

# DEVOIR 1 – VÉRIFICATION DE CODE

## Diffusion du sel dans un pilier de béton poreux

---

**Eduards Blandin 1893699**  
**Jacques Desfossés 61902**  
**Timothée Duruisseau 1949883**

[https://github.com/tiduru/MEC8211\\_VetV/tree/main/Devoir1](https://github.com/tiduru/MEC8211_VetV/tree/main/Devoir1)



Atchafalaya Basin Bridge, I-10, Whiskey Bay  
[© MICHAELAT1, CC BY-SA 3.0, via Wikimedia Commons]



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE

## A) SIMPLIFIER ET ÉTABLIR LE PROBLÈME

- a. L'équation  $\frac{\partial C}{\partial t} = D_{eff} \nabla^2 C - S$  est de type **parabolique**
- b. En coordonnées cylindriques, l'équation devient  $\frac{\partial C}{\partial t} = D_{eff} \left[ \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial C}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 C}{\partial \theta^2} + \frac{\partial^2 C}{\partial z^2} \right] - S$
- Nous sommes en présence d'une **symétrie axiale de révolution**
  - On réduit le problème à une **diffusion radiale seulement**, i.e.  $C(t, r, \theta, z) = C(t, r)$ 
    - Cylindre infiniment haut
    - Concentration constante  $C_e$  à la surface du pilier
    - Flux nul en  $r=0$
    - Concentration initiale nulle à l'intérieur du pilier

L'équation simplifiée est

$$\frac{\partial C}{\partial t} = D_{eff} \left[ \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial C}{\partial r} \right) \right] - S = D_{eff} \left[ \frac{1}{r} \frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial r^2} \right] - S$$

## A) SIMPLIFIER ET ÉTABLIR LE PROBLÈME (SUITE)

### c. Discrétisation du domaine pour 5 nœuds

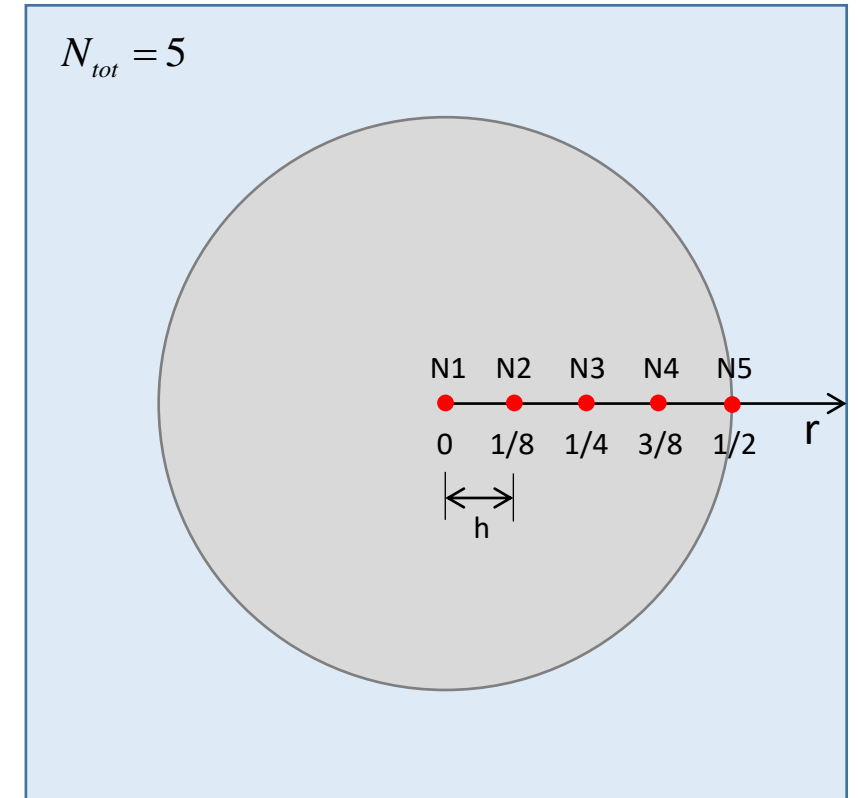
- Intervalles **constants**  $h = R / (N_{tot} - 1) = 1/8 m$
- Position sur l'axe  $r$ :  $r_i = (i - 1)h$ ,  $1 \leq i \leq N_{tot}$

