

Techniques de réduction des ordonnances modèles

Jeu de problèmes 2 | RB pour les problèmes elliptiques linéaires A ne

Christophe Prud'homme

1 Énoncé du problème | Conception d'une nageoire thermique

Nous considérons le problème de la conception d'un n thermique décrit dans le jeu de problèmes 1. Dans PS1 nous avons examiné quelques questions thoeoretical (formulation faible et formulation d'optimisation, convergence de l'approximation de base réduite) et avons dérivé les quantités de base réduites nécessaires, c.-à-d. expressions pour $AN()$, FN , et LN . Cet ensemble de problèmes est consacré à la mise en œuvre de l'approximation de base réduite et à la résolution d'un problème de conception simple.

1,1 Partie 1 { Approximation de base réduite

Le point de départ pour l'approximation de base réduite est une { haute discrétisation de vérité d'élément de nite dimensionnelle. Dans l'étape o ine, nous avons besoin de la solution d'élément de nite pour construire la base réduite et nous avons donc également besoin des matrices FE. Dans cet ensemble de problèmes, nous sautons l'étape d'assemblage du technicien de maintenance et fournissons toutes les données nécessaires à l'utilisation dans MATLAB (voir l'annexe 1).

Nous avons vu en classe que la solution de base réduite $u_N() \in \mathbb{R}^N$ satis l'ensemble de N équations linéaires,

$$AN()u_N() = FN; \quad (1)$$

et que les outputis donnés par

$$T_{rootN}() = L^T u_N() : \quad (2)$$

Nous avons dérivé des expressions pour $AN() \in \mathbb{R}^{N \times N}$ en termes de $AN()$ et Z , $FN \in \mathbb{R}^N$ en termes de FN et Z , et $LN \in \mathbb{R}^N$ en termes de LN et Z ; ici Z est une matrice $N \times N$ dont la colonne j th est $u_N(j)$ (les valeurs nodales de $u_N(j)$). Enfin, il résulte d'une dépendance de ne paramètre que $AN()$ peut être exprimé comme

$$AN() = \sum_{q=1}^Q AN_q; \quad (3)$$

L'objectif est de mettre en œuvre une version o ine / online de la méthode { base réduite suivant la décomposition computationnelle indiquée ci-dessous.

O ine

1. Choisissez N .
2. Choisir le numéro de série de l'échantillon.
3. Construction Z .
4. Construire $le^N QA$; $q = 1; : : ; Q$; FN ; et

LN : En ligne

1. Formule $AN()$ de (3).
2. Résoudre $AN()u_N() = FN$:
3. Évaluer la sortie $T_{rootN}()$ de (2).

1 L'idée est que l'étape offline ne se fait qu'une seule fois, générant un petit nombre de données avec l'AqN ; $q = 1; \dots; Q$, FN et LN; l'étape en ligne accède ensuite à ces données pour fournir une réponse en temps réel aux nouvelles requêtes. Pour les calculs de l'élément offline requis dans cette question et dans les questions suivantes, vous devez d'abord utiliser la triangulation grossière Th;grossière.

a) Montrez que le nombre d'opérations pour l'étape en ligne de votre code est indépendant de N. En particulier, montrer que le nombre d'opérations (nombre d'opérations de point d'avoine) pour la phase en ligne, pour chaque nouveau d'intérêt, peut être exprimé comme

$$c_1 N^1 + c_2 N^2 + c_3 N^3; \quad (4)$$

pour $c_1; c_2; c_3; 1; 2$; et 3 indépendants de N. Donner des valeurs pour les constantes $c_1; c_2; c_3; 1; 2$; et 3.

b) Nous considérons d'abord un problème d'un paramètre ($P = 1$). À cette fin, nous gardons le numéro Biot fixé à $Bi = 0.1$ et supposons que les conductivités de tous les ns sont équivalentes, c.-à-d., $k_1 = k_2 = k_3 = k_4$, mais sont autorisés à varier entre 0.1 et 10 nous avons donc $D = [0.1; 10]$: L'ensemble d'échantillons SN pour $N_{max} = 8$ est donné dans les données le RB_sample.sample1.

