

# Résolution de l'équation $\text{rot rot } A^*$ par éléments finis de Nédélec

Ton Nom

## Résumé

Cet article présente la résolution numérique de l'équation  $\text{rot}(\mu^{-1} \text{rot } A) + \sigma \frac{\partial A}{\partial t} = J_s$  par la méthode des éléments finis de Nédélec. Nous détaillons la formulation variationnelle, les propriétés mathématiques du problème, la construction des éléments finis d'arête, et les aspects numériques de l'assemblage et de la résolution du système.

## 1 Introduction

Les équations de Maxwell jouent un rôle fondamental en électromagnétisme. Dans le cas de problèmes dépendants du temps avec des matériaux conducteurs, l'équation du potentiel vecteur magnétique  $A$  permet de modéliser les phénomènes d'induction.

## 2 Problème continu et formulation variationnelle

### 2.1 Équation aux dérivées partielles

On considère le potentiel vecteur  $A : \Omega \rightarrow \mathbb{R}^3$  solution du problème :

$$\text{rot}(\mu^{-1} \text{rot } A) + \sigma \frac{\partial A}{\partial t} = J_s \quad \text{dans } \Omega \quad (1)$$

$$A \times n = 0 \quad \text{sur } \partial\Omega \quad (2)$$

où  $n$  est la normale unitaire sortante à  $\partial\Omega$ ,  $\sigma$  est la conductivité,  $\mu$  la perméabilité, et  $J_s$  une densité de courant source.

### 2.2 Espaces fonctionnels

**Définition 2.1** (Espaces de Sobolev de champs vectoriels). On note

$$L^2(\Omega) = \{u : \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} |u|^2 < \infty\},$$

$$H(\text{curl}; \Omega) = \{v \in (L^2(\Omega))^3 \mid \text{curl } v \in (L^2(\Omega))^3\},$$

$$H_0(\text{curl}; \Omega) = \{v \in H(\text{curl}; \Omega) \mid v \times n = 0 \text{ sur } \partial\Omega\},$$

où  $n$  est la normale unitaire à  $\partial\Omega$ .

### 2.3 Formulation variationnelle

**Lemme 2.2** (Formule de Green pour le rotationnel). Soit  $\Omega$  un domaine borné de  $\mathbb{R}^3$  à frontière lipschitzienne. Pour tout  $u \in H(\text{curl}; \Omega)$  et  $v \in H^1(\Omega)^3$ , on a :

$$\int_{\Omega} (\text{curl } u) \cdot v d\Omega = \int_{\Omega} u \cdot (\text{curl } v) d\Omega + \int_{\partial\Omega} (u \times n) \cdot v dS$$

où  $n$  est la normale unitaire sortante à  $\partial\Omega$ .

À partir de l'équation forte :

$$\text{rot}(\mu^{-1} \text{rot } A) + \sigma \frac{\partial A}{\partial t} = J_s \quad \text{dans } \Omega$$

avec la condition aux limites  $A \times n = 0$  sur  $\partial\Omega$ , on multiplie par une fonction test  $\mathbf{w} \in H_0(\text{curl}; \Omega)$  et on intègre sur  $\Omega$  :

$$\int_{\Omega} \left[ \text{rot}(\mu^{-1} \text{rot } A) + \sigma \frac{\partial A}{\partial t} \right] \cdot \mathbf{w} d\Omega = \int_{\Omega} J_s \cdot \mathbf{w} d\Omega$$

En appliquant la formule de Green au terme contenant le rotationnel et l'identité vectorielle :

$$(u \times v) \cdot n = (n \times u) \cdot v = -(u \times n) \cdot v, \quad (3)$$

on obtient :

$$\int_{\Omega} \sigma \mathbf{w} \cdot \frac{\partial A}{\partial t} d\Omega + \int_{\Omega} \mu^{-1} (\text{curl } \mathbf{w}) \cdot (\text{curl } A) d\Omega = \int_{\Omega} \mathbf{w} \cdot J_s d\Omega. \quad (4)$$

La condition aux limites  $A \times n = 0$  sur  $\partial\Omega$  et le choix  $\mathbf{w} \in H_0(\text{curl}; \Omega)$  font que le terme de bord s'annule.

