

# Méthodes Numériques

## Cours 4 :

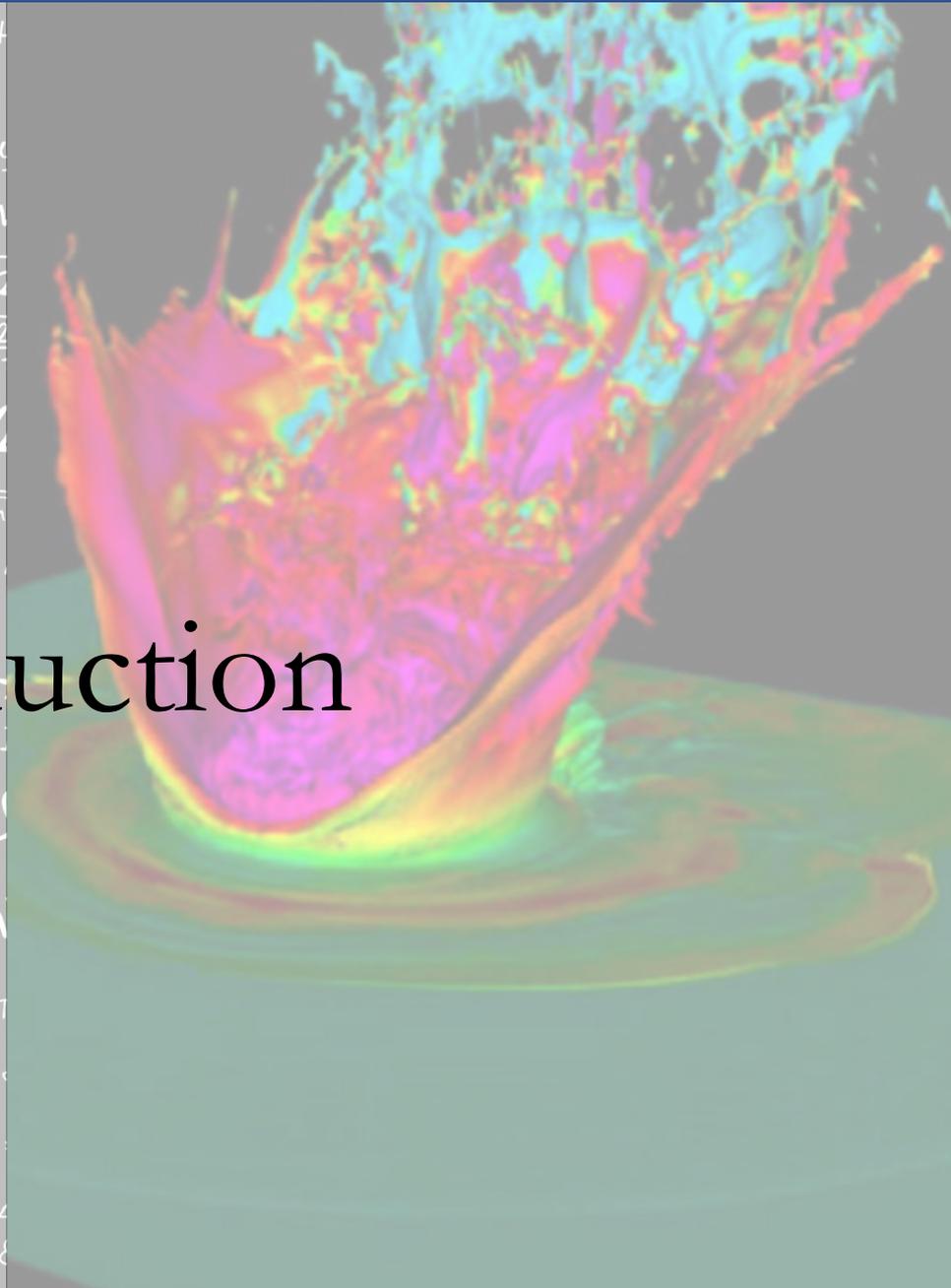
## Applications 1D

Dr. Barbara PERRI

[barbara.perri@universite-paris-saclay.fr](mailto:barbara.perri@universite-paris-saclay.fr)

$$\begin{aligned}
 & v_2 \operatorname{tg} \vartheta_B = \frac{w_2}{w_1} = w_{21} & \rho V = nRT & \vec{\Psi} = \iint \vec{D} d\vec{S} = AD \\
 & M_e = \sigma T^4 & \phi_e = \frac{L}{4\pi r^2} & \frac{\Delta\varphi}{2\pi} = \frac{\Delta x}{\lambda} = \frac{x_2 - x_1}{\lambda} & v = c/\lambda \\
 & \psi = E\psi & \Delta t = \frac{\Delta t'}{\sqrt{1 - v^2/c^2}} & X_L = \frac{U_m}{I_m} = \omega L = 2\pi f L & F_g = \frac{m_1 m_2}{(n_2 + n_1)^2} \\
 & E = \hbar\omega & U = \frac{W_{AB}}{|E_{PA} - E_{PB}|} = \frac{\varphi_A - \varphi_B}{T} = \frac{4 n_1 n_2}{(n_2 + n_1)^2} & \vec{v}_k = \sqrt{\frac{M_z}{R_z}} & \vec{F}_m = \vec{B} I l = \dots \\
 & \frac{M_m}{V_A} = \frac{M_r \cdot 10^{-3}}{N_A} & \varphi_E = \frac{E_c}{a} \int_{-a/L}^{+a/L} \sin(\omega t + \phi) dy & & \\
 & & l_t = l_0(1 + d\Delta t) & I = \frac{U_e}{R + R_i} & \omega = \dots \\
 & \overline{m_e} R = \rho \frac{\ell}{S} & E = mc^2 & \frac{\sin \alpha}{\sin \beta} = \frac{v_1}{v_2} = \frac{w_2}{w_1} & v = \frac{1}{\sqrt{\epsilon \cdot \mu}} \\
 & x) = \sqrt{2/L} \sin \frac{n\pi x}{L} & E = \frac{1}{2} \hbar \sqrt{k/m} & \beta = \frac{\Delta I_c}{\phi_e} = \frac{\Delta E}{\Delta t} \frac{w_1}{x} + \frac{w_2}{x'} = \dots \\
 & \iint \vec{J} d\vec{S} & \vec{S} = \frac{1}{\mu_0} (\vec{E} \times \vec{B}) & \Delta I_c = \frac{\hbar^2}{8mL^2} & \phi = \frac{2\pi \sin^2 \theta}{\lambda} \\
 & \frac{N_A}{m} = \sqrt{\frac{3R_m T}{M_r \cdot 10^{-3}}} & E = \hbar k^2 & 1 \text{ pc} = \frac{1 \text{ AU}}{r} & S = \frac{U}{I} & F_v = \dots \\
 & h = Sh\rho g & f_0 = \frac{1}{2\pi \sqrt{LC}} & \sigma = \frac{Q}{M} & M = Fd \cos \alpha & \\
 & \cos \vartheta_2 & R = R_0 \sqrt[3]{A} & \int \vec{E} d\vec{l} = - \iint \frac{\partial \vec{B}}{\partial t} \cdot d\vec{S} & \rho = \frac{E}{c} = \frac{\hbar f}{c} = \frac{\hbar}{\lambda} \\
 & \int_{C(S)} \vec{H} d\vec{l} = \iint_S (\vec{J} + \frac{\partial \vec{D}}{\partial t}) \cdot d\vec{S} & \varphi = mc\Delta t & F_g = \dots & \\
 & = \mu_0 \sum I_i & \rho = \frac{\vec{F}}{\Delta S} = \frac{m \Delta \vec{v}}{\Delta S \Delta t} & P = UI & h = \frac{1}{2} g t^2 & v = v_1(1 + \beta) \\
 & + \frac{\partial^2}{\partial x^2} & f' = \frac{v_a \cdot v_b}{(v_b - 1)(v_b - v_a)} & \nabla \times \left( -\frac{\partial \vec{B}}{\partial t} \right) = -\frac{\partial}{\partial t} (\operatorname{rot} \vec{B}) = -\mu_0 \frac{\partial}{\partial t} \left( \frac{\partial \vec{B}}{\partial t} \right) = \dots
 \end{aligned}$$

# Introduction



# Plan de l'UE

## Idée générale :

Au premier semestre, on va introduire les notions de base, et s'intéresser en détails à une méthode numérique précise

Tous les jeudi matin (8h45-12h45) au bâtiment 625

21 Novembre : Cours 1 + Cours 2

4 Décembre : Cours 3 + TP 1

5 Décembre : **Cours 4** + TP 2

12 Décembre : Cours 5 + TP 3

19 Décembre : Cours 6 + TP 4

9 Janvier : Cours 7 + TP 4

16 Janvier : Cours 8 + TP 5

23 Janvier : TP 5

30 Janvier : Examen

## Modalités d'évaluation :

TPs + examen oral (question de cours + exercice)

# Différences finies : Principe

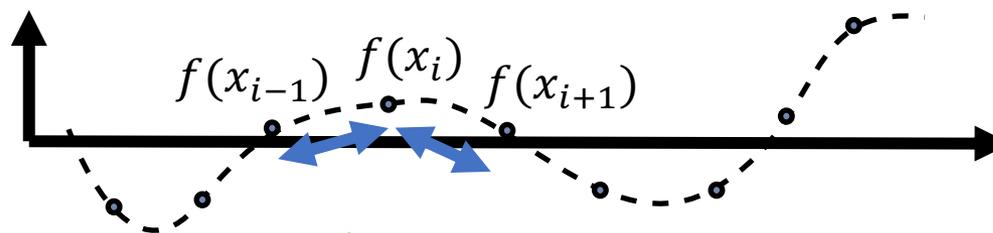
On va approximer les dérivées comme une différence de variations

→ En connaissant les valeurs voisines, on peut estimer la dérivée

$$\frac{df(x)}{dx} + f(x) = 0, \forall x \in D \quad \longrightarrow \quad \frac{f(x_{i+1}) - f(x_i)}{\delta x} + f(x_i) = 0, \forall i \in [0, N - 1]$$

→ À partir des équations et conditions aux limites, on écrit  $N$  équations algébriques  
(1 pour chaque point)

→ On résout chaque équation à partir des valeurs des points voisins



**Avantages :**

Formulation la plus intuitive et la plus simple, applicable à un grand nombre de problèmes

**Inconvénients :**

On ne prend pas en compte ce qui se passe entre les points

→ Très dépendant de la résolution choisie

# Différences finies : Méthodes

Pour trouver la formulation en différence finie, on peut utiliser 3 méthodes :



**La tangente**

$$\frac{df}{dx}(x) = \lim_{\epsilon \rightarrow 0} \frac{f(x + \epsilon) - f(x)}{\epsilon}$$

Ordres supérieurs :

Combinaison de Forward,  
Backward et Centré

**Avantages :**

Facile et intuitif

**Inconvénients :**

Pas systématique



**Les développements limités**

$$\frac{df}{dx}(x_i) = \sum_{k=0}^p A_k f(x_{i+k})$$

Ordres supérieurs :

Bonne combinaison  
linéaire de DLs

**Avantages :**

Précis

**Inconvénients :**

Difficile



**Le polynôme de Lagrange**

$$\frac{df}{dx}(x_i) \approx \frac{dP_L}{dx}(x_i)$$

Ordres supérieurs :