### d. i) Conditions frontières

- Condition frontière de **Dirichlet**:  $C(t, r = R) = C_e$
- Condition frontière de **Neumann**:  $\frac{\partial C}{\partial r}(t, r = 0) = 0$

### ii) Conditions initiales

- Concentration initiale **nulle**:  $C(t = 0, r < R) = 0$



## B) DIFFÉRENCES FINIES

a. **Équations nodales** pour Euler implicite en temps et  $\frac{\partial C}{\partial r}|_i = \frac{C_{i+1}^t - C_i^t}{\Delta r}$ ,  $\frac{\partial^2 C}{\partial r^2}|_i = \frac{C_{i+1}^t - 2C_i^t + C_{i-1}^t}{\Delta r^2}$  en espace

- Dirichlet :  $C_5^t = C_e$
- Neumann :  $(C_2^t - C_1^t)/h = 0 \Rightarrow C_1^t = C_2^t$
- Noeuds 2 à 4:  $\frac{C_i^t - C_i^{t-1}}{\Delta t} = D_{eff} \left[ \frac{1}{r_i} \frac{C_{i+1}^t - C_i^t}{h} + \frac{C_{i+1}^t - 2C_i^t + C_{i-1}^t}{h^2} \right] - S$

$$-\underbrace{\left( \frac{r_i D_{eff} \Delta t}{A_i} \right)}_{A_i} C_{i-1}^t + \underbrace{\left( r_i h^2 + 2r_i D_{eff} \Delta t + D_{eff} h \Delta t \right)}_{B_i + A_i + E} C_i^t - \underbrace{\left( D_{eff} h \Delta t + r_i D_{eff} \Delta t \right)}_{E + A_i} C_{i+1}^t = \underbrace{\left( r_i h^2 \right)}_{F_i} C_i^{t-1} - S r_i h^2 \Delta t$$

$$N_1: C_1^t - C_2^t = 0$$

$$N_2: -A_2 C_1^t + (B_2 + 2A_2 + E) C_2^t - (E + A_2) C_3^t = F_2$$

$$N_3: -A_3 C_2^t + (B_3 + 2A_3 + E) C_3^t - (E + A_3) C_4^t = F_3$$

$$N_4: -A_4 C_3^t + (B_4 + 2A_4 + E) C_4^t - (E + A_4) C_5^t = F_4$$

$$N_5: C_5^t = C_e$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -A_2 & B_2 + 2A_2 + E & -E - A_2 & 0 & 0 \\ 0 & -A_3 & B_3 + 2A_3 + E & -E - A_3 & 0 \\ 0 & 0 & -A_4 & B_4 + 2A_4 + E & -E - A_4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1^t \\ C_2^t \\ C_3^t \\ C_4^t \\ C_5^t \end{bmatrix} = \begin{bmatrix} 0 \\ F_2 \\ F_3 \\ F_4 \\ C_e \end{bmatrix}$$

Système matriciel à résoudre à chaque pas de temps (forme non-compacte)

## B) DIFFÉRENCES FINIES (SUITE)

- b. Le système matriciel **linéaire** est de la forme  $[M]\{C\} = \{V\}$
- Il est résolu **à chaque pas de temps** par une **méthode directe** étant donné sa petite taille:  $\{C\} = [M]^{-1}\{V\}$
  - Dans notre code, on laisse Matlab décider du meilleur algorithme avec l'opérateur '\':  $C = M \setminus V$ ;
- c. Les ordres de précision attendus du schéma global sont  $O(h + \Delta t)$
- Temps: ordre 1
  - Espace: ordre 1
- d. Le schéma numérique est **inconditionnellement stable**

