

# Analyse numérique et optimisation

## Réduction de modèles

Alexandre Ern

ern@cermics.enpc.fr

### 1 Motivations

Dans ce projet, on s'intéresse à un problème elliptique (pour fixer les idées, une équation de diffusion en une ou deux dimensions d'espace) où le coefficient de diffusion dépend d'un paramètre, et on suppose qu'on a besoin de résoudre le problème pour un nombre important de valeurs de ce paramètre. Un exemple typique (que l'on considérera ci-après) est celui où le coefficient de diffusion est incertain (du fait, par exemple, d'une caractérisation expérimentale insuffisante), ce qui conduit à le décrire par le biais d'une variable aléatoire dont on se fixe a priori la loi de distribution.

Dans ce contexte, résoudre un nombre important de fois le problème elliptique (par la méthode des éléments finis par exemple) devient extrêmement onéreux. Le principe de la méthode des bases réduites est de procéder à ces évaluations en deux temps : (1) on résout d'abord le problème elliptique (par la méthode des éléments finis) pour un nombre relativement modeste de valeurs du paramètre, puis (2) on utilise les solutions ainsi obtenues pour engendrer un espace vectoriel de dimension réduite où l'on cherchera une approximation de la solution du problème elliptique pour toutes les autres valeurs du paramètre ; cette approximation sera obtenue en ayant recours à la méthode de Galerkin mais en évitant la discrétisation par éléments finis.

### 2 Étude du cas unidimensionnel

On se place sur l'intervalle  $\Omega = ]0, 1[$  et on considère un problème de diffusion sous la forme

$$-\frac{d}{dx} \left( \kappa(x) \frac{du}{dx}(x) \right) + u(x) = f(x), \quad x \in \Omega, \quad (1)$$

où  $\kappa$  est le coefficient de diffusion et  $f$  le terme source. Par la suite, on prendra toujours  $f(x) = \sin(2\pi x)$  sur  $\Omega$  et on imposera des conditions de Dirichlet homogènes sur le bord de  $\Omega$ . Le coefficient de diffusion  $\kappa$  est variable (constant par morceaux) en espace et s'écrit sous la forme

$$\kappa(x) = \begin{cases} 1 & x \in \Omega_1, \\ \mu & x \in \Omega_2, \end{cases} \quad (2)$$

où  $\Omega_2 := ]0.19; 0.21[ \cup ]0.38; 0.42[ \cup ]0.58; 0.62[ \cup ]0.79; 0.81[$ ,  $\Omega_1 := \Omega \setminus \Omega_2$  et  $\mu \in \mathcal{P}$  où  $\mathcal{P}$  est un ensemble de valeurs strictement positives. Le coefficient  $\mu$  sert à paramétrer le coefficient de diffusion ; les ensembles  $\Omega_1$  et  $\Omega_2$  sont,

quant à eux, toujours fixés. Pour plus de clarté, on explicitera par la suite la dépendance de  $\kappa$  en  $\mu$  et on écrira  $\kappa_\mu(x)$ . La solution  $u$  de (1) dépend elle aussi de  $\mu$ , et on la notera  $u_\mu(x)$ .

En résumé, pour tout  $\mu \in \mathcal{P}$ , on cherche la fonction  $u_\mu(x) : \Omega \rightarrow \mathbb{R}$  telle que

$$\begin{cases} -\frac{d}{dx} \left( \kappa_\mu(x) \frac{du_\mu}{dx}(x) \right) + u_\mu(x) = f(x), & x \in \Omega, \\ u_\mu(0) = u_\mu(1) = 0. \end{cases} \quad (3)$$

Un cadre d'application (idéalisé!) du problème (3) est la modélisation d'un matériau composite contenant des inclusions dont on connaît la position (l'ensemble  $\Omega_2$ ). Le coefficient de diffusion dans les inclusions est  $\mu$ .

**Question 1** Établir la formulation variationnelle du problème (3) dans l'espace  $V = H_0^1(\Omega)$ . On notera  $a_\mu$  la forme bilinéaire associée (définie sur  $V \times V$ ) et  $b$  la forme linéaire associée (définie sur  $V$ ).