Dérivation du  
polynôme de Lagrange

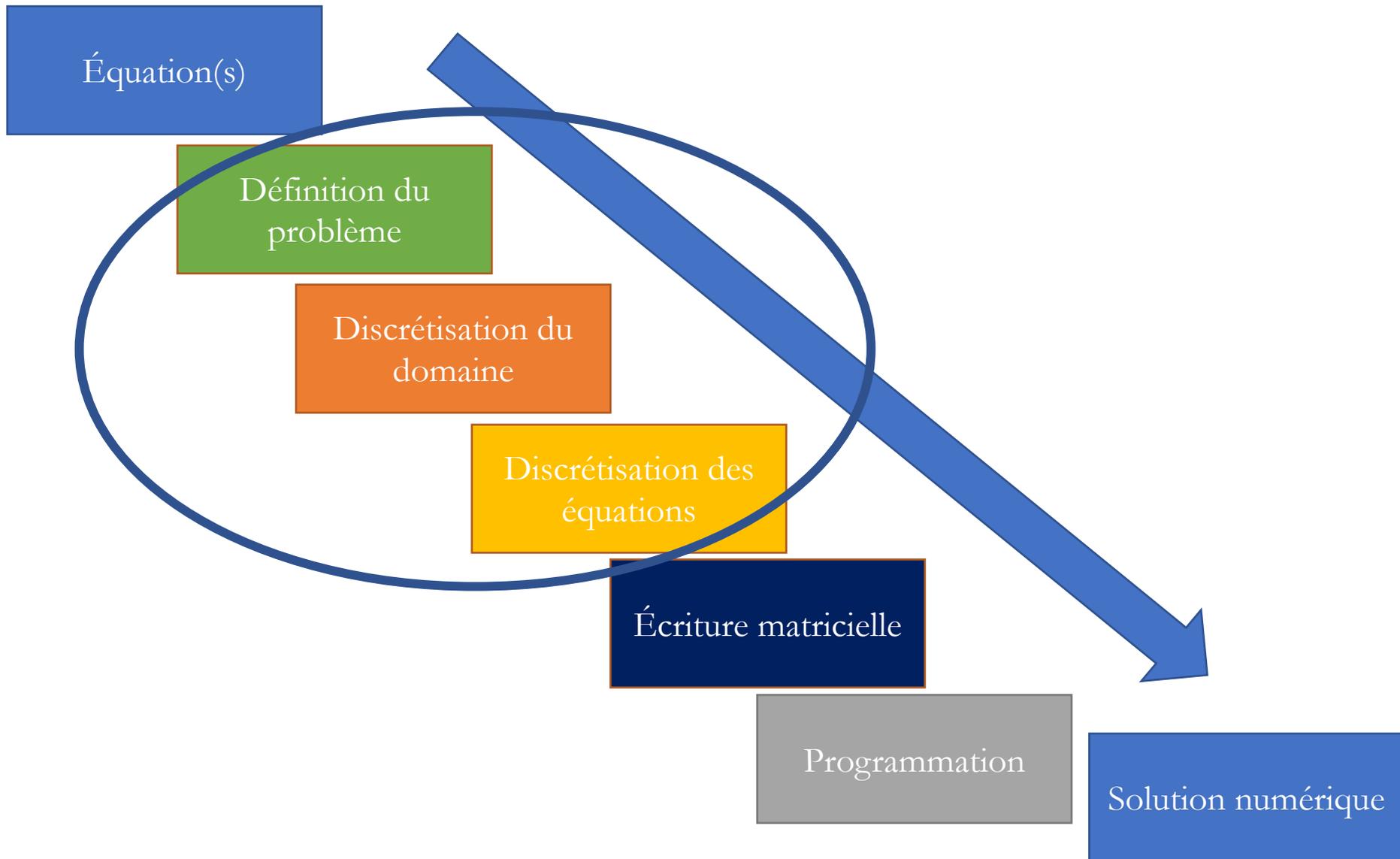
**Avantages :**

Systématique

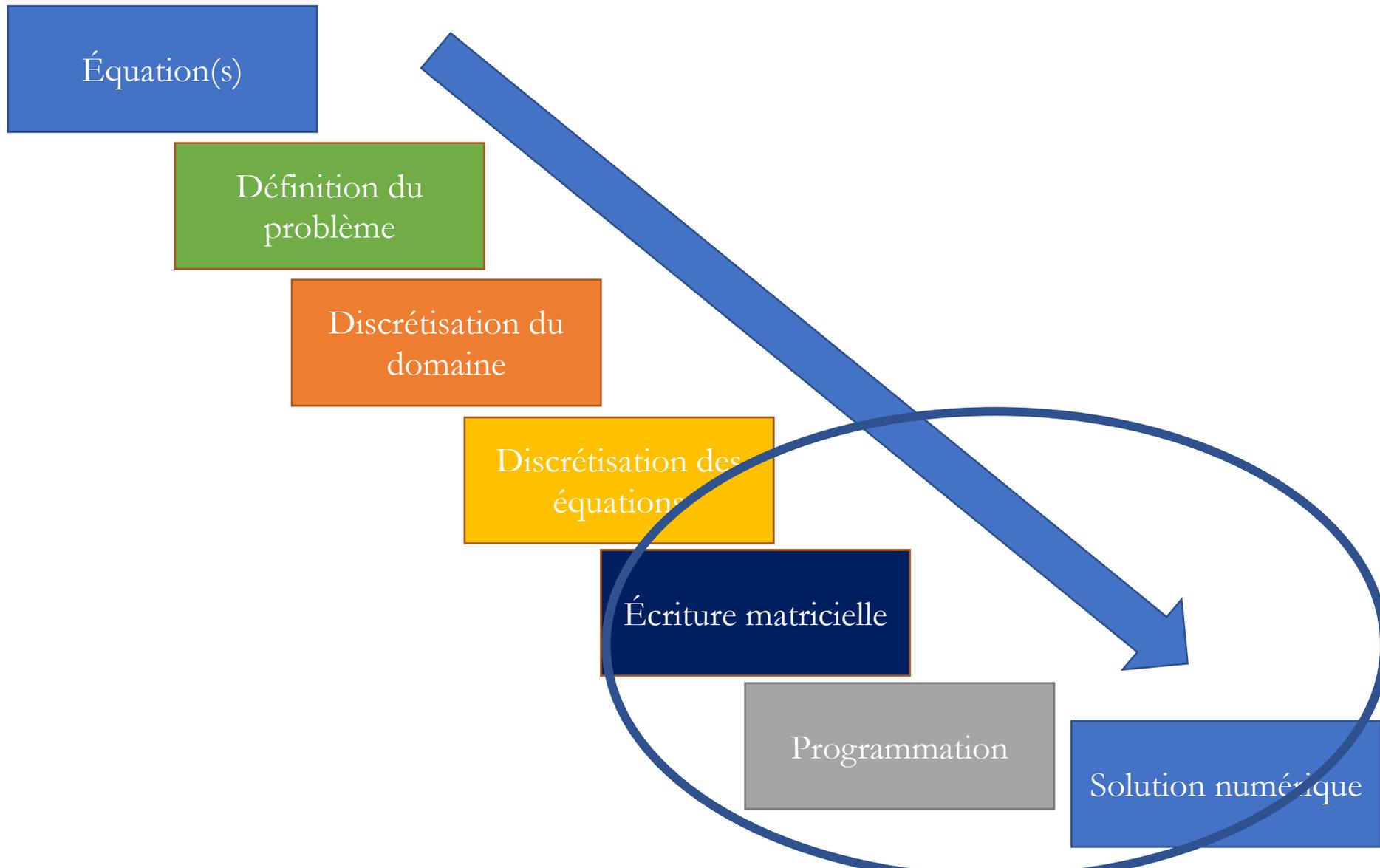
**Inconvénients :**

Peu précis

# Cycle des méthodes numériques



# Cycle des méthodes numériques

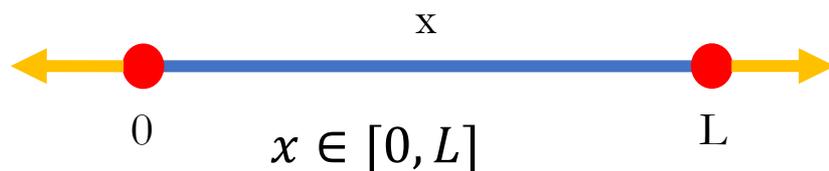


# Hypothèse : 1D

Pour cette première application, on ne va s'intéresser qu'à des problèmes à 1 dimension :



**Domaines**  
sous forme de segments



0

 $x \in [0, L]$ 

L

 $\delta D$  : 2 pointsGauche :  $x = 0$ Droite :  $x = L$ 

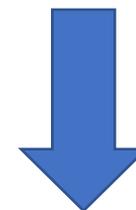
Cela permet de simplifier l'écriture du problème dans un premier temps



**Équations différentielles ordinaires (EDO)**

= les inconnues ne dépendent que d'une seule variable

$$\frac{df(x)}{dx} + f(x) = g(x)$$



MAIS la même méthode sera généralisable aux EDPs

# Hypothèse : Problèmes stationnaires

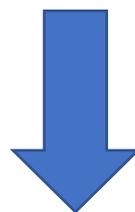
On rajoute l'hypothèse supplémentaire que les dérivées étudiées sont forcément par rapport à l'espace et non au temps :



Cela permet de se concentrer d'abord sur les conditions aux limites (avant d'ajouter la condition initiale)

+

Cela permet d'obtenir une seule solution finale stationnaire (pas d'évolution temporelle) → plus facile à valider



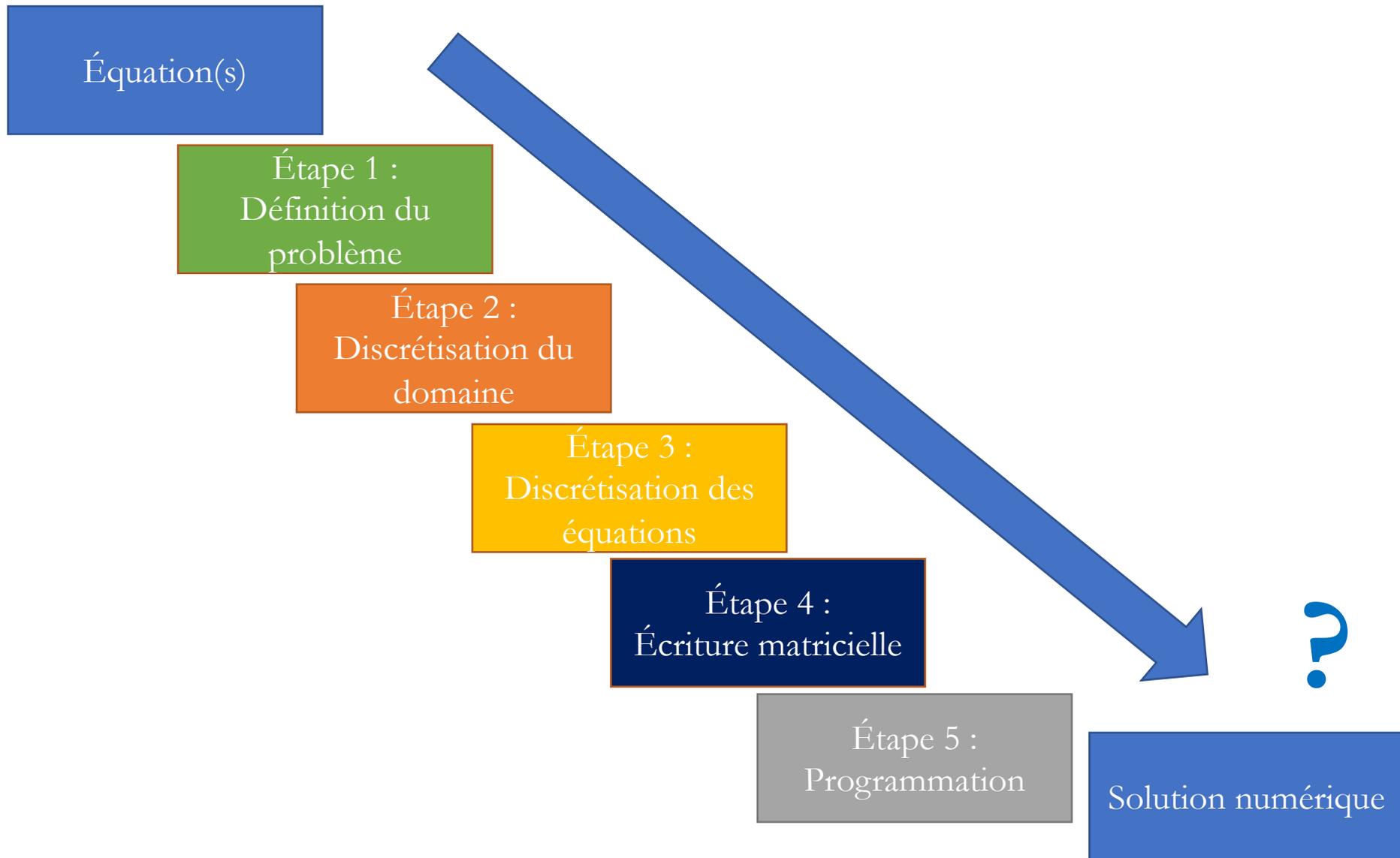
La dimension temporelle sera ajoutée à la séance 6 (EDP spatiale + temporelle)

$$\begin{aligned}
 & \nu = 2 \operatorname{tg} \vartheta_B = \frac{w_2}{w_1} = w_{21} \\
 & M_e = \sigma T^4 \\
 & \phi_e = \frac{L}{4\pi r^2} \\
 & E = h\nu \\
 & U = \frac{W_{AB}}{|E_{PA} - E_{PB}|} \\
 & \varphi_E = \frac{E_c}{k} \frac{\Phi}{r} \\
 & m = N \cdot m_0 = \frac{Q}{V_e} \frac{M_m}{N_A} \\
 & l_t = l_0(1 + d\Delta t) \\
 & I = \frac{U_e}{R + R_i} \\
 & R = \rho \frac{l}{S} \\
 & E = \frac{1}{2} h \sqrt{k/m} \\
 & \vec{S} = \frac{1}{\mu_0} (\vec{E} \times \vec{B}) \\
 & E = \frac{h k^2}{2m} \\
 & f_0 = \frac{1}{2\pi \sqrt{CL}} \\
 & \vec{H} d\vec{l} = \oint (\vec{J} + \frac{\partial \vec{D}}{\partial t}) \cdot d\vec{S} \\
 & \vec{H} = \mu_0 \sum \vec{I} \\
 & \rho = \frac{\vec{F}}{\Delta S} = \frac{m \Delta \vec{V}}{\Delta S \Delta t} \\
 & f' = \frac{v_a \cdot v_b}{(v_a - 1)(v_b - v_a)} \\
 & \nabla \times \left( -\frac{\partial \vec{B}}{\partial t} \right) = -\frac{\partial}{\partial t} (\operatorname{rot} \vec{B}) = -\mu_0 \frac{\partial}{\partial t} \left( \frac{\partial \vec{B}}{\partial t} \right) = \dots
 \end{aligned}$$