### Problème variationnel final

On cherche  $A(t) \in H_0(\text{curl}; \Omega)$  telle que pour tout  $\mathbf{w} \in H_0(\text{curl}; \Omega)$  :

$$a(A, \mathbf{w}) = L(\mathbf{w}),$$

où

$$a(A, \mathbf{w}) = \int_{\Omega} \sigma \mathbf{w} \cdot \frac{\partial A}{\partial t} d\Omega + \int_{\Omega} \mu^{-1} (\text{curl } A) \cdot (\text{curl } \mathbf{w}) d\Omega,$$

$$L(\mathbf{w}) = \int_{\Omega} \mathbf{w} \cdot J_s d\Omega.$$

## 3 Existence et unicité

**Théorème 3.1** (Lax–Milgram). Soit  $a(\cdot, \cdot)$  une forme bilinéaire continue et coercive sur un espace de Hilbert  $V$ , et  $L$  une forme linéaire continue sur  $V$ . Alors il existe une unique solution  $u \in V$  telle que  $a(u, v) = L(v)$  pour tout  $v \in V$ .

a justifier plus tard

## 4 Problème discret

Le principe des éléments finis est d'approximer  $H_0(\text{curl}; \Omega)$  par un sous-espace vectoriel de dimension finie  $V_h \subset H_0(\text{curl}; \Omega)$  construit sur un maillage  $\mathcal{T}_h$  de  $\Omega$ .

### 4.1 Formulation semi-discrète en espace

Le problème faible continu se discrétise ainsi : trouver  $A_h(t) \in V_h$  tel que, pour tout  $w_h \in V_h$  :

$$a(A_h(t), \mathbf{w}_h) = L(\mathbf{w})$$

En développant  $A_h$  sur une base de  $V_h$  :

$$A_h(x, t) = \sum_{j=1}^N a_j(t) \phi_j(x)$$

on obtient le système d'EDO suivant :

$$M \frac{d\mathbf{a}}{dt} + K \mathbf{a} = \mathbf{f}$$

où :

$$M_{ij} = \int_{\Omega} \sigma \phi_i \cdot \phi_j d\Omega$$

$$K_{ij} = \int_{\Omega} \mu^{-1} (\text{curl} \phi_i) \cdot (\text{curl} \phi_j) d\Omega$$

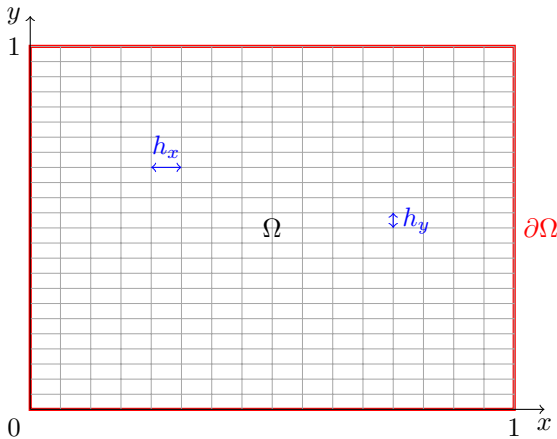
$$f_i = \int_{\Omega} \phi_i \cdot \mathbf{J}_s d\Omega$$

$$\mathbf{a}(t) = [a_1(t), \dots, a_N(t)]^T$$

$M$  est appelée **matrice de masse**,  $K$  la **matrice de rigidité**,  $\mathbf{f}$  le **second membre**, et  $\mathbf{a}$  le **vecteur des degrés de liberté**.