**Question 2** On considère une approximation par la méthode des éléments finis de Lagrange de degré 1, l'intervalle  $\Omega$  étant maillé de manière uniforme en utilisant  $(N+1)$  mailles. En posant  $h = \frac{1}{N+1}$  le pas de maillage, la solution approchée est notée  $u_{h,\mu}$ . On introduit le vecteur  $X_\mu$  (d'ordre  $N$ ) formé par les composantes de  $u_{h,\mu}$  dans la base canonique des fonctions chapeau  $(\varphi_1, \dots, \varphi_N)$ . Le vecteur  $X_\mu$  est solution du système linéaire (d'ordre  $N$ )

$$A_\mu X_\mu = B. \quad (4)$$

Pour tout  $0 \leq i \leq N$ , on note  $K_i$  la maille  $[x_i, x_{i+1}]$  avec  $x_i = ih$ . Par la suite, on suppose que chaque maille  $K_i$  se trouve soit dans  $\overline{\Omega}_1$  soit dans  $\overline{\Omega}_2$ . Évaluer les coefficients de la matrice de rigidité  $A_\mu$  et du vecteur  $B$ .

**Question 3** Écrire un code Scilab afin d'approcher, pour une valeur de  $\mu \in \mathcal{P}$  fixée, la solution fournie par la méthode des éléments finis. On prendra  $N = 999$ . Tracer les solutions  $u_{h,\mu}(x)$  pour  $\mu = 0.01$ ,  $\mu = 0.1$  et  $\mu = 1$  et commenter les profils obtenus.

**Question 4** La méthode des bases réduites. On choisit un sous-ensemble  $\mathcal{P}_0 \subset \mathcal{P}$  de cardinal fini  $N_0$  avec  $N_0 \ll N$  ( $N_0$  sera de l'ordre de 3 à 5). On énumère les éléments de  $\mathcal{P}_0$  sous la forme  $\{\mu_1, \dots, \mu_{N_0}\}$ . Pour tout  $1 \leq i \leq N_0$ , on introduit la fonction  $u_i : \Omega \rightarrow \mathbb{R}$  telle que

$$u_i(x) := u_{h,\mu_i}(x), \quad \forall 1 \leq i \leq N_0. \quad (5)$$

Ces fonctions sont obtenues en résolvant le problème approché par éléments finis pour chaque  $\mu_i$ . Par la suite, on supposera que les  $N_0$  fonctions  $u_i$  sont

indépendantes. On pose

$$V_0 := \text{vect}(u_1, \dots, u_{N_0}). \quad (6)$$

Il s'agit d'un sous-espace de  $H_0^1(\Omega)$  de dimension finie  $N_0$ . Soit maintenant  $\mu \in \mathcal{P}$  donné. On approche la fonction  $u_\mu$  solution de (3) par une méthode de Galerkin dans  $V_0$ , ce qui revient à chercher la fonction  $u_\mu^{\text{RB}} \in V_0$  telle que

$$a_\mu(u_\mu^{\text{RB}}, u_i) = b(u_i), \quad \forall 1 \leq i \leq N_0. \quad (7)$$

En décomposant  $u_\mu^{\text{RB}}$  sous la forme  $u_\mu^{\text{RB}} = \sum_{i=1}^{N_0} X_{\mu,i}^{\text{RB}} u_i$ , montrer que le vecteur  $X_\mu^{\text{RB}} \in \mathbb{R}^{N_0}$  est solution d'un système linéaire (d'ordre  $N_0$ ) de la forme

$$A_\mu^{\text{RB}} X_\mu^{\text{RB}} = B^{\text{RB}}. \quad (8)$$

où on donnera l'expression des composantes de la matrice  $A_\mu^{\text{RB}}$  et du vecteur  $B^{\text{RB}}$ . On justifiera le caractère symétrique défini positif de la matrice  $A_\mu^{\text{RB}}$ .