1. Générer la matrice de base réduite Z et toutes les quantités de base réduites nécessaires. Vous avez deux options : vous pouvez utiliser la solution "instantanés" directement dans Z ou effectuer une orthonormalisation Gram-Schmidt pour construire Z (Notez que vous avez besoin du produit interne X pour effectuer Gram-Schmidt; ici, nous utilisons $(;)X = a(;)$, où $\cup = 1$ toutes les conductivités sont 1 et le nombre de Biot est 0.1). Calculer le numéro de condition de AN () pour $N = 8$ et pour $\cup = 1$ et $\cup = 10$ avec et sans orthonormalisation Gram { Schmidt. Qu'est-ce que tu observes ? Résolvez l'approximation de base réduite (où vous utilisez les instantanés directement en Z) pour $\cup = 0.1$ et $N = 8$. Qu'est-ce que $u_N (\cup)$? Comment vous attendez-vous à ce que $u_N (\cup)$ ressemble à $\cup = 10.0$? Qu'en est-il de $\cup = 1.0975$? Résoudre le Gram { Schmidt orthonormalized approximation de base réduite pour $\cup = 0.1$ et $\cup = 10$ pour $N = 8$. Que observez-vous? Pouvez-vous justifier le résultat ? Pour les questions restantes, vous devez utiliser l'approximation de base réduite orthonormalisée Gram { Schmidt.
2. Vérifier que, pour $\cup = 1.5$ (rappeler que Biot est toujours fixé à 0.1) et $N = 8$, la valeur de la sortie est $T_{rootN} (\cup) = 1.53107$.
3. Nous introduisons ensuite un échantillon de test régulier, test D, de taille $n_{test} = 100$ (dans MATLAB, vous pouvez simplement utiliser `linspace(0.1, 10, 100)` pour générer le test). Tracer la convergence du maximum erreur relative dans la norme d'énergie $\max_{\cup} \frac{\|j_{jj}u(\cup) - u_N(\cup)\|}{\|j_{jj}u(\cup)\|}$ et l'erreur de sortie relative maximale $\max_{\cup} \frac{|T_{root}(\cup) - T_{rootN}(\cup)|}{|T_{root}(\cup)|}$ en fonction de N (utiliser la semilogie de commande MATLAB pour le tracé).
4. Comparer le temps moyen de l'unité centrale sur l'échantillon de test nécessaire pour résoudre l'étape en ligne de base réduite avec la solution directe de l'approximation du technicien de maintenance en fonction de N.
5. De quelle valeur de N avez-vous besoin pour obtenir une précision relative de la sortie de 1%. A quelle économie de temps CPU correspond ce % ?
6. Résoudre les problèmes b) 3. à 5. en utilisant le médium et la triangulation ne FE. La dépendance à l'égard de l'azote est-elle ce que vous prévoyez?

c) Nous considérons maintenant un autre paramètre ($P = 1$) problème. Cette fois, nous supposons que les conductivités sont fixées à $f_{k1}; k_2; k_3; k_4g = f_{0.4}; 0.6; 0.8; 1.2g$, et que seul le numéro Biot, Bi, est autorisé à varier de 0.01 à 1. L'ensemble d'échantillons SN pour $N_{max} = 11$ est donné dans les données le RB_sample.sample2. Générer un Z orthonormal à partir de l'ensemble d'échantillons à l'aide de la triangulation moyenne.

Vérifier que, pour $\cup = 0.4; 0.6; 0.8; 1.2; 0.15$, i.e. $Bi = 0.15$, la valeur de la sortie est $T_{rootN} (\cup) = 1.51561$.

Nous introduisons ensuite un échantillon de test régulier, test D, de taille $n_{test} = 100$ (dans MATLAB, vous pouvez simplement utiliser `linspace(0.01, 1, 100)` pour générer le test). Tracer la convergence de l'erreur relative maximale dans la norme d'énergie $\max_{\cup} \frac{\|j_{jj}u(\cup) - u_N(\cup)\|}{\|j_{jj}u(\cup)\|}$

y_j et l'erreur de sortie relative maximale $\max_{j=1, \dots, N} \frac{|y_j - T_{\text{root}}(y_j)|}{|y_j|}$ en fonction de N (utiliser la semilogie de commande MATLAB pour le tracé).

