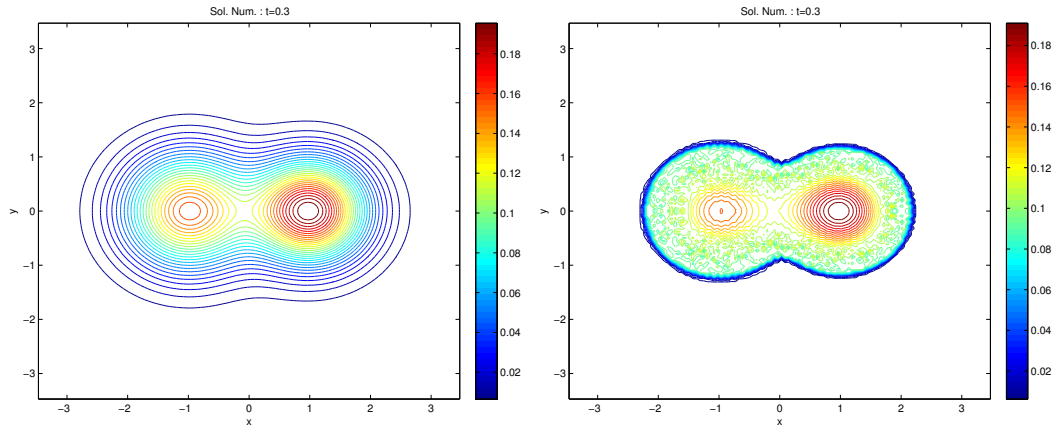


FIGURE 34 – Solutions numériques au temps $t = 0.30$ HEAT (left) et HEAV (right)

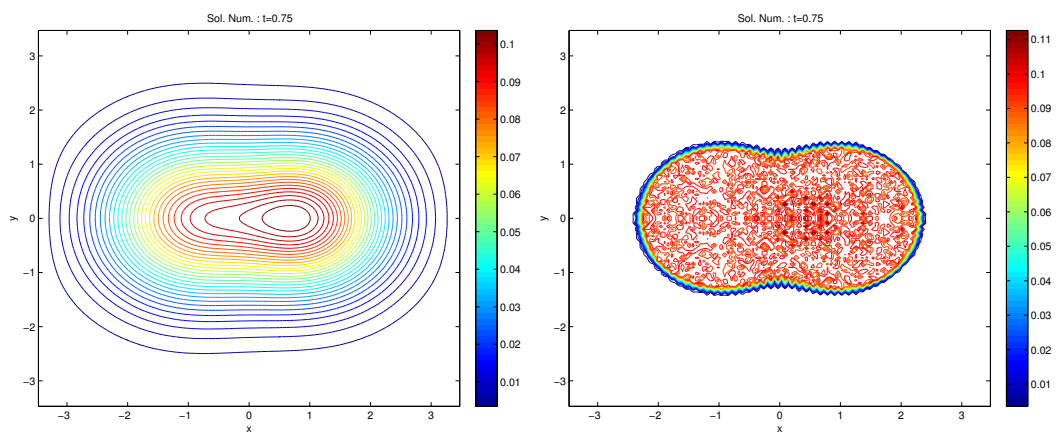


FIGURE 35 – Solutions numériques au temps $t = 0.75$ HEAT (left) et HEAV (right)