### 4.2 Définition du maillage

On considère le domaine carré  $\Omega = [0, 1]^2$  avec un maillage rectangulaire uniforme de (module élément finis fonctionne même pour des parallélogramme non uniforme mais le main pas du tout numérotation et génération de maillage car la transformation géométrique reste affine par rapport à l'élément de référence).



### 4.3 Choix de l'espace d'approximation $V_h$

#### 4.3.1 Exigences de conformité et limitations des éléments nodaux

Pour assurer la convergence de la méthode (pas toujours), l'espace  $V_h$  doit être *conforme* dans  $H(\text{curl}; \Omega)$ , c'est-à-dire :

$$V_h \subset H_0(\text{curl}; \Omega)$$

Une approche consisterait à utiliser des éléments finis nodaux de type Lagrange ( $P_1$  ou  $Q_1$ ) composante par composante. Cependant, ces éléments imposent une *continuité de toutes les composantes* du champ aux interfaces, ce qui implique :

$$V_h \subset H^1(\Omega)^3$$

Or, la condition de conformité dans  $H(\text{curl})$  ne requiert que la *continuité de la composante tangentielle* aux interfaces (trouver meilleure justification).

#### 4.3.2 Éléments de Nédélec d'ordre minimal

Les éléments de Nédélec (ou *edge elements*) sont spécialement conçus pour respecter la conformité  $H(\text{curl})$ . Leur construction repose sur une définition astucieuse des degrés de liberté pour n'imposer que la continuité tangentielle.

**Définition 4.1** (Élément de Nédélec de première famille d'ordre minimal). Soit  $K$  un rectangle quelconque. L'espace local d'approximation d'ordre minimal est défini par :

$$\mathcal{N}_0^{(1)}(K) = \mathbf{P}_0(K) \oplus \mathbf{x} \times \tilde{\mathbf{P}}_0(K)$$

où :

- $\mathbf{P}_0(K) = [P_0(K)]^3$  : espace des champs vectoriels constants sur  $K$
- $\tilde{\mathbf{P}}_0(K)$  : espace des polynômes homogènes de degré exactement 0 (constantes)
- $\mathbf{x} \times \tilde{\mathbf{P}}_0(K)$  : espace des champs de la forme  $\mathbf{x} \times \mathbf{c}$  où  $\mathbf{c}$  est constant

Les degrés de liberté associés sont les circulations sur les arêtes :

$$a_e(t) = \int_e \mathbf{A}(\mathbf{t}) \cdot \boldsymbol{\tau}_e ds \quad (1 \text{ par arête, } 4 \text{ au total})$$

où  $\boldsymbol{\tau}_e$  est le vecteur tangent unitaire à l'arête  $e$ .

#### 4.3.3 Justification de la construction de l'espace local

L'espace  $\mathcal{N}_0^{(1)}(K)$  présente une structure mathématique particulière qui mérite une explication détaillée. Pour un tétraèdre, le nombre d'arêtes est exactement 6, ce qui correspond à la dimension de cet espace. En effet :

- $\dim(\mathbf{P}_0(K)) = 3$  (champs constants)
- $\dim(\mathbf{x} \times \tilde{\mathbf{P}}_0(K)) = 3$  (produit vectoriel avec l'espace des constantes)
- La somme étant directe,  $\dim(\mathcal{N}_0^{(1)}(K)) = 6$

La forme spécifique  $\mathbf{x} \times \tilde{\mathbf{P}}_0(K)$  est cruciale pour préserver les propriétés de transformation sous l'application de la transformation de Piola covariante. Cette transformation

garantit que la continuité tangentielle est préservée lors du mapping de l'élément de référence à l'élément physique, assurant ainsi la conformité dans  $H(\text{curl})$ .

