TP2 : Equation intégrale pour l'équation de Helmholtz 2D

Clément Baillet

Ce TP est la conclusion d'une trilogie dont l'objectif était de mettre en place notre premier solveur BEM (Boundary Element Method), et de l'appliquer à la résolution de l'équation de Helmholtz en 2D. On rappelle le contexte : on considère le cas de la diffraction d'une onde incidente plane, i.e. $u^{\text{inc}} = \mathrm{e}^{-\mathbf{k}\cdot\mathbf{x}}$, par un disque de rayon a centré en 0 et de frontière Γ. $\mathbf{k}$ est le vecteur qui permet de déterminer l'angle d'incidence de l'onde. Le nombre d'onde du problème est donné par $k = |\mathbf{k}|$. Le domaine extérieur est noté Ω^+. On approximera Γ par un maillage à N noeuds et segments $(e_i)_i$. Durant ce TP, notre but va être de programmer de manière effective le solveur BEM en passant par une formulation variationnelle du problème, et en nous aidant des résultats obtenus dans les TPs précédents pour vérifier nos résultats. Ce TP se divise en 3 étapes:

- Déterminer la formulation variationnelle de notre problème.
- Résoudre numériquement ce problème variationnel pour avoir la trace de p sur notre frontière.
- Application de la représentation intégrale que l'on a programmé dans le TP1 pour obtenir le champ dans tout le domaine.

On commence par mettre le problème sous forme variationnelle. On rappelle que le problème initial est :

$$\text{Trouver } p \in H^{1/2}(\Gamma) \text{ tel que } \int_\Gamma G(\mathbf{x},\mathbf{y})p(\mathbf{y})\mathrm{d}\Gamma(\mathbf{y}) = -u^{\text{inc}}.$$

Prenons donc une fonction test $\tilde{p} \in H^{1/2}(\Gamma)$, on multiplie l'équation par celle-ci, et on intègre pour obtenir :

$$\text{Trouver } p \in H^{1/2}(\Gamma) \text{ tel que pour tout } \tilde{p} \in H^{1/2}(\Gamma),$$
$$\int_\Gamma \int_\Gamma G(\mathbf{x},\mathbf{y})p(\mathbf{y})\tilde{p}(\mathbf{x})\mathrm{d}\Gamma(\mathbf{y})\mathrm{d}\Gamma(\mathbf{x}) = -\int_\Gamma u^{\text{inc}}(\mathbf{x})\tilde{p}(\mathbf{x})\mathrm{d}\Gamma(\mathbf{x}).$$

Maintenant qu'on a modifié la forme de notre problème, nous allons pouvoir le discrétiser sur notre maillage de Γ. Tout d'abord, notons que notre solution p est censé être valable pour tout $\tilde{p}$, ce qui signifie qu'en particulier on peut choisir les fonctions indicatrices

$$\mathbf{1}_{e_i}$$

où e_i est un segment du maillage du Γ. On approxime également p sur chaque segment e_i par sa valeur en le milieu, ce qui nous donne le problème suivant :

$$\text{Trouver } (p_{e_i})_i \in M_{1,N}(\mathbb{C}) \text{ tel que}$$
$$\sum_{j\in[\![0,N-1]\!]} p_{e_j} \int_{\Gamma_{e_i}} \int_{\Gamma_{e_j}} G(\mathbf{x},\mathbf{y})\mathrm{d}\Gamma_{e_j}(\mathbf{y})\mathrm{d}\Gamma_{e_i}(\mathbf{x}) = -\int_{\Gamma_{e_i}} u^{\text{inc}}(\mathbf{x})\mathrm{d}\Gamma_{e_i}(\mathbf{x}).$$

On note que résoudre ce problème revient à résoudre le système linéaire

$$\mathbb{A}\mathbf{p} = \mathbf{b}$$

où

$$\mathbb{A} = \left(\int_{\Gamma_{e_i}} \int_{\Gamma_{e_j}} G(\mathbf{x},\mathbf{y})\mathrm{d}\Gamma_{e_j}(\mathbf{y})\mathrm{d}\Gamma_{e_i}(\mathbf{x})\right)_{i,j} \text{ et } \mathbf{b} = \left(-\int_{\Gamma_{e_i}} u^{\text{inc}}(\mathbf{x})\mathrm{d}\Gamma_{e_i}(\mathbf{x})\right)_i.$$