# Premier exemple



# Définition des étapes



# Étape 1 : Définition du problème

1

Avant de passer à la partie numérique, il faut s'assurer d'avoir bien défini le problème physique/mathématique continu

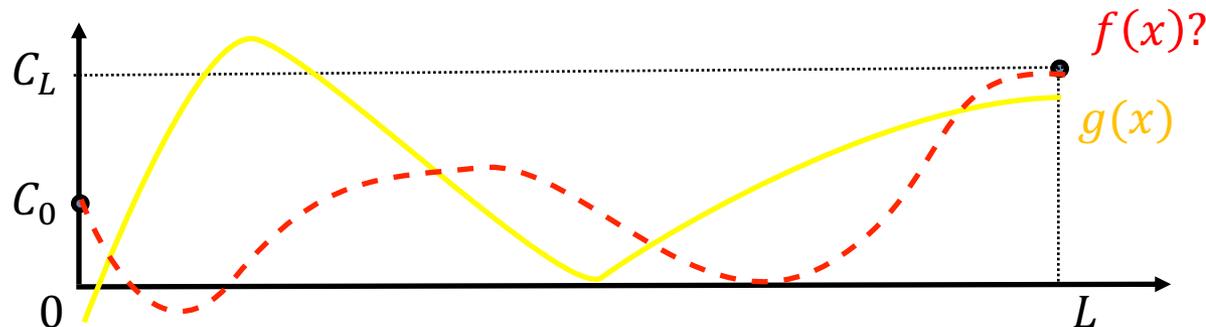
Premier exemple :  
Problème de Poisson

Équation :  $\frac{d^2 f}{dx^2}(x) = g(x)$  (problème de Poisson)

Inconnue :  $f(x)$  (fonction réelle 1D)

Domaine :  $x \in ]0, L[$  (segment 1D cartésien)

Condition aux limites :  $f(0) = C_0, f(L) = C_L$  (CL de type Dirichlet)



# Étape 2 : Discrétisation du domaine

2

On va maintenant passer d'un milieu continu à discret à l'aide d'un maillage :

Premier exemple :  
Maillage uniforme

Nombre de points :  $M + 1$  (résolution)

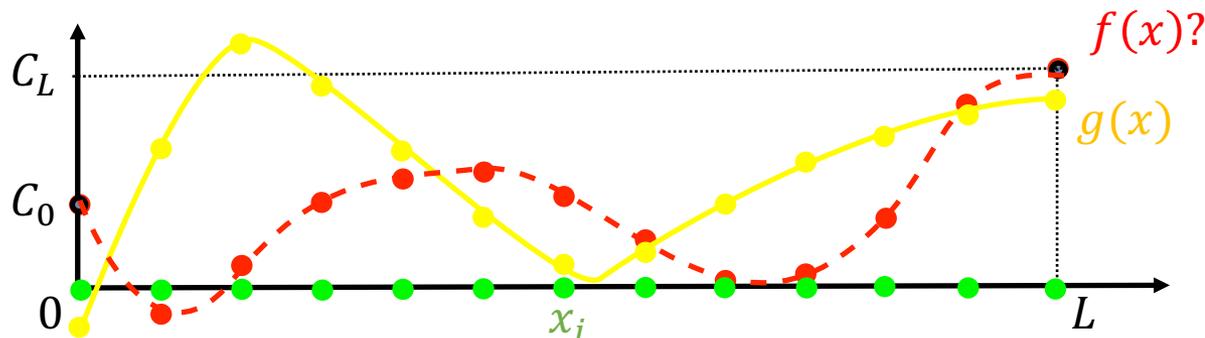
Pas d'espace :  $\delta x = L/M$

Numérotation :  $0 = x_0 < x_1 < x_2 < \dots < x_M = L$  (noeuds du maillage)  
 $x_i = i\delta x, \forall i \in [0, M]$

Valeurs nodales connues :  $g_0, g_1, g_2, \dots, g_M$  (valeurs aux noeuds)

Valeurs nodales inconnues :  $f_0, f_1, f_2, \dots, f_M$

Équivalence discret/continu :  $f(x_i) = f_i, g(x_i) = g_i$  (notation)

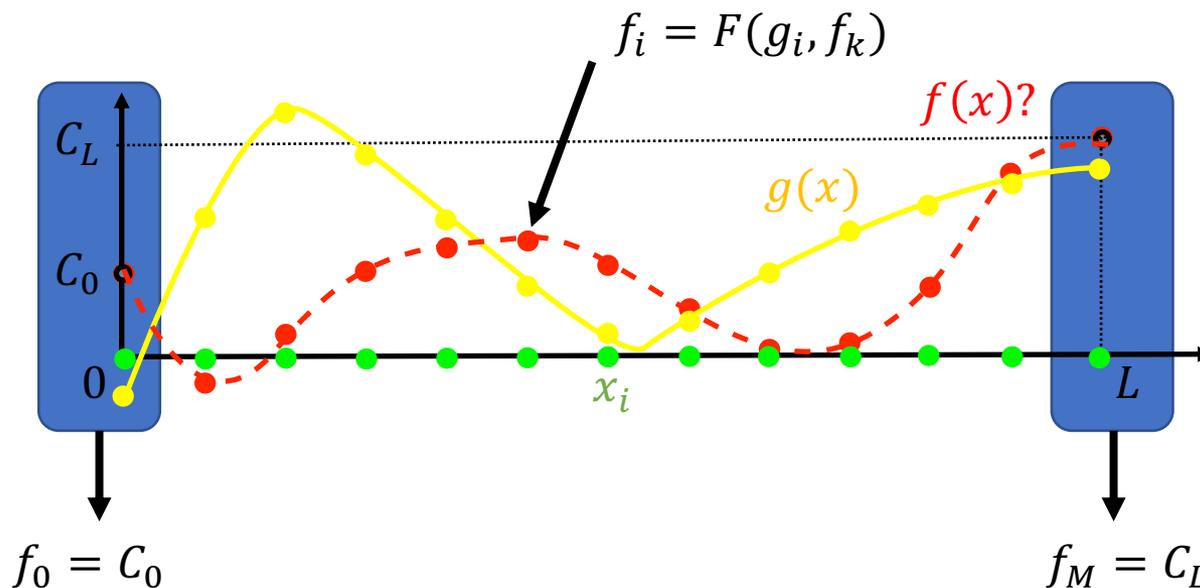


# Étape 3 : Discrétisation des équations (principe)

3

Une fois qu'on a discrétisé le domaine de calcul et les quantités physiques, il faut encore choisir comment approximer les dérivées = choix du schéma numérique

→ Le but est d'arriver à avoir une équation algébrique à résoudre pour chaque point du maillage :



→ Pour les bords, on écrit une représentation fidèle de la condition aux limites (immédiat avec Dirichlet) → en 1D, 2 équations supplémentaires

# Étape 3 : Discrétisation des équations (exemple)

3

Premier exemple :

Schéma centré pour dérivée seconde

On cherche à discrétiser l'équation suivante :

$$\frac{d^2 f}{dx^2}(x) = g(x) \quad \longrightarrow \quad \frac{d^2 f_i}{dx^2} = g_i$$

On peut alors utiliser les formules du chapitre précédent :

$$\frac{d^2 f_i}{dx^2} \approx f_{i+1} \left[ \frac{1}{(x_{i+1} - x_i)(x_i - x_{i-1})} \right] + f_i \left[ \frac{x_{i+1} - x_{i-1}}{(x_{i+1} - x_i)(x_i - x_{i-1})^2} \right] + f_{i-1} \left[ \frac{1}{(x_i - x_{i-1})^2} \right]$$

$$\longrightarrow \quad \frac{d^2 f_i}{dx^2} \approx \frac{f_{i+1} - 2f_i + f_{i-1}}{\delta x^2} \quad (\text{maillage uniforme})$$

$$\longrightarrow \quad \frac{f_{i+1} - 2f_i + f_{i-1}}{\delta x^2} = g_i \quad \longrightarrow \quad f_i = \frac{1}{2} (f_{i+1} + f_{i-1} - g_i \delta x^2)$$

$$\forall i \in ]0, M[$$

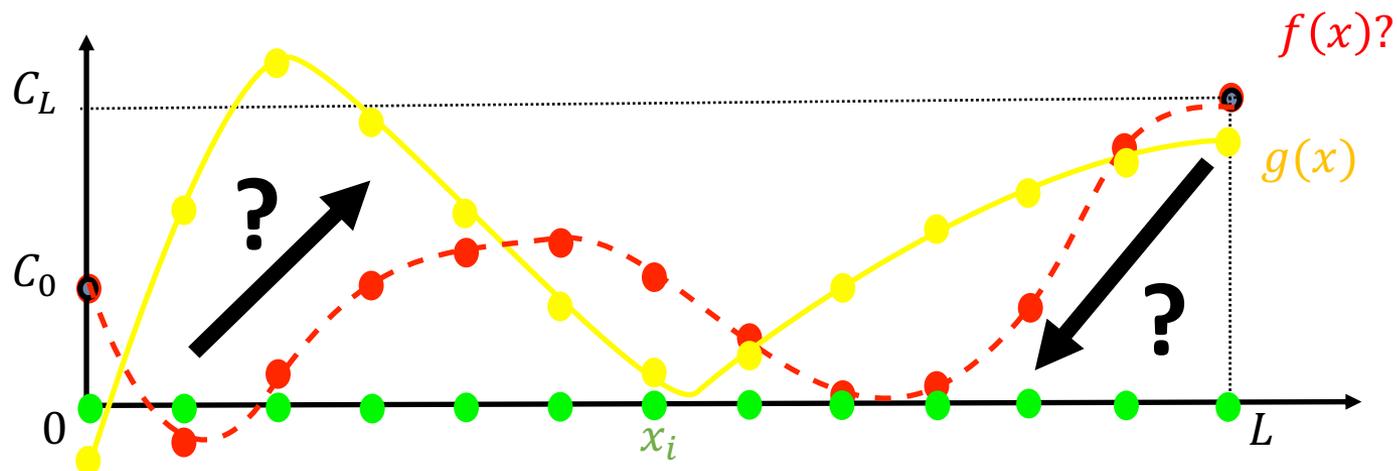
# Étape 4 : Écriture matricielle

4

On a donc finalement obtenu notre schéma de résolution numérique :

$$f_i = \frac{1}{2}(f_{i+1} + f_{i-1} - g_i \delta x^2)$$

→ On pourrait être tenté de résoudre le problème de manière itérative :



MAIS schéma implicite !! → la formulation matricielle est la plus universelle

# Étape 4 : Écriture matricielle (principe)

4

$$f_i = \frac{1}{2}(f_{i+1} + f_{i-1} - g_i \delta x^2)$$

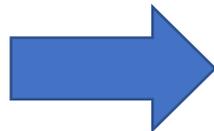


Schéma implicite !  
(besoin de points non calculés)

→ Au lieu d'écrire le problème sous forme itérative, on l'écrit sous forme matricielle :

$x_0$		$f_0 = C_0$						
$x_1$	[	$\frac{1}{\delta x^2}(f_0 - 2f_1 + f_2) = g_1$		$(M+1) \times (M+1)$ $AF = G$				
$x_2$		$\frac{1}{\delta x^2}(f_1 - 2f_2 + f_3) = g_2$						
...		...						
$x_i$		$\frac{1}{\delta x^2}(f_{i-1} - 2f_i + f_{i+1}) = g_i$						
...		...						
$x_{M-1}$		$\frac{1}{\delta x^2}(f_{M-2} - 2f_{M-1} + f_M) = g_{M-1}$						
$x_M$		$f_M = C_L$						
							$\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \dots \\ f_i \\ \dots \\ f_{M-1} \\ f_M \end{bmatrix}$	$\begin{bmatrix} C_0 \\ g_1 \\ g_2 \\ \dots \\ g_i \\ \dots \\ g_{M-1} \\ C_L \end{bmatrix}$

# Étape 4 : Écriture matricielle (exemple)

4

Premier exemple :

Schéma centré pour dérivée seconde

$$\begin{array}{l}
 x_0 \\
 x_1 \\
 x_2 \\
 \dots \\
 x_i \\
 \dots \\
 x_{M-1} \\
 x_M
 \end{array}
 \left[ \begin{array}{l}
 f_0 = C_0 \\
 \frac{1}{\delta x^2} (f_0 - 2f_1 + f_2) = g_1 \\
 \frac{1}{\delta x^2} (f_1 - 2f_2 + f_3) = g_2 \\
 \dots \\
 \frac{1}{\delta x^2} (f_{i-1} - 2f_i + f_{i+1}) = g_i \\
 \dots \\
 \frac{1}{\delta x^2} (f_{M-2} - 2f_{M-1} + f_M) = g_{M-1} \\
 f_M = C_L
 \end{array} \right]
 \quad \rightarrow \quad
 \begin{array}{l}
 x_0 \\
 x_1 \\
 x_2 \\
 \dots \\
 x_{M-1} \\
 x_M
 \end{array}
 \left[ \begin{array}{cccccc}
 f_0 & f_1 & f_2 & \dots & f_{M-1} & f_M \\
 1 & 0 & 0 & \dots & 0 & 0 \\
 1 & -2 & 1 & \dots & 0 & 0 \\
 \frac{1}{\delta x^2} & \frac{1}{\delta x^2} & \frac{1}{\delta x^2} & \dots & 0 & 0 \\
 0 & \frac{1}{\delta x^2} & \frac{-2}{\delta x^2} & \dots & 0 & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & 0 & \dots & \frac{-2}{\delta x^2} & \frac{1}{\delta x^2} \\
 0 & 0 & 0 & 0 & 0 & 1
 \end{array} \right]$$

# Étape 5 : Programmation

5

Il ne reste plus qu'à demander à l'ordinateur de résoudre le problème !