L'espace  $\mathcal{N}_0^{(1)}(K)$  capture les deux types de comportement fondamentaux pour les champs électromagnétiques :

1. Les champs constants ( $\mathbf{P}_0(K)$ )
2. Les champs de type rotationnel constant ( $\mathbf{x} \times \tilde{\mathbf{P}}_0(K)$ )

Cette décomposition correspond physiquement à la séparation entre les modes irrotationnels et les modes à divergence nulle dans la discrétisation.

Les 6 degrés de liberté (intégrales de la composante tangentielle sur les arêtes) forment une base duale de  $\mathcal{N}_0^{(1)}(K)$ . L'unisolvance est garantie par le fait que :

$$\int_e (\mathbf{a} + \mathbf{b} \times \mathbf{x}) \cdot \boldsymbol{\tau}_e ds = 0 \quad \forall \text{ arêtes } e \Rightarrow \mathbf{a} = \mathbf{b} = \mathbf{0}$$

Si l'on choisissait un autre espace de dimension 6, par exemple un sous-espace de  $[P_1(K)]^3$ , on perdrait plusieurs propriétés essentielles :

- **Non-conformité** : La transformation géométrique ne préserverait pas la continuité tangentielle
- **Modes parasites** : Apparition de solutions numériques non physiques
- **Instabilité** : Possible perte de la condition inf-sup dans les formulations mixtes
- **Manque d'invariance** : Les propriétés du rotationnel ne seraient pas préservées

L'espace de Nédélec d'ordre minimal représente donc le choix optimal pour la discrétisation des problèmes dans  $H(\text{curl})$  car il combine :

- La conformité avec l'espace fonctionnel continu
- La préservation des propriétés physiques essentielles
- La stabilité numérique
- L'invariance géométrique sous transformation de Piola

## 5 Assemblage et solveurs

### 5.1 Matrices élémentaires

L'assemblage du système global procède par sommation des contributions élémentaires sur chaque élément  $K$  du maillage  $\mathcal{T}_h$ . Pour chaque élément  $K$ , on calcule les matrices élémentaires :

$$K_{ij}^K = \int_K \mu^{-1} (\text{curl } \phi_i^K) \cdot (\text{curl } \phi_j^K) dK,$$

$$M_{ij}^K = \int_K \sigma \phi_i^K \cdot \phi_j^K dK,$$

$$f_i^K = \int_K \phi_i^K \cdot \mathbf{J}_s dK$$

Les intégrales sur le domaine global  $\Omega$  se décomposent en sommes d'intégrales sur les éléments :

$$M_{ij} = \sum_{K \in \mathcal{T}_h} M_{ij}^K, \quad K_{ij} = \sum_{K \in \mathcal{T}_h} K_{ij}^K, \quad f_i = \sum_{K \in \mathcal{T}_h} f_i^K$$

Pour un élément rectangulaire  $K$  en 2D, on utilise la transformation géométrique affine :

$$\mathbf{F}_K : \hat{K} \rightarrow K, \quad (x, y) = \mathbf{F}_K(\xi, \eta) = (x_0 + h_x \xi, y_0 + h_y \eta)$$

où  $\hat{K} = [-1, 1]^2$  est l'élément de référence, et  $h_x, h_y$  sont les dimensions de l'élément.

Le Jacobien de la transformation est :

$$J_K = \begin{pmatrix} h_x & 0 \\ 0 & h_y \end{pmatrix}, \quad \det(J_K) = h_x h_y$$

**Théorème 5.1** (Transformation de Piola). *Soit  $K$  un élément physique et  $\hat{K}$  l'élément de référence. Soit  $\mathbf{F}_K : \hat{K} \rightarrow K$  une transformation géométrique affine de Jacobienne  $J_K$ .*

*Les fonctions de base  $\phi_i^K$  sur l'élément physique  $K$  sont reliées aux fonctions de base  $\hat{\phi}_i$  sur l'élément de référence  $\hat{K}$  par la transformation de Piola covariante :*

$$\phi_i^K(x, y) = J_K^{-T} \hat{\phi}_i(\xi, \eta)$$

*Le rotationnel se transforme selon la relation :*

$$\text{curl}_{(x,y)} \phi_i^K = \frac{1}{\det(J_K)} \text{curl}_{(\xi,\eta)} \hat{\phi}_i$$

En effectuant le changement de variables

$$dK = \det(J_K) d\hat{K} = h_x h_y d\xi d\eta,$$

on exprime les intégrales définissant les matrices élémentaires sur l'élément physique comme des intégrales prises sur l'élément de référence.