On fait une première fonction `secondMembre` qui renvoie $\mathbf{b}$ en utilisant une approximation de Gauss-Legendre (comme pour le calcul de la représentation intégrale):

```
def secondMembre(N,segments,nodes,uInc,points,poids):
  b = np.zeros(N,dtype=complex)
  for e in segments:
      i = e[0]
      i2 = e[1]
      n1 = nodes[i]
      n2 = nodes[i2]
      b[i] = -integrale2D(lambda x,y:uInc(x,y,k),n1,n2,points,poids)
  return b
```

Le calcul de $\mathbb{A}$ est un peu plus délicat. Lorsque l'on est en dehors de la diagonale, il n'y a pas de soucis et on peut appliquer deux fois l'approximation de Gauss-Legendre. Sur la diagonale cependant, il va falloir gérer les singularités de la fonction de Green lorsque $e_i = e_j$. Pour ce faire, on va utiliser un développement asymptotique :

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{2\pi}\ln\frac{1}{\|\mathbf{x}-\mathbf{y}\|} + \frac{i}{4} - \frac{1}{2\pi}\left[\ln\left(\frac{k}{2}\right)+\gamma\right] + \mathcal{O}\left(\|\mathbf{x}-\mathbf{y}\|^2\ln\frac{1}{\|\mathbf{x}-\mathbf{y}\|}\right)$$

où γ est la constante d'Euler. Nous allons séparer le noyau en une partie singulière (la partie logarithmique) et une partie régulière (tout le reste). La partie régulière s'intègre sans soucis avec une quadrature de Gauss, et la partie singulière s'intègre de manière analytique (cf Figure 1) :

$$\int_{\Gamma_e} -\frac{1}{2\pi}\ln\|\mathbf{x}_h-\mathbf{y}_h\|\,\mathrm{d}\Gamma(\mathbf{y}_h) = -\frac{1}{2\pi}\left(d^e_{j+1}\cdot\boldsymbol{\tau}_e\,\ln\|d^e_{j+1}\| - d^e_j\cdot\boldsymbol{\tau}_e\,\ln\|d^e_j\| - |\Gamma_e| + d_e(\mathbf{x}_h)\,\Omega\right) \quad (1)$$

où l'on a utilisé les notations suivantes (cf. Figure 1) :

- $\boldsymbol{\tau}_e$ est le vecteur tangent unitaire,
- $d_e(\mathbf{x}_h)$ est la distance de $\mathbf{x}_h$ à Γ_e,
- Ω est l'angle solide sous lequel $\mathbf{x}_h$ voit Γ_e, $0 \le \Omega \le \pi$.

Comme $\Gamma_e = \Gamma_{e'}$, $\mathbf{x}_h$ appartient au deux segments, donc $d_e(\mathbf{x}_h) = 0$ de sorte que la formule (1) devient :

$$\int_{\Gamma_e} -\frac{1}{2\pi}\ln\|\mathbf{x}_h-\mathbf{y}_h\|\,\mathrm{d}\Gamma(\mathbf{y}_h) = -\frac{1}{2\pi}\left(d^e_{j+1}\cdot\boldsymbol{\tau}_e\,\ln\|d^e_{j+1}\| - d^e_j\cdot\boldsymbol{\tau}_e\,\ln\|d^e_j\| - |\Gamma_e|\right) \quad (2)$$

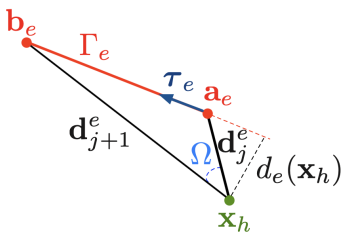

Figure 1: Notations pour l'intégration de la partie logarithmique

Dans notre cas, comme $\mathbf{x}_h$ est sur $\Gamma_{e_i} = \Gamma_{e_i}$, on a en particulier

$$d^e_{j+1}\cdot\tau_e = \|d^e_{j+1}\| \text{ et } d^e_j\cdot\tau_e = -\|d^e_j\|,$$

donc la formule (2) devient

$$\int_{\Gamma_e} -\frac{1}{2\pi} \ln \|\mathbf{x}_h - \mathbf{y}_h\| \, \mathrm{d}\Gamma(\mathbf{y}_h) = -\frac{1}{2\pi} \Big(\|d^e_{j+1}\| \ln \|d^e_{j+1}\| + \|d^e_j\| \ln \|d^e_j\| - |\Gamma_e| \Big). \tag{2}$$