Le numéro Biot est directement lié à la méthode de refroidissement; des taux de refroidissement plus élevés (B_i plus élevé) impliquent des coûts initiaux et opérationnels plus faibles (meilleurs) mais aussi plus élevés (pires). On peut donc définir une fonction de coût total comme

$$C(B_i) = B_i + T_{\text{root}}(B_i); \quad (5)$$

dont la minimisation offre une solution optimale. Appliquer votre base réduite (en ligne) { approximation for $T_{\text{root}}N$ (c'est-à-dire remplacer $T_{\text{root}}(B_i)$ en (5) par $T_{\text{root}}N(B_i)$) pour trouver le B_i optimal :

Toute (simple) procédure d'optimisation sur ces pour la minimisation.

d) Enfin, nous considérons un problème de deux paramètres ($P = 2$) où les conductivités sont supposées être équivalentes, c.-à-d. $k_1 = k_2 = k_3 = k_4$, mais peuvent varier entre 0:1 et 10; et le numéro Biot, B_i , peut varier de 0:01 à 1. L'ensemble d'échantillons SN pour $N_{\text{max}} = 46$ est donné dans les données le RB_sample.sample3. Générer un Z orthonormal à partir de l'ensemble d'échantillons à l'aide de la triangulation grossière.

1. 1. Nous introduisons ensuite une grille régulière. test D, de
taille $n_{\text{test}} = 400$ (une régulière 20 20 grille).

Tracer la convergence de l'erreur relative maximale dans la norme d'énergie $\max_j \frac{\|u_j - u_{\text{root}}\|}{\|u_j\|}$ et l'erreur relative de sortie maximale $\max_j \frac{|T_{\text{root}}(B_i) - T_{\text{root}}N(B_i)|}{|T_{\text{root}}(B_i)|}$ en tant que fonction de N (utiliser la semilogie de commande MATLAB pour le tracé).

1,2 Annexe 1 { Mise en œuvre de la méthode des éléments finis

Pour la mise en œuvre de la méthode de base réduite, les matrices d'éléments finis pour trois triangulations possibles du problème n sont fournies. Pour obtenir les données matlab requises, téléchargez le PS2_matlab.zip sur le site Web du cours et dézippez-le. Il y a trois .mat les : FE_matrix.mat contient les matrices FE, FE_grid.mat contient les données de triangulation, et RB_sample.mat contient les échantillons que vous devriez utiliser initialement (plus tard vous générerez les échantillons vous-même en utilisant une procédure gourmande). Pour charger les matrices FE dans l'espace de travail MATLAB :

```
>> load FE_matrix
```

Cela crée une variable nommée FE_matrix avec trois elds grossiers, moyens et ne. Chacun de ces champs contient un tableau de cellules Ahq de taille 6 1 et le vecteur de charge Fh. Chaque cellule d'Ahq contient le paramètre { independent FE matrix AqN ; q = 1; :::; 6; ici q = 1; :::; 4 correspond aux "submatrices" de ns 1; :::; 4, avec conductivités k_i ; i = 1; :::; 4, respectivement; q = 5 correspond à la "sous-oreillette" du poteau central avec une conductivité $k_0 = 1$; et q = 6 correspond à la "submatrix" de la ligne intégrale sur la "surface" du n (sans racine). Pour charger les échantillons de base réduits SN dans l'espace de travail MATLAB :

```
>> load RB_sample
```

Cela crée une variable nommée RB_sample avec elds sample1, sample2 et sample3, correspondant aux deux cas $P = 1$ et $P = 2$ décrits dans l'instruction problem. Noter que la triangulation est nécessaire uniquement pour tracer le tracé de la solution du technicien de maintenance (voir ci-dessous). Les informations détaillées suivantes sur la triangulation viennent d'être incluses pour vous donner une impression concernant les données requises si vous souhaitez configurer les matrices FE à partir de zéro. Pour charger les données de triangulation dans l'espace de travail MATLAB :

```
>> load FE_grid
```

Cela crée une variable FE_grid avec trois elds grossiers, moyens et ne. Chacun de ces elds est une triangulation Th pour le problème n . Plus spécifiquement

grossier de ns Th_grossier , avec 1333 nœuds et 2095 éléments.

medium de ns Th_medium , avec 4760 nœuds, et 8380 éléments, et

ne de ns Th_fine , avec 17889 nœuds, et 33520 éléments.