**Matrice de masse élémentaire :**

$$M_{ij}^K = h_x h_y \int_{-1}^1 \int_{-1}^1 \sigma(\xi, \eta) \left( J_K^{-T} \hat{\phi}_i(\xi, \eta) \right) \cdot \left( J_K^{-T} \hat{\phi}_j(\xi, \eta) \right) d\xi d\eta$$

**Matrice de rigidité élémentaire :**

$$K_{ij}^K = \frac{1}{h_x h_y} \int_{-1}^1 \int_{-1}^1 \mu^{-1}(\xi, \eta) (\text{curl } \hat{\phi}_i(\xi, \eta)) \cdot (\text{curl } \hat{\phi}_j(\xi, \eta)) d\xi d\eta$$

**Vecteur second membre élémentaire :**

$$f_i^K = h_x h_y \int_{-1}^1 \int_{-1}^1 \left( J_K^{-T} \hat{\phi}_i(\xi, \eta) \right) \cdot \mathbf{J}_s(\xi, \eta) d\xi d\eta$$

Dans notre cas, la jacobienne est constante et donc on peut effectuer le calculer sur l'élément de référence et simplement effectuer la multiplication par  $a$  droite et  $a$  gauche

### 5.2 Quadrature de Gauss en 2D

Pour calculer les intégrales élémentaires, on utilise une quadrature de Gauss sur le carré unité  $[-1, 1]^2$  :

$$\int_{-1}^1 \int_{-1}^1 g(\xi, \eta) d\xi d\eta \approx \sum_{p=1}^{N_g} \sum_{q=1}^{N_g} w_p w_q g(\xi_p, \eta_q)$$

Pour une précision d'ordre 2 (quadrature de Gauss  $2 \times 2$ ), on utilise  $N_g = 2$  points par direction avec :

$$\xi_p, \eta_q \in \left\{ -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right\}, \quad w_p = w_q = 1$$

Ainsi, l'intégrale sur l'élément de référence est approximée en évaluant la fonction  $g$  en ces quatre points et en sommant les contributions pondérées.

### 5.3 Assemblage global

Les matrices globales  $M$ ,  $K$  et le vecteur global  $\mathbf{f}$  sont construits par sommation des contributions élémentaires :

$$M = \sum_{K \in \mathcal{T}_h} M^K, \quad K = \sum_{K \in \mathcal{T}_h} K^K, \quad \mathbf{f} = \sum_{K \in \mathcal{T}_h} \mathbf{f}^K$$

L'assemblage nécessite une table de connectivité qui associe à chaque élément  $K$  les indices globaux de ses degrés de liberté.

**Orientation des arêtes** L'orientation cohérente des arêtes du maillage est cruciale pour garantir l'unicité des fonctions de base et la continuité de la composante tangentielle.

### 5.4 Discrétisation temporelle

Le système d'EDO peut être discrétisé en temps par des schémas implicites (Euler implicite, Crank-Nicolson) :

$$M \frac{\mathbf{a}^{n+1} - \mathbf{a}^n}{\Delta t} + K \mathbf{a}^{n+1} = \mathbf{f}^{n+1}$$

soit

$$\left( \frac{1}{\Delta t} M + K \right) \mathbf{a}^{n+1} = \frac{1}{\Delta t} M \mathbf{a}^n + \mathbf{f}^{n+1}$$

### 5.5 Résolution du système linéaire

La discrétisation nous a la résolution d'un système linéaire à chaque itération :

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

avec  $\mathbf{A} \in \mathbb{R}^{n \times n}$  symétrique définie positive,  $\mathbf{x} \in \mathbb{R}^n$  inconnu,  $\mathbf{b} \in \mathbb{R}^n$  second membre.