→ Sous forme matricielle :

$$AF = G \quad \rightarrow \quad F = A^{-1}G$$

Étapes informatiques :

1. Définir le vecteur  $G$ ,
2. Définir la matrice  $A$ ,
3. Inverser la matrice  $A$ ,
4. Multiplier par le vecteur  $G$  pour avoir la solution !

Premier exemple :

En Python, on pourra utiliser la librairie linalg de numpy pour calculer et inverser  $A$

# Au-delà : Validation



Bravo, vous avez obtenu une solution numérique de l'équation !

MAIS :

- ✓ Comment être sûr qu'il n'y a pas eu d'erreur au cours de la procédure ?
  - ✓ Comment être sûr que la solution est bien physique ?
- Dans des cas simples, on peut valider la solution numérique en la comparant à une solution analytique

Premier exemple :

Prenons la fonction suivante :  $f(x) = \sin(2x)$

→ On peut calculer à la main sa dérivée seconde :  $\frac{d^2f}{dx^2} = -4 \sin(2x) = g(x)$

→ On peut imposer des conditions aux limites naturelles :

$$C_0 = f(0) = \sin(0) = 0, C_L = f(L) = \sin(2L)$$

→ Si on impose  $g(x) = -4 \sin(2x)$  et qu'on utilise les conditions aux limites ci-dessus, la solution de l'équation devrait naturellement être la fonction cible

→ On peut alors tester si la procédure fonctionne correctement !

# Exemple avec code (I)

1

Étape 1 : Définition du problème

Domaine :

$$x \in ]0, L[$$

```
#Taille du domaine  
L=10
```

Conditions aux limites :

$$f(0) = C_0, f(L) = C_L$$

```
# valeurs aux bords  
C_0=0  
C_L=np.sin(2*L)
```

Fonction connue :

$$g(x)$$

```
# définition de la fonction g(x)  
def gfun(x):  
    g=-4*np.sin(2*x)  
    return g
```

# Exemple avec code (II)

2

Étape 2 : Discrétisation du domaine

Nombre de points :  $M + 1$

```
# résolution  
M=5000
```

Pas d'espace :  $\delta x = L/M$

```
# pas d'espace  
dx=L/M
```

Numérotation :  $0 = x_0 < x_1 < x_2 < \dots < x_M = L$

```
#maillage de pts équidistants  
x=np.linspace(0,L,M+1)
```

Valeurs nodales connues :  $g_0, g_1, g_2, \dots, g_M$

```
#g se crée le plus facilement sans boucle  
g=gfun(x)
```

# Exemple avec code (III)

3

4

Étape 3 & 4 : Discrétisation des équations et Écriture matricielle  
(la discrétisation des équations se retrouve dans la forme de la matrice)

Définition du vecteur  $G$  :

```
#g se crée le plus facilement sans boucle  
g=gfun(x)  
g[0]=C_0 #corriger premier point  
g[M]=C_L #corriger dernier point
```

Définition de la matrice  $A$  :

```
a=1/dx**2
```

```
#la matrice A est principalement remplie de zéros et de taille M+1,M+1  
A=np.zeros([M+1,M+1])  
  
A[0,0]=1 #première et dernière ligne ont 1 sur la diagonale  
A[M,M]=1  
  
#les autres lignes se remplissent à l'aide d'une boucle  
for j in range(1,M): # pour j allant de 1 à M-1  
    A[j,[j-1,j,j+1]]=[a,-2*a,a]
```

# Exemple avec code (IV)

5



## Étape 5 : Programmation et Validation

Calcul de la solution :

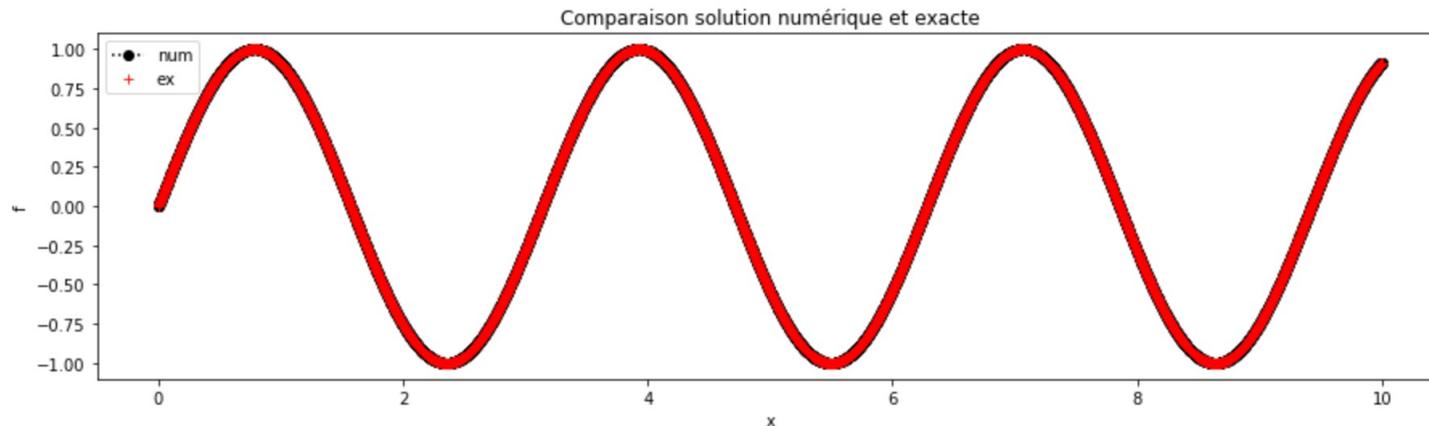
```
# 5. Solution du système linéaire
#####
# On trouve f la solution de Af = g.

f=np.linalg.solve(A,g)
```

Comparaison avec solution  
analytique :

```
#solution exacte
f_ex=np.sin(2*x)
errmax=np.max(np.abs(f-f_ex))
```

L'erreur maximale est : 2.3863663177747796e-06



Question : Que peut-on faire pour faire diminuer l'erreur ?



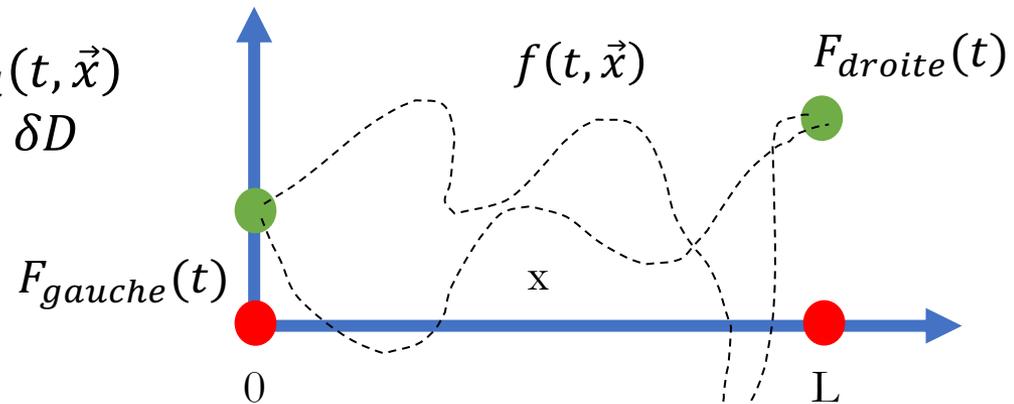
# Influence de la condition aux limites (I)

Rappel de cours :

**Dirichlet :**

On fixe la valeur de la fonction sur le bord

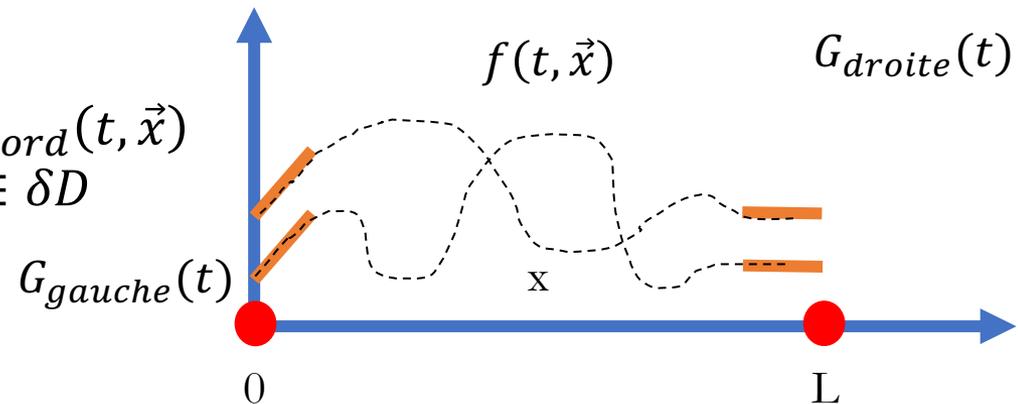
$$f(t, \vec{x}) = F_{bord}(t, \vec{x}) \\ \forall t \in I_t, \forall \vec{x} \in \delta D$$



**Neuman :**

On fixe la valeur de la dérivée de la fonction sur le bord

$$\vec{n} \cdot \vec{\nabla} f(t, \vec{x}) = G_{bord}(t, \vec{x}) \\ \forall t \in I_t, \forall \vec{x} \in \delta D$$



→ Que se passe-t-il si on utilise une condition de Neuman au lieu de Dirichlet ?

# Influence de la condition aux limites (II)

1

Étape 1 : Définition du problème

Deuxième exemple :

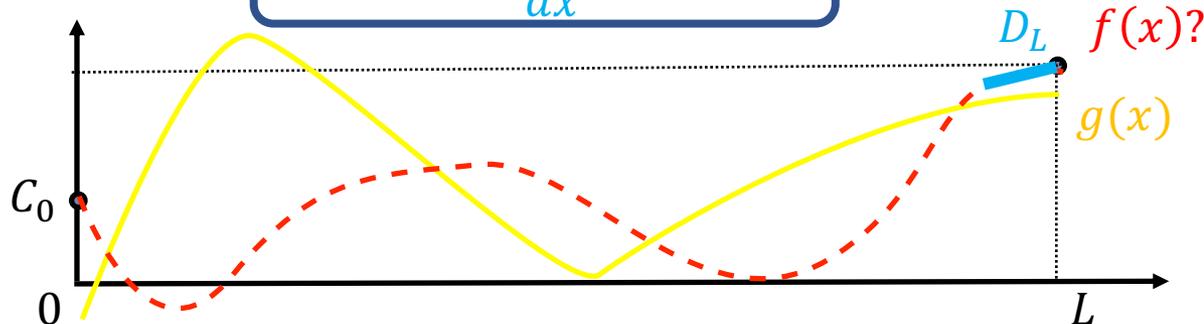
Problème de Poisson + conditions de Neuman

Équation :  $\frac{d^2 f}{dx^2}(x) = g(x)$  (problème de Poisson)

Inconnue :  $f(x)$  (fonction réelle 1D)

Domaine :  $x \in ]0, L[$  (segment 1D cartésien)

Condition aux limites :  $f(0) = C_0, \frac{df}{dx}(L) = D_L$  (CL de type Neuman)



# Influence de la condition aux limites (III)

Étape 2 : Discrétisation du domaine

Deuxième exemple :

Problème de Poisson + conditions de Neuman

Nombre de points :  $M + 1$  (résolution)

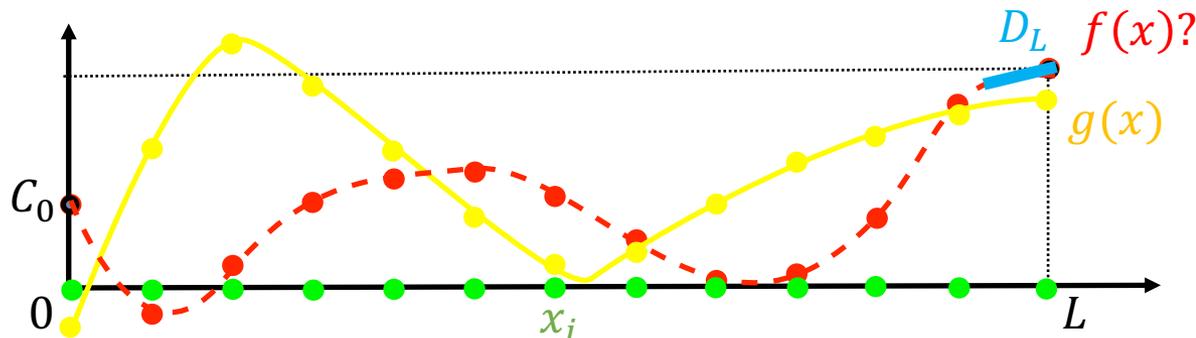
Pas d'espace :  $\delta x = L/M$

Numérotation :  $0 = x_0 < x_1 < x_2 < \dots < x_M = L$  (noeuds du maillage)  
 $x_i = i\delta x, \forall i \in [0, M]$