Chacune de ces variables est de type struct, elds.


```
>> grossier
er
grossier =
    nœuds : 1333
    coor : [1333x2 double]
    éléments : 2095
    theta : 1x7 cell
```

Description de la `grossier` : (supposons que nous utilisons la triangulation grossière)

`noeuds` : Le nombre de noeuds dans la triangulation.

`coor` : Matrice bidimensionnelle avec taille (noeuds 2), où chaque ligne i a les coordonnées x et y pour le noeud i . Par exemple, l'emplacement du noeud 49 peut être déterminé par deux coordonnées. La coordonnée dans la direction x { serait `grossier.coor(49,1)` et dans la direction y { `grossier.coor(49,2)`.

`elements` : Le nombre d'éléments dans la triangulation.

`theta` : La matrice de contiguïté ($k; i$) qui définit la cartographie globale locale { à { requise dans la procédure d'assemblage élémentaire. Comme nous avons des régions ayant des propriétés physiques différentes, une matrice de contiguïté distincte est fournie pour chaque région. Les régions considérées sont

```
{ Région 1 : Domaine 1; 1(k; i) = grossier.theta1,
{ Région 2 : Domaine 2; 2(k; i) = grossier.theta2,
{ Région 3 : Domaine 3; 3(k; i) = grossier.theta3,
{ Région 4 : Domaine 4; 4(k; i) = grossier.theta4, Région 5 : Domaine 0; 5(k; i) = grossier.theta5.
```

Pour chacune de ces régions i , l'indice k varie dans l'intervalle $k \in [1; n_i]$, où n_i sont le nombre d'éléments dans la région i . Par exemple, l'élément 12 dans la région 3 a les nœuds globaux $1 = \text{grossier.theta3}(12,1)$, $2 = \text{grossier.theta3}(12,2)$, et $3 = \text{grossier.theta3}(12,3)$. En outre, pour le traitement des conditions limites, la limite est divisée en deux sections. Le `root` est la racine, où les conditions limites de Robin sont appliquées; la seconde est la racine; où la chaleur entrante `ux` est appliquée. Pour chaque segment de ces sections, les nœuds globaux associés sont fournis.

```
{ Section 1 : root; 1(m; i) = grossier.theta6,
{ Section 2 : root; 2(m; i) = grossier.theta7.
```

Pour chacune des sections i , l'indice m varie dans l'intervalle $m \in [1; s_i]$, où s_i est le nombre de segments dans la section i . Par exemple, pour trouver les nœuds 1 , et 2 pour le segment 5 dans la première section, nous utiliserions $1 = \text{grossier.theta6}(5,1)$, et $2 = \text{grossier.theta6}(5,2)$.

Pour tracer la distribution de la température, tracez la solution. `m` peut être utilisé. Si `z` est le vecteur avec les valeurs de température calculées pour chacun des nœuds, alors un tracé de contour de la distribution de température peut être obtenu par

```
>> traceur (FE_grid.coarse, z)
```

Le premier argument est le maillage utilisé dans le calcul de `z`, et le second est le vecteur de solution `z` :

Pour le stockage des matrices d'éléments de nite, utilisez la structure de données de matrice clairsemée de MATLABs. De plus, pour la solution des systèmes linéaires résultants, utiliser les méthodes de solution par défaut fournies dans MATLAB, c.-à-d. utiliser

```
>> u=A \ F
```

à résoudre pour la solution FEM `u` :

Remerciements : Je tiens à remercier le Prof. A.T. Patera et Debbie Blanchard de nous avoir fourni certains des documents pour cet ensemble de problèmes et de nous avoir permis de les utiliser.