#### 5.5.1 Méthode du Gradient Conjugué (CG)

La méthode du Gradient Conjugué (CG) est un algorithme itératif efficace pour résoudre les systèmes linéaires où la matrice  $\mathbf{A}$  est symétrique définie positive. Elle minimise itérativement la fonctionnelle quadratique :

$$\phi(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} - \mathbf{x}^\top \mathbf{b}$$

À chaque itération  $k$ , la méthode génère une direction de recherche  $\mathbf{p}_k$  qui est **A-conjuguée** aux directions précédentes, garantissant ainsi la préservation des minimisations antérieures :

$$\mathbf{p}_i^\top \mathbf{A} \mathbf{p}_j = 0 \quad \text{pour } i \neq j$$

Cette propriété d'A-conjugaison assure une convergence en au plus  $n$  itérations dans un espace de dimension  $n$ .

#### 5.5.2 Préconditionnement

La vitesse de convergence de l'algorithme du GC dépend fortement du conditionnement de la matrice  $\mathbf{A}$  :

$$\kappa(\mathbf{A}) = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}$$

Plus la matrice est mal conditionnée ( $\kappa(\mathbf{A}) \gg 1$ ), plus l'algorithme nécessitera d'itérations. Une estimation théorique du nombre d'itérations requis pour une précision  $\epsilon$  est donnée par :

$$k \approx \frac{1}{2} \sqrt{\kappa(\mathbf{A})} \ln \left( \frac{2}{\epsilon} \right)$$

Pour accélérer la convergence, on introduit un préconditionneur  $\mathbf{M}$  qui améliore le conditionnement tout étant facilement inversible. Le système original est transformé en un système équivalent mieux conditionné :

$$\mathbf{M}^{-1} \mathbf{A} \mathbf{x} = \mathbf{M}^{-1} \mathbf{b}$$

L'objectif est d'obtenir un conditionnement réduit :

$$\kappa(\mathbf{M}^{-1} \mathbf{A}) \ll \kappa(\mathbf{A})$$

Ce qui se traduit concrètement par une réduction significative du nombre d'itérations nécessaires.

#### 5.5.3 Préconditionneur de Jacobi

Parmi les préconditionneurs les plus simples, le préconditionneur de Jacobi (ou préconditionneur diagonal) utilise uniquement les éléments diagonaux de la matrice :

$$\mathbf{M} = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$$

Son inversion est triviale et peu coûteuse en calcul :

$$\mathbf{M}^{-1} = \text{diag} \left( \frac{1}{a_{11}}, \frac{1}{a_{22}}, \dots, \frac{1}{a_{nn}} \right)$$

## 6 Implémentation et algorithme

Le code est composé de trois modules : `mod_fem`, `mod_functions`, `mod_var`, ainsi que d'un programme principal `main_v2.f90`. Dans cette section, nous détaillons la structure du programme principal.

### 6.1 Domaine et Discrétisation

Le domaine de calcul est un carré unitaire défini par

$$[0, 1] \times [0, 1].$$

La discrétisation spatiale utilise une grille structurée régulière avec les paramètres suivants :