## C) SOLUTION ANALYTIQUE DU RÉGIME STATIONNAIRE

L'équation **elliptique** du régime **stationnaire** est  $D_{eff} \left[ \frac{1}{r} \frac{d}{dr} \left( r \frac{dC}{dr} \right) \right] = S \Rightarrow \boxed{\frac{d}{dr} \left( r \frac{dC}{dr} \right) = \frac{Sr}{D_{eff}}}$

On intègre deux fois:  $\int d \left( r \frac{dC}{dr} \right) = \int \frac{S}{D_{eff}} r dr \Rightarrow r \frac{dC}{dr} = \frac{Sr^2}{2D_{eff}} + C_1$

$$\int dC = \int \left( \frac{Sr}{2D_{eff}} + \frac{C_1}{r} \right) dr \Rightarrow \boxed{C(r) = \frac{Sr^2}{4D_{eff}} + C_1 \ln r + C_2}$$

On trouve les **constantes d'intégration** à l'aide des conditions frontières:

- Dirichlet:  $C(r=R) = C_e \Rightarrow C_2 = C_e - \frac{SR^2}{4D_{eff}} - C_1 \ln R$
- Neumann:  $C'(r=0) = 0 \Rightarrow C_1 = 0$

La solution analytique est  $C(r) = \frac{Sr^2}{4D_{eff}} + C_e - \frac{SR^2}{4D_{eff}} \Rightarrow \boxed{C(r) = \frac{S}{4D_{eff}} R^2 \left( \frac{r^2}{R^2} - 1 \right) + C_e}$

## D) CODE DE CALCUL GÉNÉRIQUE

La fonction **FickDF** a été écrite dans **Matlab** et calcule  $C(t,r)$  pour :

- Un nombre de nœuds total  $N_{tot}$
- Un pas de temps  $dt$  (en années)
- Un nombre d'années  $N_{dt}$
- Les schémas de différenciation 1 et 2
- Un terme source  $tsMeth$  constant ou du 1<sup>er</sup> ordre

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% École Polytechnique de Montréal
% MEC8211 A2022 Devoir 1
%
% Eduards Blandin
% Jacques Desfossés
% Timothée Duruisseau
%
% Cette fonction résout l'équation différentielle représentant la 2e loi
% de Fick exprimée en coordonnées cylindriques et calcule C(r,t) pour:
%
% - Un pilier de béton de rayon R submergé dans une eau saline
% - Un pilier (cylindre) infiniment haut
% - Une concentration Ce=10 mol/m^3 constante à la surface du pilier
%   correspondant à une condition frontière de Dirichlet
% - Un flux nul en r=0 correspondant à une condition frontière de Neumann
% - Une concentration initiale nulle dans le pilier C(r<R,0) = 0
%
%
% Variables
% -----
% entrée : Ntot - Nombre de noeuds, Entier >= 3
%           dt - Pas de temps [an], > 0
%           Ndt - Nombre de pas de temps, Entier >= 1
%           schema - Schéma de différenciation: 1 - Ordre 1
%                   2 - Ordre 2
%           tsMeth - Terme source: 0 - Constant S=1E-8 [mol/m^3/s]
%                   1 - 1er ordre, S=kC avec k=4E-9 [s^-1]
%
% sortie : C - Concentrations [mol/m^3]. Taille Ndt+1 x Ntot
%           Rangées (temps) : Ndt + 1
%           Colonnes (noeuds): Ntot
%           Ex: C(1,1) = Concentration initiale au noeud 1.
%               C(101,3) = Concentration au noeud 3, temps 100*dt
%
%           temps - Temps discrets de la simulation [an]. Taille Ndt+1
%
% test : 50 noeuds, 30 incréments d'une année, schéma d'ordre 2,
%        terme src cst: C = FickDF(50, 1, 30, 2, 0);
%
% Historique
% 02-Oct-2022 : Création
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [C, temps] = FickDF(Ntot, dt, Ndt, schema, tsMeth)
```

[https://github.com/tiduru/MEC8211\\_VetV/tree/main/Devoir1/FickDF.m](https://github.com/tiduru/MEC8211_VetV/tree/main/Devoir1/FickDF.m)

## E) SOLUTION NUMÉRIQUE DU RÉGIME STATIONNAIRE

a. La solution **numérique** du régime **stationnaire** est obtenue de deux façons:

1) Méthode **transitoire**: On fait rouler le code jusqu'au régime stationnaire avec les paramètres ci-dessous

- Pas de temps de 1 an:  $dt = 1$
- Nombre de pas de temps:  $Ndt=1000$

2) Méthode **directe**: On utilise une nouvelle fonction **FickDFStat**, dérivée de la fonction **FickDF**, qui résout directement l'équation elliptique du régime stationnaire et qui ne nécessite donc aucune discrétisation temporelle.

b. La **vérification** du code, pour le **schéma d'ordre 1**, donne des résultats similaires pour les 2 méthodes. La fonction **FickVerifStat** fait le graphe des erreurs et donne la pente entre les 2 intervalles les plus élevés. On utilise

- Intervalle maximum ( $N_{\min} = 3$ ) :  $h_{\max} = R / (N_{\min} - 1) = 0.25 \text{ m}$
- Intervalle minimum ( $N_{\max} = 1001$ ) :  $h_{\min} = R / (N_{\max} - 1) = 0.0005 \text{ m}$

Matlab