Maintenant que le problème de la singularité est réglé pour la première intégration, la seconde peut se faire avec Gauss-Legendre comme précédemment (puisque $\lim_{x\to 0}(x \ln(x)) = 0$). Voici le code appliquant tout ce qu'on vient de décrire :

```
def matriceA(N,segments,nodes,G,points,poids):
  A = np.zeros((N,N),dtype=complex)
  for ex in segments:
      i = ex[0]
      for ey in segments:
          j = ey[0]
          if i==j:
              coeff = -1/(2*np.pi)

              d = lambda x,y,j: np.sqrt((nodes[j][0]-x)**2 +
              (nodes[j][1]-y)**2)
              n1,n2 = nodes[(j+1)mod N]

              longueur = d(n1,n2,j)

              f = lambda x,y: coeff*(d(x,y,(j+1)modN)*
              np.log(d(x,y,(j+1)mod N)) +
              d(x,y,j)*np.log(d(x,y,j)) - longueur)

              A[i][j] += integrale2D(f,nodes[i],
              nodes[(i+1)mod N],points,poids)
              const = 1j/4 + coeff*(np.log(k/2)+np.euler_gamma)

              g = lambda X,Y: integrale2D(lambda x,y: const,nodes[j],
              nodes[(j+1)mod N],points,poids)

              A[i][j] += integrale2D(g,nodes[i],nodes[(i+1)mod N],
              points,poids)
          else:
              I = lambda X,Y: integrale2D(lambda x,y: G((x,y),(X,Y),k),
              nodes[i],nodes[(i+1)mod N],points,poids)

              A[i][j] += integrale2D(I,nodes[j],nodes[(j+1)mod N],
              points,poids)
  return A
```

On peut maintenant résoudre le système afin de déterminer p. La manière la plus efficace pour le faire serait d'utiliser une méthode itérative, qui permettrait de manière optimale de trouver la solution en $\mathcal{O}(N^2)$, mais comme la matrice $\mathbb{A}$ est dense et sans propriété particulière, il est difficile de prouver la convergence d'une telle méthode dans notre cas. On va donc se servir de la fonction `numpy.linalg.solve`: les calculs se font en $\mathcal{O}(N^3)$. La Figure 2 montre la comparaison entre le vecteur $\mathbf{p}$ que l'on vient de calculer, et la fonction p calculée analytiquement.

On fournit un graphe montrant l'erreur relative maximale commise pour différentes valeurs de k, et pour différentes tailles de maillage (cf Figure 3). On remarque une décroissance de l'erreur lorsque N augmente, ce qui témoigne de la convergence de notre code. Pour ce qui est de sa complexité temporelle, en supposant qu'on utilise une méthode itérative pour résoudre le système. Le calcul de la matrice se fait en $\mathcal{O}(N^2)$ (deux boucles `for` de longueur N avec uniquement des opérations en temps constant effectuées dans chacune d'elle), et le calcul de $\mathbf{b}$ se fait en $\mathcal{O}(N)$. Comme la résolution de notre système se fait en $\mathcal{O}(N^3)$, alors la complexité de tout le calcul est

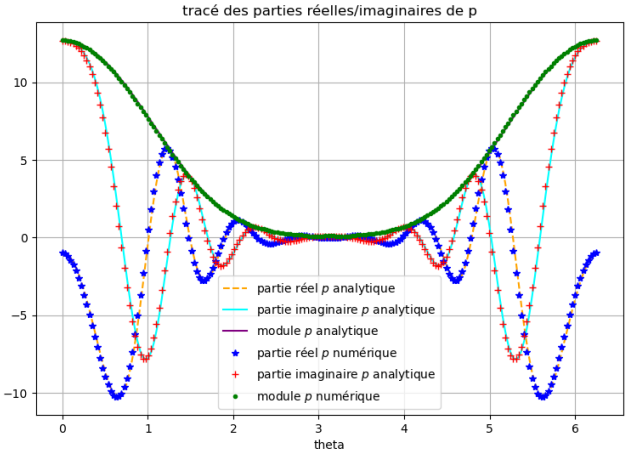

Figure 2: tracé des parties réelles et imaginaires de p pour $k = 2\pi$, $N = 200$, $a = 1.0$ et $nb_{\text{Gauss}} = 2$

finalement de l'ordre de

$$\mathcal{O}(N^3).$$

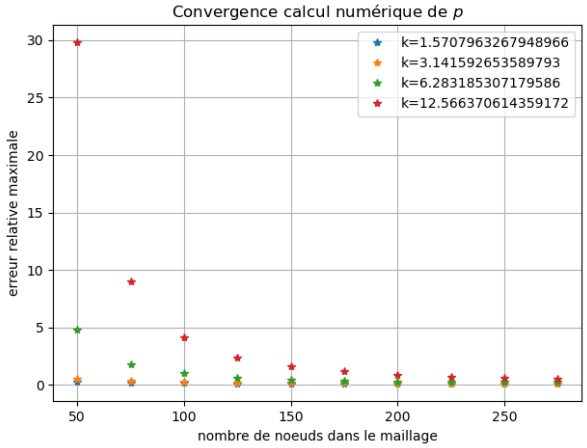

Figure 3: Erreur relative maximale pour p pour k allant de $\pi/2$ à 4π, et pour un nombre de noeuds variant de 50 à 200