— Nombre d'éléments :

$$n_x \times n_y \quad \text{éléments quadrilatéraux}$$

— Nombre de nœuds :

$$N_{\text{nœuds}} = (n_x + 1) \times (n_y + 1)$$

— Pas d'espace :

$$\Delta x = \frac{1}{n_x}, \quad \Delta y = \frac{1}{n_y}$$

Le tableau `nodes`, de dimensions

$$N_{\text{nodes}} \times 2,$$

contient les coordonnées de tous les nœuds du maillage. Les coordonnées du nœud d'indice global  $i$  sont données par :

$$x_i = (i_x - 1) \Delta x, \quad y_i = (i_y - 1) \Delta y,$$

où

$$i_x = 1, \dots, n_x + 1, \quad i_y = 1, \dots, n_y + 1.$$

L'indice global  $i$  est calculé comme suit :

$$i = (i_y - 1)(n_x + 1) + i_x.$$

Les nœuds sont organisés par ligne, de bas en haut.  
Organisation :

$$[x_1, y_1], [x_2, y_2], \dots, [x_{N_{\text{nodes}}}, y_{N_{\text{nodes}}}]$$

Le tableau `quad_connectivity`, de dimensions

$$(n_x \times n_y) \times 4,$$

stocke la connectivité des éléments.

Chaque ligne correspond à un élément quadrilatéral et contient les indices de ses quatre nœuds, organisés comme suit :

1. Nœud inférieur gauche
2. Nœud inférieur droit
3. Nœud supérieur droit
4. Nœud supérieur gauche

## 6.2 Assemblage

On sait que l'assemblage des matrices globale de masse  $M$  et de raideur  $K$  s'écrit sous la forme :

$$M_{i,j} = \sum_{K \in \mathcal{T}_h} M_{i,j}^K, \quad K_{i,j} = \sum_{K \in \mathcal{T}_h} K_{i,j}^K,$$

où  $M^K$  et  $K^K$  désignent les matrices élémentaires associées à l'élément  $K$ . Ces matrices élémentaires sont définies par :

$$M_{ij}^K = h_x h_y \int_{-1}^1 \int_{-1}^1 \sigma(\xi, \eta) \left( J_K^{-T} \widehat{\phi}_i(\xi, \eta) \right) \cdot \left( J_K^{-T} \widehat{\phi}_j(\xi, \eta) \right) d\xi d\eta,$$

$$K_{ij}^K = \frac{1}{h_x h_y} \int_{-1}^1 \int_{-1}^1 \mu^{-1}(\xi, \eta) \left( \text{curl } \widehat{\phi}_i(\xi, \eta) \right) \cdot \left( \text{curl } \widehat{\phi}_j(\xi, \eta) \right) d\xi d\eta.$$

Ces contributions élémentaires sont calculées par la subroutine du module `mod.fem` :

```
process_nedelec_element(elem_nodes, elem_edges,
                        elem_coords, Me, Ke, fe, ierr)
```

Cette routine prend en entrée :

- la numérotation globale des nœuds de l'élément (via le tableau de connectivité),
- les coordonnées de ces nœuds (via le tableau `nodes`).

Elle retourne les matrices élémentaires  $M^K$ ,  $K^K$  calculés à partir des formules ci-dessus.

L'assemblage global exploite le caractère creux (sparse) des matrices. On utilise deux structures dynamiques `M_triplet` et `K_triplet`, chacune contenant trois tableaux parallèles `row`, `col`, `val`.

L'assemblage se déroule ainsi :

1. Calcul des contributions locales (`Me`, `Ke`, `fe`) via la subroutine `process_nedelec_element`.
2. Pour chaque paire locale  $(a, b) \in \{1, \dots, 4\}^2$ , insertion de  $Ke(a, b)$  et  $Me(a, b)$  dans les triplets aux indices globaux correspondants à l'aide de la subroutine `triplet_add_entry`.

La routine `triplet_add_entry` n'insère pas les valeurs numériquement nulles (si  $|\text{val}| < 10^{-14}$ ). Cela permet de manipuler des tableaux beaucoup plus petits que si l'on stockait toutes les valeurs, offrant ainsi un gain mémoire et, potentiellement, un gain en temps lors des opérations d'algèbre linéaire et de la résolution du système.