Valeurs nodales connues :  $g_0, g_1, g_2, \dots, g_M$  (valeurs aux noeuds)

Valeurs nodales inconnues :  $f_0, f_1, f_2, \dots, f_M$

Équivalence discret/continu :  $f(x_i) = f_i, g(x_i) = g_i$  (notation)



# Influence de la condition aux limites (IV)

3

Étape 3 : Discrétisation des équations

Deuxième exemple :

Problème de Poisson + conditions de Neuman

(maillage uniforme)

$$\frac{d^2 f}{dx^2}(x) = g(x) \quad \longrightarrow \quad \frac{f_{i+1} - 2f_i + f_{i-1}}{\delta x^2} = g_i \quad \forall i \in ]0, M[$$

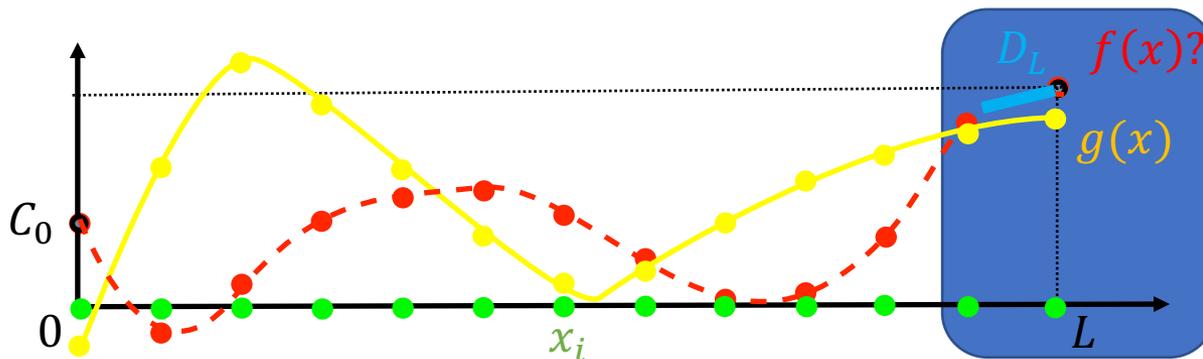
On complète avec les conditions aux limites :

$$f_0 = C_0$$

$$\frac{df_M}{dx} = D_L$$

→ Il faut également approximer la dérivée première pour la condition de Neuman !

NB : on préférera une formulation cohérente avec le reste des équations  
(même précision) → Attention aux points choisis !



$$\frac{df_2}{dx} = D_L$$

(Polynôme de Lagrange)

$$\frac{1}{2\delta x} (f_0 - 4f_1 + 3f_2) = D_L$$

# Influence de la condition aux limites (V)

4

Étape 4 : Écriture matricielle

Deuxième exemple :

Problème de Poisson + conditions de Neuman

$$\begin{array}{l}
 x_0 \\
 x_1 \\
 x_2 \\
 \dots \\
 x_i \\
 \dots \\
 x_{M-1} \\
 x_M
 \end{array}
 \left[ \begin{array}{l}
 f_0 = C_0 \\
 \frac{1}{\delta x^2} (f_0 - 2f_1 + f_2) = g_1 \\
 \frac{1}{\delta x^2} (f_1 - 2f_2 + f_3) = g_2 \\
 \dots \\
 \frac{1}{\delta x^2} (f_{i-1} - 2f_i + f_{i+1}) = g_i \\
 \dots \\
 \frac{1}{\delta x^2} (f_{M-2} - 2f_{M-1} + f_M) = g_{M-1} \\
 \frac{1}{2\delta x} (f_{M-2} - 4f_{M-1} + 3f_M) = D_L
 \end{array} \right]
 \quad \longrightarrow \quad
 \begin{array}{l}
 x_0 \\
 x_1 \\
 x_2 \\
 \dots \\
 x_{M-1} \\
 x_M
 \end{array}
 \left[ \begin{array}{cccccc}
 f_0 & f_1 & f_2 & \dots & f_{M-1} & f_M \\
 1 & 0 & 0 & \dots & 0 & 0 \\
 \frac{1}{\delta x^2} & \frac{-2}{\delta x^2} & \frac{1}{\delta x^2} & \dots & 0 & 0 \\
 0 & \frac{1}{\delta x^2} & \frac{-2}{\delta x^2} & \dots & 0 & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & 0 & \dots & \frac{-2}{\delta x^2} & \frac{1}{\delta x^2} \\
 0 & 0 & 0 & \dots & \frac{1}{2\delta x} & \frac{-4}{2\delta x} & \frac{3}{2\delta x}
 \end{array} \right]$$

# Influence de la condition aux limites (VI)

5



Étape 5 & 6 : Programmation et Validation

Même concept pour la programmation, pas de changement :

$$AF = G \quad \longrightarrow \quad F = A^{-1}G$$

Pour la validation, on propose la même fonction que précédemment :

Deuxième exemple :

Prenons la fonction suivante :  $f(x) = \sin(2x)$

→ On peut calculer à la main sa dérivée seconde :  $\frac{d^2f}{dx^2} = -4 \sin(2x) = g(x)$

→ On peut imposer des conditions aux limites naturelles :

$$C_0 = f(0) = \sin(0) = 0 \quad D_L = \frac{df}{dx}(L) = 2\cos(2L)$$

→ Si on impose  $g(x) = -4 \sin(2x)$  et qu'on utilise **les conditions aux limites ci-dessus**, la solution de l'équation devrait naturellement être la fonction cible

→ On peut alors tester si la procédure fonctionne correctement !

# Influence de la condition aux limites (VI)

Exemple avec code :

Conditions aux limites :

```
# valeurs aux bords
C_0=0
C_L=np.sin(2*L)
```

```
# valeurs aux bords
C_0=0
D_L=2*np.cos(2*L)
```

Définition du vecteur G :

```
#g se crée le plus facilement sans boucle
g=gfun(x)
g[0]=C_0 #corriger premier point
g[M]=D_L #corriger dernier point
```

Définition de la matrice A :

```
a=1/dx**2
b=1/(2*dx)
```

```
#la matrice A est principalement remplie de zéros et de taille M+1,M+1
A=np.zeros([M+1,M+1])

#première et dernière ligne
A[0,0]=1
A[M,[M-2,M-1,M]] = [b,-4*b,3*b]

#les autres lignes se remplissent à l'aide d'une boucle
for j in range(1,M): # pour j allant de 1 à M-1
    A[j,[j-1,j,j+1]]=a
```

# Influence du schéma (I)

Rappel de cours :

Une fois qu'on a choisi la méthode numérique qu'on souhaite utiliser, il reste encore de nombreux choix possibles = schéma numérique

$$\text{ex : différences finies} \rightarrow \frac{df(x)}{dx} + f(x) = 0$$

→ comment approximer la dérivée ? quelle formulation ?



$$\frac{df(x_i)}{dx} \approx \frac{f(x_{i+1}) - f(x_i)}{\delta x}$$

Schéma d'Euler Forward



$$\frac{df(x_i)}{dx} \approx \frac{f(x_i) - f(x_{i-1})}{\delta x}$$

Schéma d'Euler Backward



$$\frac{df(x_i)}{dx} \approx \frac{f(x_{i+1}) - f(x_{i-1}))}{2\delta x}$$

Schéma d'Euler centré

NB : on a vu en différences finies que le choix du schéma a un impact sur la précision de l'approximation et sur les points sélectionnés

# Influence du schéma (II)

1

Étape 1 : Définition du problème

Troisième exemple :

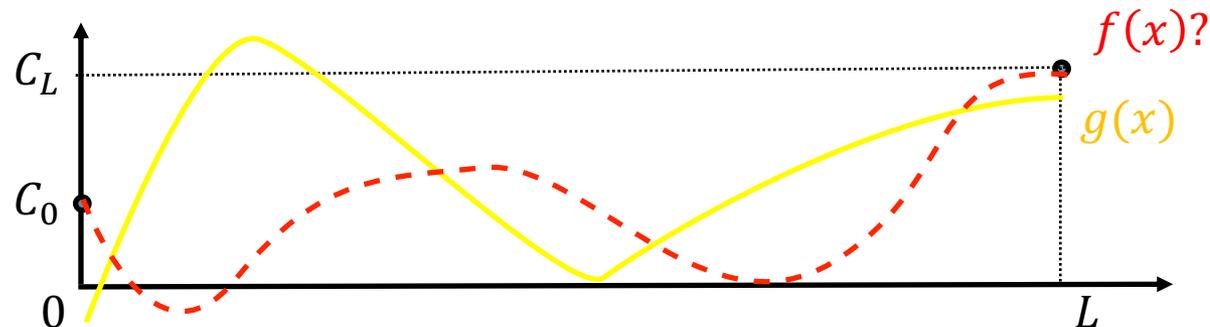
Problème de Poisson + ordre supérieur

Équation :  $\frac{d^2 f}{dx^2}(x) = g(x)$  (problème de Poisson)

Inconnue :  $f(x)$  (fonction réelle 1D)

Domaine :  $x \in ]0, L[$  (segment 1D cartésien)

Condition aux limites :  $f(0) = C_0, f(L) = C_L$  (CL de type Dirichlet)



# Influence du schéma (III)

2

Étape 2 : Discrétisation du domaine

Troisième exemple :

Problème de Poisson + ordre supérieur

Nombre de points :  $M + 1$  (résolution)

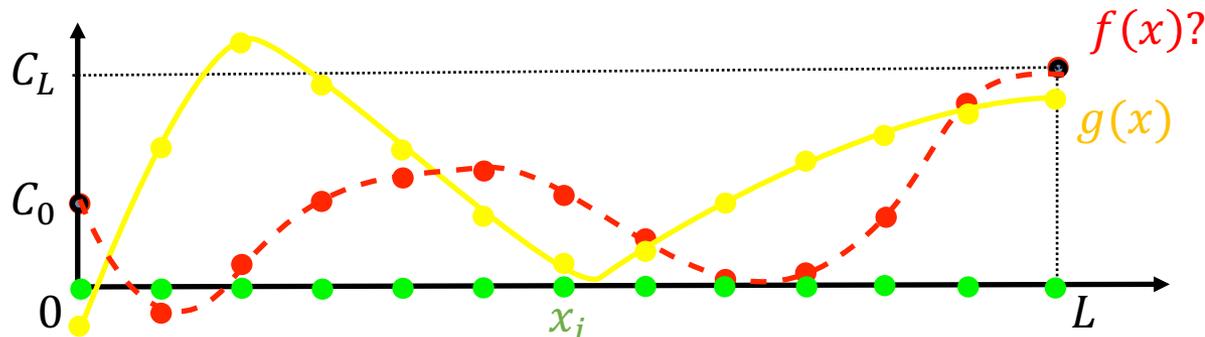
Pas d'espace :  $\delta x = L/M$

Numérotation :  $0 = x_0 < x_1 < x_2 < \dots < x_M = L$  (noeuds du maillage)  
 $x_i = i\delta x, \forall i \in [0, M]$

Valeurs nodales connues :  $g_0, g_1, g_2, \dots, g_M$  (valeurs aux noeuds)

Valeurs nodales inconnues :  $f_0, f_1, f_2, \dots, f_M$

Équivalence discret/continu :  $f(x_i) = f_i, g(x_i) = g_i$  (notation)



# Influence du schéma (IV)

3

Étape 3 : Discrétisation des équations

Troisième exemple :

Problème de Poisson + ordre supérieur

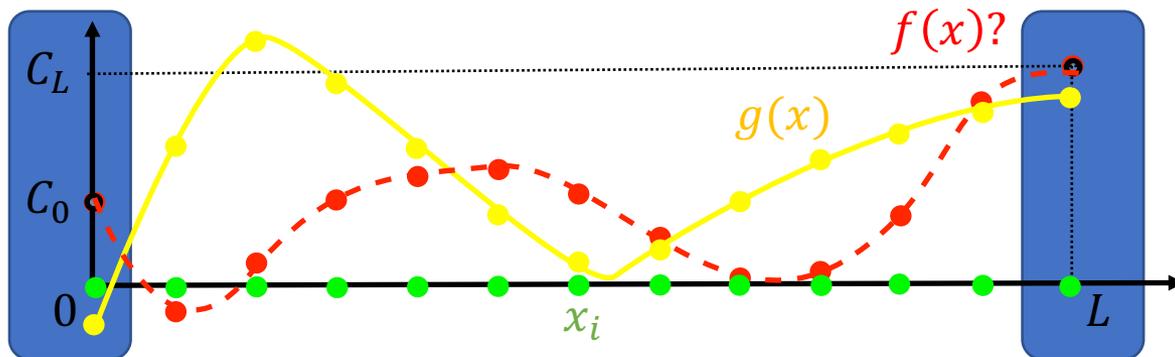
$$\frac{d^2 f}{dx^2}(x) = g(x) \quad \longrightarrow \quad \boxed{\frac{-f_{i+2} + 16f_{i+1} - 30f_i + 16f_{i-1} - f_{i-2}}{12\delta x^2}} = g_i \quad \forall i \in ]0, M[ \quad (\text{maillage uniforme})$$

NB : on peut prendre une formulation avec une précision différente !