```
>> FickVerifStat(3, 1001, 1, "directe")  
pentes O(1): L1=1.000000, L2=1.085079, Linf=1.000000
```



## E) SOLUTION NUMÉRIQUE DU RÉGIME STATIONNAIRE (SUITE)

b. L'ordre de précision **observé** avec les méthodes **directe** et **transitoire** est

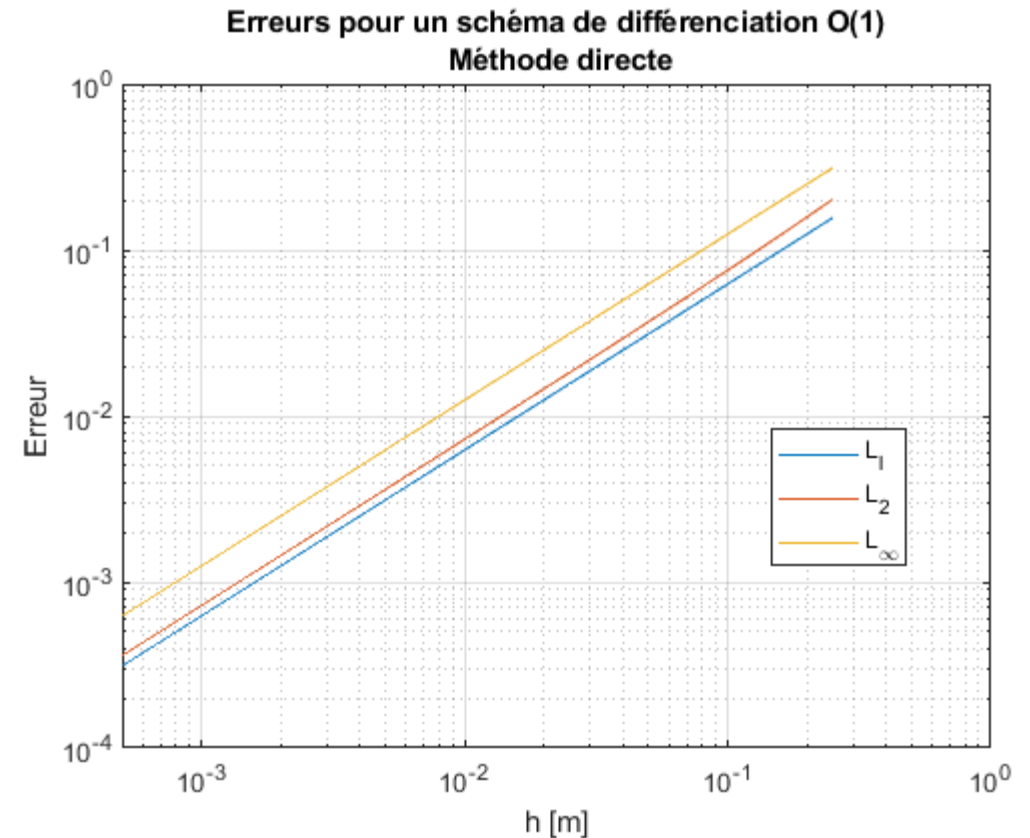
- Ordre de 1 pour les erreurs  $L_1$  et  $L_\infty$
- Ordre de 1.09 pour  $L_2$



L'ordre de convergence observé atteint l'ordre de convergence formel (ordre 1)

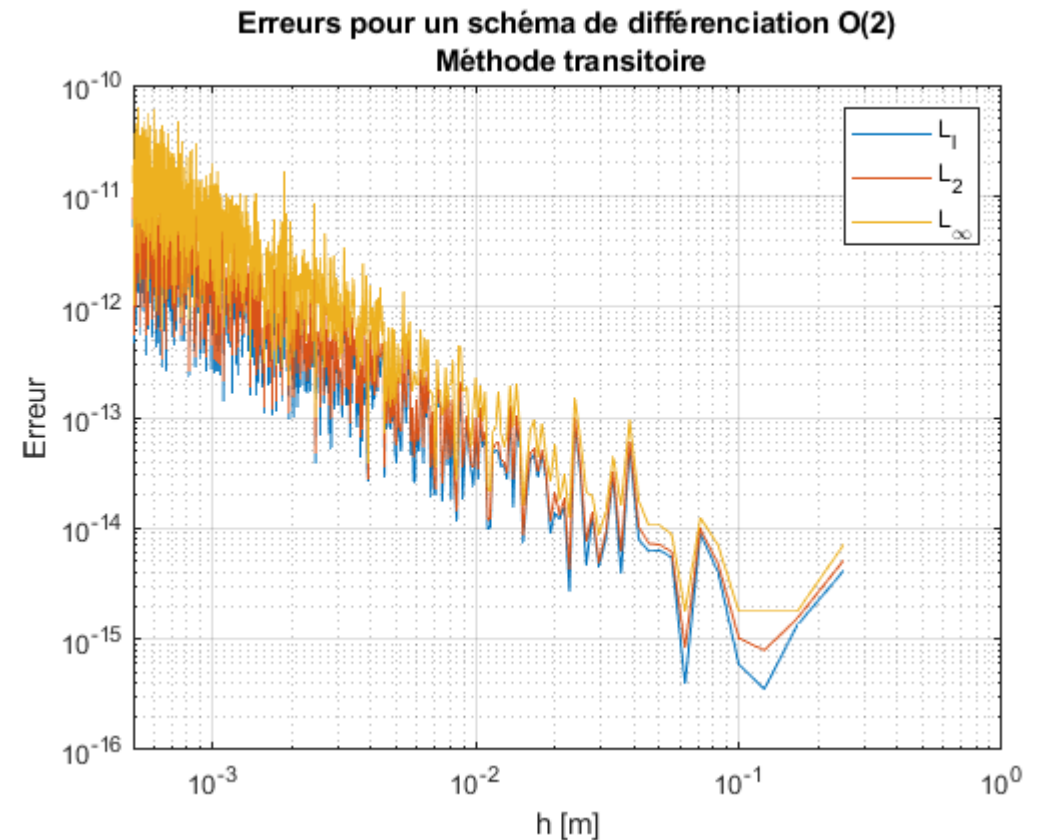
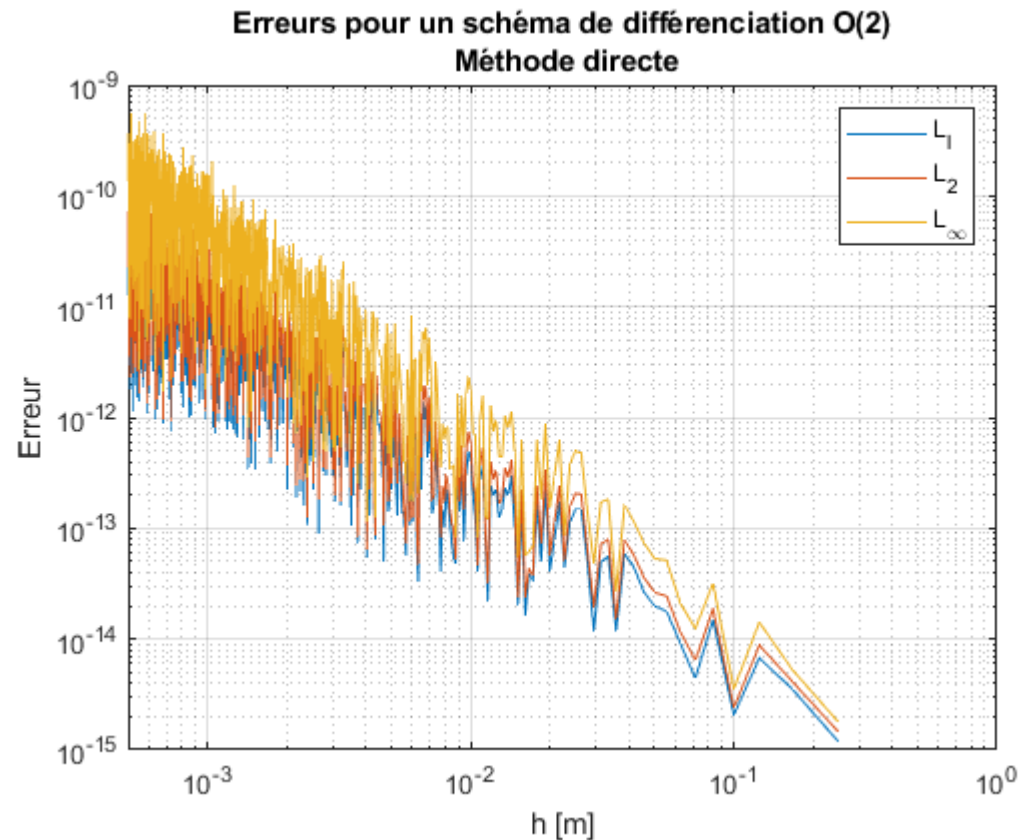
c. Problèmes constatés:

- Cette méthode de vérification requiert une **solution analytique calculable**, ce qui n'est pas toujours possible.
- Avec la condition de **Neumann** en  $r=0$  et un schéma d'ordre 1, les nœuds 1 et 2 ont toujours la **même concentration**, ce qui n'est pas le cas en réalité. Cela contribue à l'erreur. L'utilisation d'un pas de temps  $h$  variable pourrait palier ce problème.



## F) SCHÉMAS DE DIFFÉRENCIATION D'ORDRE 2

- a. Pour le schéma d'ordre 2, l'approximation avant « de Gear » est utilisée pour la condition frontière de Neumann.



## F) SCHÉMAS DE DIFFÉRENCIATION D'ORDRE 2

b. Les méthodes **directe** et **transitoire** ont les propriétés suivantes:

- Erreur obtenue de l'ordre de la **précision machine** (erreurs  $< 1E-14$  pour  $h_{\min}$ )
- Pour  $h < 0.1$  m, l'erreur augmente lorsque le pas de temps diminue. C'est donc l'**erreur de représentation des nombres** qui domine.
- Pour la méthode transitoire, la solution obtenue contient aussi une erreur associée au fait que l'on n'atteint qu'**asymptotiquement** la solution stationnaire.



Le schéma d'ordre 2 est du même ordre que le problème stationnaire elliptique. La solution par différences finies est donc très précise, car l'erreur de discrétisation est minime.