À ce stade, aucune sommation n'est effectuée, donc le tableau résultant doit contenir des doublons : plusieurs entrées avec le même couple  $(i, j)$  mais des valeurs différentes.

row	col	val
1	1	5.0
1	3	2.0
2	1	1.5
2	2	4.0
3	2	-0.5
3	3	3.0

TABLE 1 – Exemple de structure dynamique `triplet_matrix`

Ici, chaque entrée  $(\text{row}(k), \text{col}(k), \text{val}(k))$  correspond à une contribution locale d'un élément. Les indices peuvent apparaître plusieurs fois si plusieurs éléments contribuent à la même case de la matrice globale.

Ensuite, les deux structures `M_triplet` et `K_triplet` sont converties au format **CSR (Compressed Sparse Row)**. Ce format permet d'effectuer des opérations d'algèbre linéaire (produits matrice-vecteur, résolutions de systèmes, etc.) avec un coût mémoire et temporel minimal. Lors de cette conversion, les contributions locales issues de chaque élément sont automatiquement *sommées* si elles correspondent au même couple d'indices  $(i, j)$ .

**Exemple illustratif :** Supposons qu'après assemblage en COO on ait le tableau suivant pour une matrice  $3 \times 3$  :

row	col	val
1	1	5.0
1	3	2.0
2	1	1.5
2	2	4.0
3	2	-0.5
3	3	3.0
1	1	1.0

Lors de la conversion COO  $\rightarrow$  CSR, les doublons sur (1, 1) sont sommés et le résultat CSR est :

$$\text{values} = [6.0, 2.0, 1.5, 4.0, -0.5, 3.0]$$

$$\text{col\_indices} = [1, 3, 1, 2, 2, 3]$$

$$\text{row\_ptr} = [1, 3, 5, 7]$$

Ici, `row_ptr(i)` indique la position dans `values` où commence la ligne  $i$ .

Après l'assemblage sous format COO et la conversion vers CSR, les boundary nodes sont identifiés pour faciliter l'application de la condition au bord de type dirichlet.

Concernant la condition au bords comme les degré de liberté sont les circulations du potentiel vecteurs sur les arrêtes il suffit d'annuler les dof correspondant aux arretes frontaliers. Ceci sera effectuer a l'intérieur de la time loop.

## 7 Expériences numériques

Pour valider le code, nous utilisons la méthode des solutions manufacturées (MSM). Le principe est de choisir une solution analytique arbitraire, ici le potentiel vecteur  $\mathbf{A}$ , qui respecte les conditions aux limites du problème. On injecte ensuite cette solution dans l'équation pour calculer le terme source  $\mathbf{J}_s$  nécessaire pour que l'équation soit exactement satisfaite. Sans oublier d'initialiser le potentiel vector pour la valeur de la solution manufacturé en  $t=0$ .

### 7.1 Solutions manufacturées pour le potentiel vecteur

Les composantes du potentiel vecteur sont choisies comme suit :

$$A_x(x, y, t) = \sin(\pi y) \cdot \cos(\pi x) \cdot e^{-t}$$

$$A_y(x, y, t) = -\cos(\pi y) \cdot \sin(\pi x) \cdot e^{-t}$$

Ces fonctions respectent les conditions aux limites du problème.

En substituant les solutions manufacturées dans l'équation originale, on obtient l'expression du terme source  $\mathbf{J}_s$  qui doit être ajouté pour que les solutions soient exactes. Les composantes du courant source sont données par :

$$J_{s_x}(x, y, t) = \left( \frac{2\pi^2}{\mu(x, y)} - \sigma(x, y) \right) \cdot A_x(x, y, t)$$

$$J_{s_y}(x, y, t) = \left( \frac{2\pi^2}{\mu(x, y)} - \sigma(x, y) \right) \cdot A_y(x, y, t)$$

On peut alors comparer la solution numérique calculée par le code avec cette solution exacte pour étudier la convergence et la précision de la méthode numérique.

## 8 Conclusions et perspectives

### Références