On complète avec les conditions aux limites :

$$f_0 = C_0$$

$$f_M = C_L$$



Le schéma ne marche pas en  $i = 1$  et  $i = M - 1$  !

→ Il faut d'autres CLs :

$$f_1 = C_1$$

$$f_{M-1} = C_{L1}$$

# Influence du schéma (V)

4

Étape 4 : Écriture matricielle

Troisième exemple :

Problème de Poisson + ordre supérieur

$$\begin{array}{l}
 x_0 \\
 x_1 \\
 x_2 \\
 \dots \\
 x_i \\
 \dots \\
 x_{M-1} \\
 x_M
 \end{array}
 \left[ \begin{array}{l}
 f_0 = C_0 \\
 f_1 = C_1 \\
 \frac{1}{12\delta x^2} (-f_0 + 16f_1 - 30f_2 + 16f_3 - f_4) = g_2 \\
 \dots \\
 \frac{1}{12\delta x^2} (-f_{i-2} + 16f_{i-1} - 30f_i + 16f_{i+1} - f_{i+2}) = g_i \\
 \dots \\
 f_{M-1} = C_{L1} \\
 f_M = C_L
 \end{array} \right.
 \begin{array}{l}
 f_0 \quad f_1 \quad f_2 \quad \dots \quad f_{M-2} \quad f_{M-1} \quad f_M \\
 x_0 \left[ \begin{array}{cccccccc}
 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\
 0 & -a & 16a & -30a & 16a & -a & \dots & 0 \\
 \dots & \dots \\
 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array} \right]
 \end{array}
 \right.$$


# Influence du schéma (VI)

Exemple avec code :

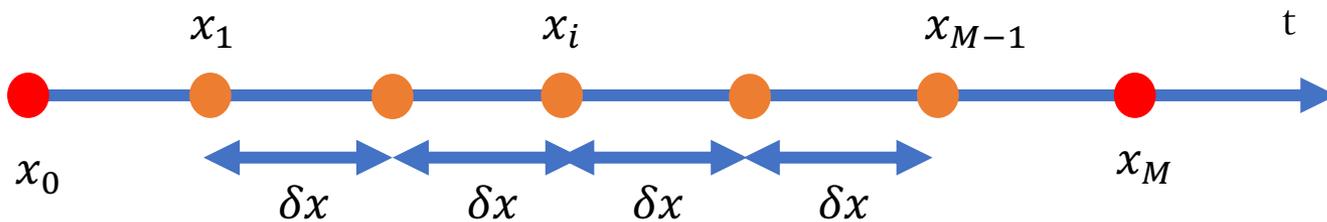
```
# valeurs aux bords  
C_0=0  
C_1=np.sin(2*dx)  
C_L=np.sin(2*L)  
C_L1=np.sin(2*(L-dx))
```

```
#première et dernière ligne  
A[0,0]=1  
A[1,1]=1  
A[M-1,M-1]=1  
A[M,M]=1  
  
#les autres lignes se remplissent à l'aide d'une boucle  
for j in range(2,M-1): # pour j allant de 1 à M-1  
    A[j, [j-2, j-1, j, j+1, j+2]] = [-a, 16*a, -30*a, 16*a, -a]
```

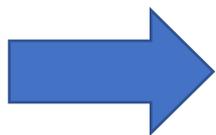
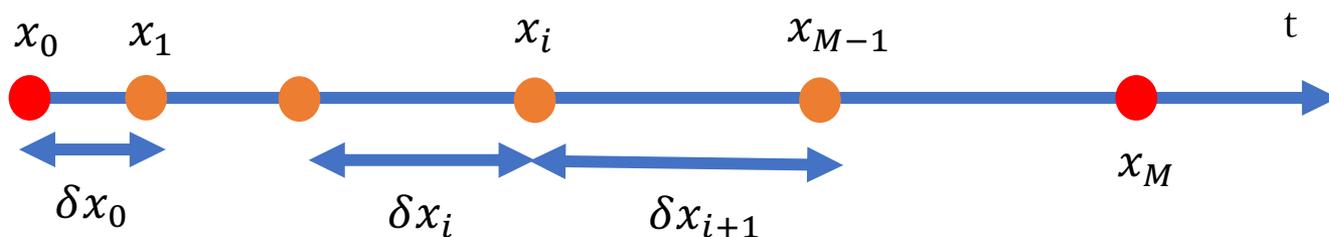
# Influence du maillage (I)

Rappel de cours :

Un maillage est dit uniforme (ou régulier)  
quand l'espacement entre les points du maillage est toujours le même :



Un maillage est dit irrégulier  
quand l'espacement entre les points du maillage est différent entre chaque point :



$\delta x = \text{constante}$



$\delta x_i \neq \delta x_{i+1}, \forall i \in [0, M - 1]$

# Influence du maillage (II)

1

Étape 1 : Définition du problème

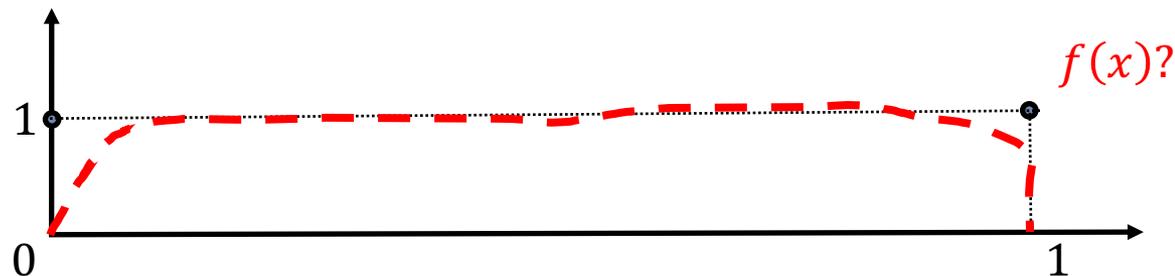
Quatrième exemple :  
Problème de couche limite

Équation :  $-\epsilon^2 \frac{d^2 f}{dx^2}(x) + f(x) = 1, \epsilon \ll 1$  (problème de couche limite)

Inconnue :  $f(x)$  (fonction réelle 1D)

Domaine :  $x \in [0,1]$  (segment 1D cartésien)

Condition aux limites :  $f(0) = 0, f(1) = 0$  (CL de type Dirichlet)



# Influence du maillage (III)

2

Étape 2 : Discrétisation du domaine

Quatrième exemple :

Problème de couche limite

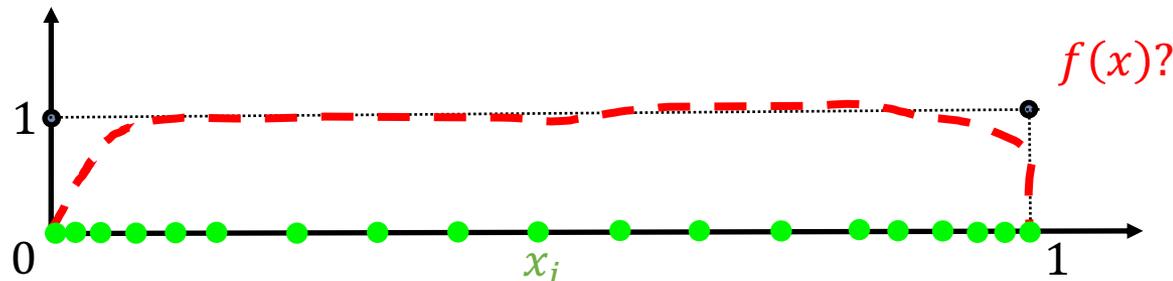
Nombre de points :  $M + 1$  (résolution)

Loi d'espacement :  $x_i = \frac{1}{2} - \frac{1}{2} \cos\left(\frac{(2i-1)\pi}{2(M-1)}\right), \forall i \in [1, M-1]$  (noeuds du maillage)

Numérotation :  $0 = x_0 < x_1 < x_2 < \dots < x_M = L$

Valeurs nodales inconnues :  $f_0, f_1, f_2, \dots, f_M$  (valeurs aux noeuds)

Équivalence discret/continu :  $f(x_i) = f_i$  (notation)



# Influence du maillage (IV)

3

Étape 3 : Discrétisation des équations

Quatrième exemple :

Problème de couche limite

(maillage uniforme)

$$-\epsilon^2 \frac{d^2 f}{dx^2}(x) + f(x) = 1 \quad \longrightarrow \quad -\epsilon^2 \frac{f_{i+1} - 2f_i + f_{i-1}}{\delta x^2} + f_i = 1 \quad \forall i \in ]0, M[$$

→ On ne peut plus utiliser cette formule !

→ On utilise alors l'approximation par Polynôme de Lagrange :

$$\frac{d^2 P_L}{dx^2}(x) = f_0 \frac{2}{(x_0 - x_1)(x_0 - x_2)} + f_1 \frac{2}{(x_1 - x_0)(x_1 - x_2)} + f_2 \frac{2}{(x_2 - x_0)(x_2 - x_1)}$$

$$-\epsilon^2 \left[ \frac{2f_{i-1}}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})} + \frac{2f_i}{(x_i - x_{i-1})(x_i - x_{i+1})} + \frac{2f_{i+1}}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} \right] + f_i = 1$$

On complète avec les conditions aux limites :

$$f_0 = 0$$

$$f_M = 0$$

# Influence du maillage (V)

4

Étape 4 : Écriture matricielle

Quatrième exemple :

Problème de couche limite

$$\delta x_{i-1} = (x_i - x_{i-1})$$

$$\delta x_i = (x_{i+1} - x_i)$$

$$\delta x_{i+1} = (x_{i+1} - x_i)$$

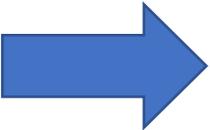
$$\begin{array}{l}
 x_0 \\
 x_1 \\
 x_2 \\
 \dots \\
 x_i \\
 \dots \\
 x_{M-1} \\
 x_M
 \end{array}
 \left[ \begin{array}{l}
 f_0 = 0 \\
 -\epsilon^2 \left[ \frac{2f_0}{\delta x_0 \delta x_2} - \frac{2f_1}{\delta x_0 \delta x_1} + \frac{2f_2}{\delta x_2 \delta x_1} \right] + f_1 = 1 \\
 -\epsilon^2 \left[ \frac{2f_1}{\delta x_1 \delta x_3} - \frac{2f_2}{\delta x_1 \delta x_2} + \frac{2f_3}{\delta x_3 \delta x_2} \right] + f_2 = 1 \\
 \dots \\
 -\epsilon^2 \left[ \frac{2f_{i-1}}{\delta x_{i-1} \delta x_{i+1}} - \frac{2f_i}{\delta x_{i-1} \delta x_i} + \frac{2f_{i+1}}{\delta x_{i+1} \delta x_i} \right] + f_i = 1 \\
 \dots \\
 -\epsilon^2 \left[ \frac{2f_{M-2}}{\delta x_{M-2} \delta x_M} - \frac{2f_{M-1}}{\delta x_{M-2} \delta x_{M-1}} + \frac{2f_M}{\delta x_M \delta x_{M-1}} \right] + f_{M-1} = 1 \\
 f_M = 0
 \end{array} \right.$$

# Influence du maillage ( $V$ )

4

Étape 4 : Écriture matricielle

Quatrième exemple :  
Problème de couche limite



$$\begin{array}{c}
 x_0 \\
 x_1 \\
 x_2 \\
 \dots \\
 x_{M-1} \\
 x_M
 \end{array}
 \begin{array}{c}
 f_0 \quad f_1 \quad f_2 \quad \dots \quad f_{M-1} \quad f_M \\
 \left[ \begin{array}{ccccccc}
 0 & 0 & 0 & \dots & 0 & 0 & 0 \\
 \frac{-2\epsilon^2}{\delta x_0 \delta x_2} & \frac{-2\epsilon^2}{\delta x_0 \delta x_1} + 1 & \frac{-2\epsilon^2}{\delta x_2 \delta x_1} & \dots & 0 & 0 & 0 \\
 0 & \frac{-2\epsilon^2}{\delta x_1 \delta x_3} & \frac{-2\epsilon^2}{\delta x_1 \delta x_2} + 1 & \dots & 0 & 0 & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & 0 & \dots & \frac{-2\epsilon^2}{\delta x_{M-2} \delta x_{M-1}} + 1 & \frac{-2\epsilon^2}{\delta x_M \delta x_{M-1}} & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array} \right]
 \end{array}$$

# Influence du maillage (VI)

5



Étape 5 & 6 : Programmation et Validation

Même concept pour la programmation, pas de changement :

$$AF = G \quad \longrightarrow \quad F = A^{-1}G$$

Pour la validation, il faut trouver une autre fonction solution de ce problème :