Le calcul numérique de p permet désormais de réutiliser la fonction `representionIntegrale` de notre TP1. Voici alors la comparaison entre u^+ obtenu numériquement grâce au calcul de p, et la solution exacte (cf Figure 4):

On a également fait le test sur d'autres parties du domaine extérieur et le résultat était tout aussi concluant. On fournit le tracé des erreurs relatives afin de confirmer la convergence de notre solveur (cf Figure 5).

On rappelle que le complexité temporelle de la représentation intégrale du TP1 était de l'ordre de $\mathcal{O}(N \times N_{\text{xi}})$. Le temps de calcul dépendra principalement de la taille de la discrétisation du domaine extérieur:

$$\mathcal{O}(\max(N \times N_{\text{xi}} \ ; N^3)).$$

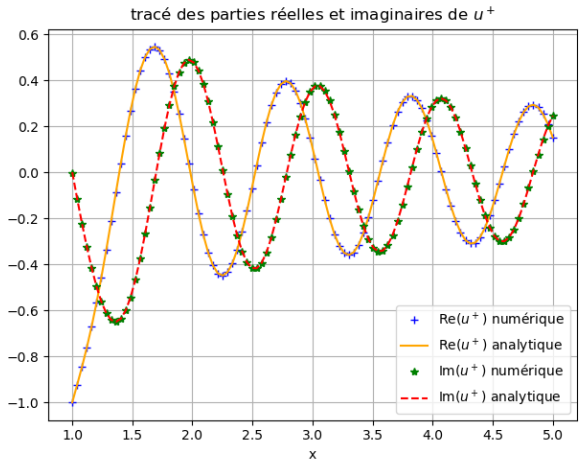

Figure 4: Tracé des parties réelles et imaginaires de u^+ pour $k = 2\pi$, $N = 200$, $a = 1.0$, $N_{\text{xi}} = 100$ et $nb_{\text{Gauss}} = 2$

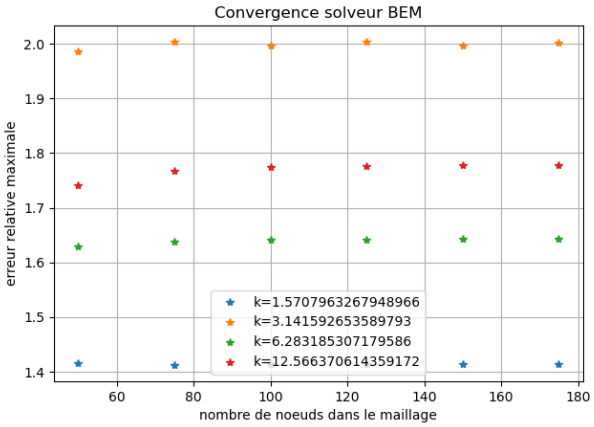

Figure 5: Erreur relative maximale de notre solveur BEM pour k allant de $\pi/2$ à 4π, et pour un nombre de noeuds variant de 50 à 200

Conclusion

Ce TP marque l'aboutissement de la mise en œuvre complète d'un solveur BEM pour l'équation de Helmholtz en deux dimensions. Après avoir établi la formulation variationnelle du problème, nous avons discrétisé la frontière puis implémenté la résolution numérique du système intégral associé. Une attention particulière a été portée au traitement de la singularité logarithmique du noyau de Green, dont la prise en compte correcte s'est révélée essentielle pour obtenir des résultats précis.

Les comparaisons avec les solutions analytiques montrent une très bonne concordance, et l'erreur relative maximale décroît de manière cohérente avec le raffinement du maillage, attestant de la convergence de la méthode. La complexité temporelle observée, de l'ordre de $\mathcal{O}(N^3)$, est conforme aux attentes pour une implémentation dense du BEM.

Ce travail nous a permis de consolider la compréhension des formulations intégrales et des aspects numériques liés à leur discrétisation. Il ouvre la voie à des extensions possibles, comme l'utilisation de méthodes itératives plus efficaces (de type GMRES ou FMM-BEM) ou l'étude de géométries plus complexes.