Quatrième exemple :

Prenons la fonction suivante :  $f(x) = 1 - \frac{\sinh\left(\frac{x}{\epsilon}\right)}{\sinh\left(\frac{1}{\epsilon}\right)} + \frac{\sinh\left(\frac{x-1}{\epsilon}\right)}{\sinh\left(\frac{1}{\epsilon}\right)}$

→ On vérifie les conditions aux limites :

$$f(0) = 1 - \frac{\sinh(0)}{\sinh\left(\frac{1}{\epsilon}\right)} + \frac{\sinh\left(-\frac{1}{\epsilon}\right)}{\sinh\left(\frac{1}{\epsilon}\right)} = 1 - 0 - 1 = 0, \quad f(1) = 1 - \frac{\sinh\left(\frac{1}{\epsilon}\right)}{\sinh\left(\frac{1}{\epsilon}\right)} + \frac{\sinh(0)}{\sinh\left(\frac{1}{\epsilon}\right)} = 1 - 1 + 0 = 0$$

# Influence du maillage (VII)

Exemple avec code :

Maillage :

```
## 3. Maillage
#####

x=np.zeros(M+1)
x[0]=0
x[M]=1
for j in range(1,M): #pour j de 1 a M-1
    x[j]=0.5-0.5*np.cos((2*j-1)*np.pi/(2*(M-1)))
```

Matrice A :

```
#première et dernière ligne 0
A[0,0]=1
A[M,M]=1

#les autres lignes se remplissent à l'aide d'une boucle
#on utilise la formule générale
for j in range(1,M): # pour j allant de 1 à M-1
    A[j,j-1]=-2*epsil**2/((x[j-1]-x[j])*(x[j-1]-x[j+1]))
    A[j,j]=-2*epsil**2/((x[j]-x[j-1])*(x[j]-x[j+1]))+1
    A[j,j+1]=-2*epsil**2/((x[j+1]-x[j-1])*(x[j+1]-x[j]))
```

$$\begin{aligned}
 & 2 \operatorname{tg} \vartheta_B = \frac{w_2}{w_1} = w_{21} & \rho V = n R T & \vec{\psi} = \iint \vec{D} d\vec{S} = A D & v = c/\lambda \\
 & M_e = \sigma T^4 & \phi_e = \frac{L}{4\pi r^2} & \int \frac{\Delta \varphi}{2\pi} = \frac{\Delta x}{\lambda} = \frac{x_2 - x_1}{\lambda} & v_k = \sqrt{\frac{M_z}{R_z}} & \vec{F}_m = \vec{B} I l = \dots \\
 & \vec{J} = E \vec{\psi} & \Delta t = \frac{\Delta t'}{\sqrt{1 - v^2/c^2}} & X_L = \frac{U_m}{I_m} = \omega L = 2\pi f L & \vec{F}_g = \frac{m_1 m_2}{(n_2 + n_1)^2} & \vec{g} = \frac{m_1 m_2}{r^2} \\
 & E = h\nu & U = \frac{W_{AB}}{|E_{PA} - E_{PB}|} = \frac{\varphi_A - \varphi_B}{T} = \frac{4 n_1 n_2}{(n_2 + n_1)^2} & & & R_m = \frac{c}{T} k = \pm \sqrt{\dots} \\
 & E = \frac{h c}{\lambda} & \varphi_E = \frac{E_c}{\varphi_0} = k \frac{\varphi}{r} & & & \\
 & \frac{M_m}{V_A} = \frac{M_r \cdot 10^{-3}}{N_A} & m = N \cdot m_0 = \frac{Q}{V_e} \frac{M_m}{N_A} & E = \frac{E_c}{a} \int_{-a/L}^{+a/L} \sin(\omega t + \phi) dy & & \\
 & & l_t = l_0(1 + d \Delta t) & I = \frac{U_e}{R + R_i} & \omega = \dots & \\
 & \overline{m_e} R = \rho \frac{h}{S} & E = m c^2 & & & \\
 & \langle x \rangle = \sqrt{2/L} \sin \frac{n\pi x}{L} & & & & \\
 & \iint \vec{J} d\vec{S} & \vec{S} = \frac{1}{\mu_0} (\vec{E} \times \vec{B}) & \Delta I_B & \phi = \frac{2\pi \sin^2 \vartheta}{\lambda} & \\
 & \frac{N_A}{m} = \sqrt{\frac{3 R_m T}{M_r \cdot 10^{-3}}} & E = \frac{h^2 k^2}{2m} & 1 \text{ pc} = \frac{1 \text{ AU}}{r} & S = \frac{U}{I} & \vec{F}_V = \dots \\
 & h = S h \rho g & f_0 = \frac{1}{2\pi \sqrt{LC}} & \sigma = \frac{Q}{M} & M = F d \cos \alpha & \\
 & \cos \vartheta_2 & & & & \\
 & R = R_0 \sqrt[3]{A} & \int \vec{E} d\vec{l} = - \iint \frac{\partial \vec{B}}{\partial t} \cdot d\vec{S} & \rho = \frac{E}{c} = \frac{h f}{c} = \frac{h}{\lambda} & \omega = U_m \sin \omega(t - T) = U_m \sin 2\pi t & \\
 & \oint \vec{H} d\vec{l} = \iint (\vec{J} + \frac{\partial \vec{D}}{\partial t}) \cdot d\vec{S} & \varphi = m c \Delta t & F_g = \dots & & \\
 & & L = 10 \log \frac{I}{I_0} & \Delta \psi = \frac{2\pi \Delta x}{\lambda} = \frac{2\pi d \sin \vartheta}{\lambda} & & \\
 & = \mu_0 \sum I_i & \rho = \frac{\vec{F}}{\Delta S} = \frac{m \Delta v}{\Delta S \Delta t} & P = UI & h = \frac{1}{2} g t^2 & v = v_1(1 + \beta) \\
 & + \frac{\partial^2}{\partial x^2} & f' = \frac{v_a \cdot v_b}{(v - 1)(v_0 - v_a)} & \nabla \times \left( -\frac{\partial \vec{B}}{\partial t} \right) = -\frac{\partial}{\partial t} (\operatorname{rot} \vec{B}) = -\mu_0 \frac{\partial}{\partial t} \left( \frac{\partial \vec{B}}{\partial t} \right) = \dots & & 
 \end{aligned}$$

# Problème plus général



# Définition d'un problème aux valeurs propres

On appelle **problème aux valeurs propres** le problème suivant :

Soient  $A$  et  $B$  deux matrices carrées de dimension  $n \times n$

Soit  $\lambda$  un scalaire

Problème :

Trouver le vecteur  $V$  de dimension  $n$  qui vérifie :

$$AV = \lambda BV$$

→ On appelle alors  $V$  le vecteur propre de  $A$  et  $B$

→ On appelle alors  $\lambda$  la valeur propre de  $A$  et  $B$

# Étape 1 : Définition du problème

1

Exemple général :

Problème aux valeurs propres

Équation :  $-\frac{d^2 f}{dx^2}(x) = \lambda f(x)$  (problème de Poisson)

Inconnue :  $f(x)$  (fonction réelle 1D)

Domaine :  $x \in [0, L]$  (segment 1D cartésien)

Condition aux limites :  $f(0) = 0, f(L) = 0$  (CL de type Dirichlet)

NB : la problématique d'un problème aux valeurs propres est plus large que trouver  $f$   
→ On veut également trouver les  $\lambda$  pour lesquels il existe une solution  $f$  non nulle !

# Étape 2 : Discrétisation du domaine

2

## Exemple général :

Problème aux valeurs propres

On va repasser sur un maillage uniforme :

Nombre de points :  $M + 1$  (résolution)

Pas d'espace :  $\delta x = L/M$

Numérotation :  $0 = x_0 < x_1 < x_2 < \dots < x_M = L$  (noeuds du maillage)  
 $x_i = i\delta x, \forall i \in [0, M]$

Valeurs nodales inconnues :  $f_0, f_1, f_2, \dots, f_M$  (valeurs aux noeuds)

Équivalence discret/continu :  $f(x_i) = f_i$  (notation)

# Étape 3 : Discrétisation des équations

3

Exemple général :

Problème aux valeurs propres

On cherche à discrétiser l'équation suivante :

$$-\frac{d^2 f}{dx^2}(x) = \lambda f(x) \quad \longrightarrow \quad \frac{d^2 f_i}{dx^2} = \lambda f_i$$

$$\frac{d^2 f_i}{dx^2} \approx \frac{f_{i+1} - 2f_i + f_{i-1}}{\delta x^2} \quad \longrightarrow \quad -\frac{f_{i+1} - 2f_i + f_{i-1}}{\delta x^2} = \lambda f_i \quad \forall i \in ]0, M[$$

(maillage uniforme)

On complète avec les conditions aux limites :

$$f_0 = 0$$

$$f_M = 0$$

# Étape 4 : Écriture matricielle (principe)

4

$$-\frac{f_{i+1} - 2f_i + f_{i-1}}{\delta x^2} = \lambda f_i$$

$$\begin{array}{l}
 x_0 \\
 x_1 \\
 x_2 \\
 \dots \\
 x_i \\
 \dots \\
 x_{M-1} \\
 x_M
 \end{array}
 \left[ \begin{array}{l}
 f_0 = 0 \\
 \frac{1}{\delta x^2} (f_0 - 2f_1 + f_2) = \lambda f_1 \\
 \frac{1}{\delta x^2} (f_1 - 2f_2 + f_3) = \lambda f_2 \\
 \dots \\
 \frac{1}{\delta x^2} (f_{i-1} - 2f_i + f_{i+1}) = \lambda f_i \\
 \dots \\
 \frac{1}{\delta x^2} (f_{M-2} - 2f_{M-1} + f_M) = \lambda f_{M-1} \\
 f_M = 0
 \end{array} \right.
 \xrightarrow{\quad}
 \begin{array}{l}
 (M+1) \times (M+1) \\
 AF = \boxed{\lambda BF} \\
 \left[ \begin{array}{c}
 f_0 \\
 f_1 \\
 f_2 \\
 \dots \\
 f_i \\
 \dots \\
 f_{M-1} \\
 f_M
 \end{array} \right] \\
 (M+1) \times (M+1)
 \end{array}$$



# Étape 5 : Programmation

5

## Exemple général :

Problème aux valeurs propres

Il ne reste plus qu'à demander à l'ordinateur de résoudre le problème !

→ Sous forme matricielle :

$$AF = \lambda BF \quad \longrightarrow \quad \lambda? F?$$

Étapes informatiques :

1. Définir la matrice A,
2. Définir la matrice B,
3. Trouver les valeurs propres,
4. Trouver les vecteurs propres F associés.

## Premier exemple :

En Python, on pourra utiliser la librairie linalg de numpy pour calculer les valeurs et vecteurs propres

# Validation



## Exemple général :

Problème aux valeurs propres

En fait, on cherche la solution analytique de :

$$\frac{d^2 f}{dx^2} + \lambda f = 0 \quad \longrightarrow \quad \lambda = k^2 \quad \longrightarrow \quad \frac{d^2 f}{dx^2} + k^2 f = 0$$

→ C'est une équation hyperbolique qui admet une solution ondulatoire !

Solution :

$$f(x) = A \sin(kx) + B \cos(kx)$$

On trouve les constantes grâce aux conditions aux limites :

$$f(0) = 0 \Rightarrow B = 0 \qquad f(L) = 0 \Rightarrow A \sin(kL) = 0 \Rightarrow k = k_n = \frac{n\pi}{L}, n \in \mathbb{N}$$

Les fonctions propres sont donc :

$$f_n(x) = \sin\left(\frac{n\pi x}{L}\right), \lambda_n = \frac{n^2 \pi^2}{L^2}$$

$$\begin{aligned} L &= \pi \\ k_n &= n \\ \lambda_n &= n^2 \end{aligned}$$

# Exemple avec code

```
#4. Matrices A et B
A=np.zeros([M+1,M+1]) #la matrice A est principalement remplie de zéros et de taille M+1,M+1
B=np.eye(M+1) #la matrice B est principalement une matrice identité

A[0,0]=1 #première et dernière ligne ont 1 sur la diagonale
A[M,M]=1
B[0,0]=0 #première et dernière ligne ont 0 sur la diagonale
B[M,M]=0

#les autres lignes se remplissent à l'aide d'une boucle
for j in range(1,M): # pour j allant de 1 à M-1
    A[j,[j-1,j,j+1]]=[-a,2*a,-a]

#5. Calculer les valeurs et vecteurs propres
lam,v=eig(A,B)

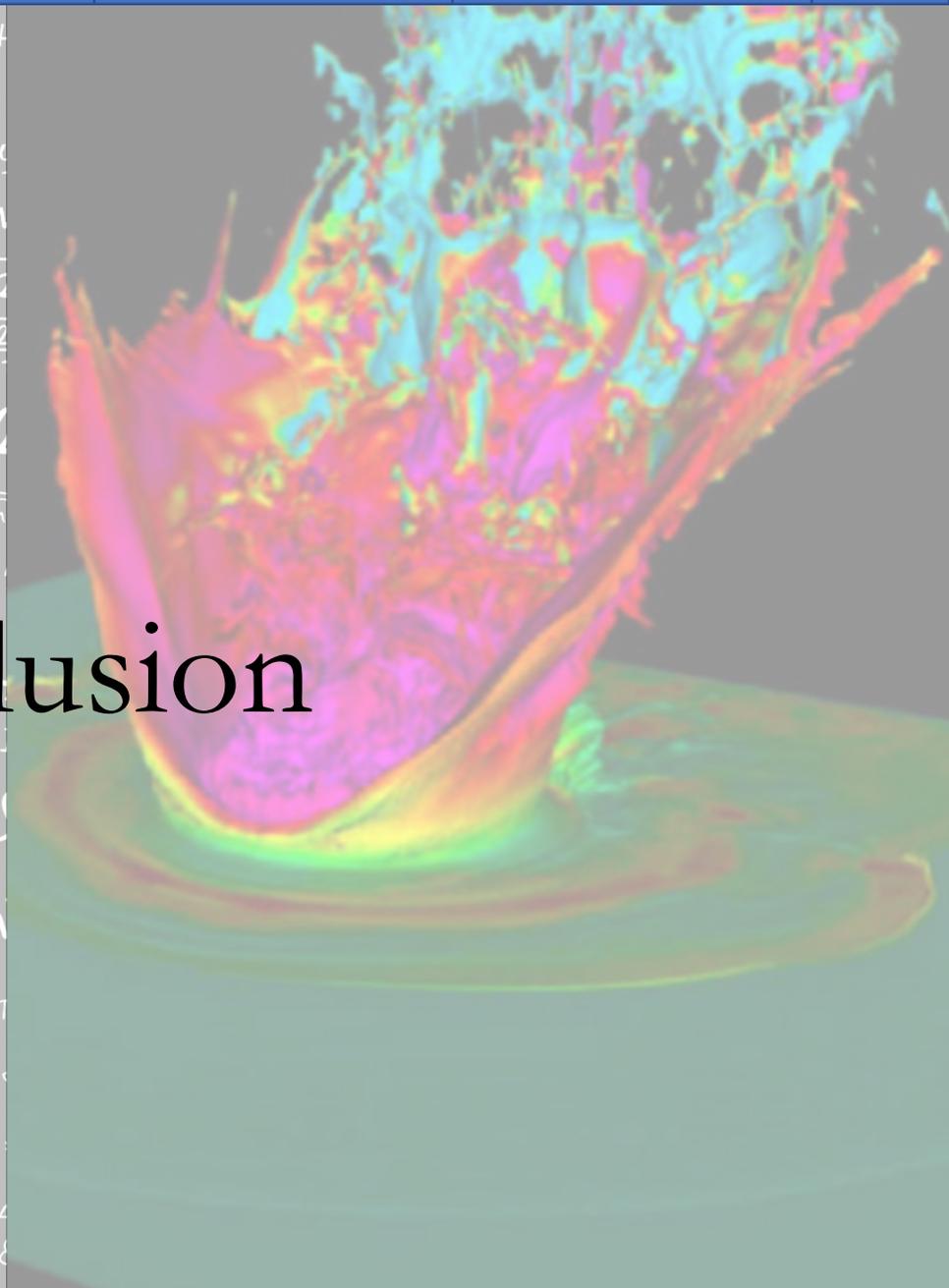
#ecarter les valeurs propres inf
test=lam!=np.inf
lam=lam[test]
v=v[:,test]

#ordonner les valeurs propres (ordre croissant)
ind=np.argsort(np.real(lam)) #on récupère les indices de permutation

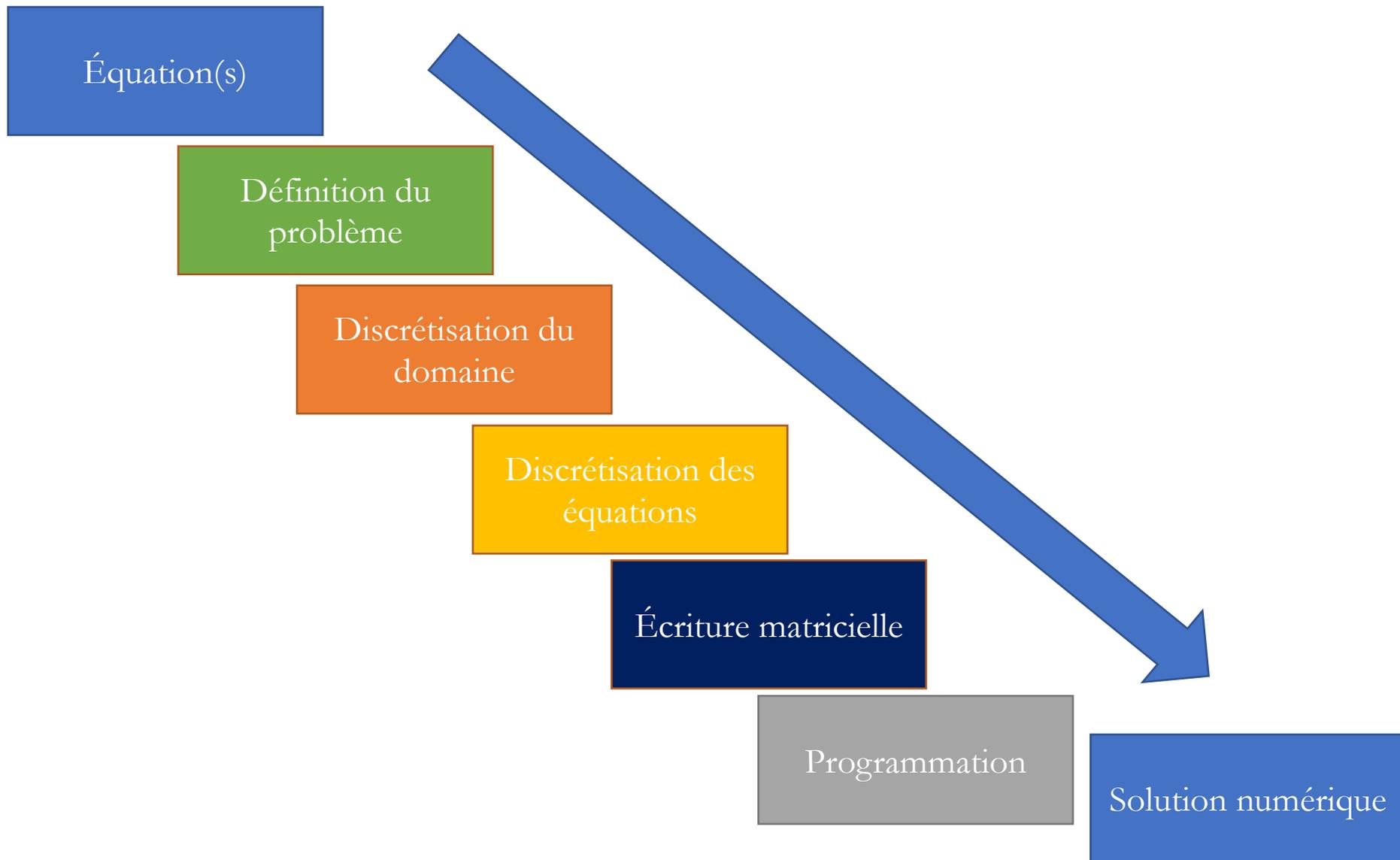
#on réordonne
lam=lam[ind]
v=v[:,ind]
```

$$\begin{aligned}
 & 2 \operatorname{tg} \vartheta_B = \frac{w_2}{w_1} = w_{21} & \rho V = nRT & \vec{\psi} = \iint \vec{D} d\vec{S} = AD \\
 & M_e = \sigma T^4 & \phi_e = \frac{L}{4\pi r^2} & \frac{\Delta\varphi}{2\pi} = \frac{\Delta x}{\lambda} = \frac{x_2 - x_1}{\lambda} & v = c/\lambda \\
 & \psi = E\psi & \Delta t = \frac{\Delta t'}{\sqrt{1 - v^2/c^2}} & X_L = \frac{U_m}{I_m} = \omega L = 2\pi f L & F_g = \frac{m_1 m_2}{(n_2 + n_1)^2} \\
 & E = \hbar\omega & U = \frac{W_{AB}}{|E_{PA} - E_{PB}|} = \frac{|\varphi_A - \varphi_B|}{T} = \frac{4 n_1 n_2}{(n_2 + n_1)^2} & \vec{v}_k = \sqrt{\frac{M_z}{R_z}} & \vec{F}_m = \vec{B} I l = \dots \\
 & \frac{M_m}{V_A} = \frac{M_r \cdot 10^{-3}}{N_A} & \varphi_E = \frac{E_c}{\rho_0} = k \frac{\varphi}{r} & \vec{g} = \frac{c}{T} & k = \pm \sqrt{\dots} \\
 & m = N \cdot m_0 = \frac{\varphi}{V_e} \frac{M_m}{N_A} & E = \frac{E_c}{a} \int_{-a/L}^{+a/L} \sin(\omega t + \phi) dy & R_m = \frac{c}{T} & k = \pm \sqrt{\dots} \\
 & l_t = l_0(1 + d \Delta t) & I = \frac{U_e}{R + R_i} & \omega = \dots & \\
 & R = \rho \frac{l}{S} & E = mc^2 & \frac{\sin \alpha}{\sin \beta} = \frac{v_1}{v_2} = \frac{w_2}{w_1} & v = \frac{1}{\sqrt{\epsilon \cdot \mu}} \\
 & \vec{x} = \sqrt{2/L} \sin \frac{n\pi x}{L} & E = \frac{1}{2} \hbar \sqrt{k/m} & \beta = \frac{\Delta I_c}{\Delta I_B} & \phi_e = \frac{\Delta E}{\Delta t} \frac{w_1}{x} + \frac{w_2}{x'} = \dots \\
 & \iint \vec{J} d\vec{S} & \vec{S} = \frac{1}{\mu_0} (\vec{E} \times \vec{B}) & \Delta I_B & \phi = \frac{2\pi \sin^2 \theta}{\lambda} \\
 & N_A = \sqrt{\frac{3R_m T}{M_r \cdot 10^{-3}}} & E = \hbar k^2 & 1 \text{ pc} = \frac{1 \text{ AU}}{r} & S = \frac{U}{I} & F_v = \dots \\
 & h = Sh\rho g & f_0 = \frac{1}{2\pi \sqrt{LC}} & \sigma = \frac{Q}{M} & M = F d \cos \alpha & \\
 & \cos \vartheta_2 & \int \vec{E} d\vec{l} = - \iint \frac{\partial \vec{B}}{\partial t} \cdot d\vec{S} & \rho = \frac{E}{c} = \frac{\hbar f}{c} = \frac{\hbar}{\lambda} & \omega = U_m \sin \omega(t - T) = U_m \sin 2T \\
 & R = R_0 \sqrt[3]{A} & \oint \vec{H} d\vec{l} = \iint (\vec{J} + \frac{\partial \vec{D}}{\partial t}) \cdot d\vec{S} & \varphi = mc\Delta t & F_g = \dots \\
 & L = 10 \log \frac{I}{I_0} & \Delta\psi = \frac{2\pi \Delta x}{\lambda} = \frac{2\pi d \sin \vartheta}{\lambda} & P = UI & h = \frac{1}{2} g t^2 & v - v_1(1 + \beta) \\
 & = \mu_0 \sum I_i & \rho = \frac{\vec{F}}{\Delta S} = \frac{m \Delta \vec{v}}{\Delta S \Delta t} & \nabla \times \left( -\frac{\partial \vec{B}}{\partial t} \right) = -\frac{\partial}{\partial t} (\operatorname{rot} \vec{B}) = -\mu_0 \frac{\partial}{\partial t} \left( \frac{\partial \vec{B}}{\partial t} \right) = \dots
 \end{aligned}$$

# Conclusion



# Récapitulatif



# Application à l'UE

## Définition du problème

- Dirichlet et Neuman,
- Problème aux valeurs propres.

## Discretisation du domaine :

- Maillage (non) uniforme.

## Discretisation des équations :

- Points internes,
- Points sur les bords,
- Choix du schéma.

## Écriture matricielle :

- Passer d'un système linéaire à une matrice,
- Définir les matrices et vecteurs nécessaires.

## Programmation :

- Programmation Python de chaque étape,
- Bibliothèques Python utiles.

## Validation :

- Trouver une solution analytique,
- Calculer la différence avec la solution numérique

# Plan de l'UE

## Idée générale :

Au premier semestre, on va introduire les notions de base, et s'intéresser en détails à une méthode numérique précise

Tous les jeudi matin (8h45-12h45) au bâtiment 625

21 Novembre : Cours 1 + Cours 2

4 Décembre : Cours 3 + TP 1

5 Décembre : Cours 4 + **TP 2**

12 Décembre : Cours 5 + TP 3

19 Décembre : Cours 6 + TP 4

9 Janvier : Cours 7 + TP 4

16 Janvier : Cours 8 + TP 5

23 Janvier : TP 5

30 Janvier : **Examen**

## Modalités d'évaluation :

TPs + examen oral (question de cours + exercice)