**Question 5** On observe que la grande matrice de rigidité  $A_\mu$  peut se décomposer sous la forme  $A_\mu = A_0 + \mu A_1$  où  $A_0$  et  $A_1$  sont deux matrices d'ordre  $N$  indépendantes de  $\mu$ . Exprimer les termes génériques de ces deux matrices en utilisant les fonctions chapeau. Pour tout  $1 \leq i \leq N_0$ , on note  $X_i$  le vecteur de  $\mathbb{R}^N$  formé par les composantes de la fonction  $u_i$  dans la base des fonctions chapeau  $(\varphi_1, \dots, \varphi_N)$ . Montrer que

$$A_\mu^{\text{RB}} = A_0^{\text{RB}} + \mu A_1^{\text{RB}}, \quad (9)$$

où  $A_0^{\text{RB}}$  et  $A_1^{\text{RB}}$  sont deux matrices d'ordre  $N_0$  indépendantes de  $\mu$  dont on précisera les composantes en fonction des matrices  $A_0$  et  $A_1$  et des vecteurs  $(X_i)_{1 \leq i \leq N_0}$  (on observera que les deux matrices  $A_0^{\text{RB}}$  et  $A_1^{\text{RB}}$  sont de petite taille mais pleines). Donner également l'expression des composantes du vecteur  $B^{\text{RB}}$ .

**Question 6** Implémentation sous Scilab. On structurera le script de la manière suivante :

1. évaluation de la grande matrice de rigidité  $A_\mu$  (en fait, des deux grandes matrices  $A_0$  et  $A_1$ ) et du membre de droite  $B$  ;
2. choix de l'ensemble  $\mathcal{P}_0$  et calcul des fonctions de la base réduite  $(u_i)_{1 \leq i \leq N_0}$  en résolvant le grand système linéaire (4) pour chaque valeur du paramètre  $\mu \in \mathcal{P}_0$  ; on stockera ces solutions dans une famille de vecteurs  $(X_1, \dots, X_{N_0})$ , chaque vecteur étant de taille  $N$  ;
3. évaluation des matrices  $A_0^{\text{RB}}$  et  $A_1^{\text{RB}}$  et du vecteur  $B^{\text{RB}}$  ;

4. pour chaque valeur de  $\mu \in \mathcal{P}$  dont on souhaite connaître la solution correspondante  $u_\mu$ , résolution du petit système linéaire (8); un point d'implémentation important est que dans cette quatrième partie du script, aucune boucle de longueur  $N$  ne doit être utilisée.

Application numérique : on prendra  $N_0 = 3$  avec les valeurs  $\mu_1 = 0.05$ ,  $\mu_2 = 0.2$  et  $\mu_3 = 1$  pour évaluer les fonctions de la base réduite. Puis, on choisira  $\mu = 0.075$  et on comparera sur un même graphique  $u_\mu$  et  $u_\mu^{\text{RB}}$ ; on tracera également l'erreur  $|u_\mu - u_\mu^{\text{RB}}|$  en fonction de  $x \in \Omega$ .

**Question 7** Application à l'échantillonnage dans le cas d'un modèle probabiliste d'incertitudes sur le coefficient de diffusion. Pour cette question, on considère que  $\mu$  est une variable aléatoire de loi uniforme sur  $I_\mu := [0.01, 1]$ , ce que l'on note

$$\mu \sim \mathcal{U}([0.01, 1]). \quad (10)$$

En utilisant le script développé à la question précédente, évaluer la fonction de  $\Omega$  dans  $\mathbb{R}$  définie par

$$\mathbb{E}(u_\mu(x)) := \int_{I_\mu} u_\mu(x) d\mu. \quad (11)$$

On utilisera la méthode de Monte Carlo consistant à effectuer  $M$  tirages de  $\mu$ ,  $\{\mu^{(1)}, \dots, \mu^{(M)}\}$ , et à remplacer l'espérance par la moyenne empirique

$$\mathbb{E}^{\text{MC}}(u_\mu(x)) := \frac{1}{M} \sum_{i=1}^M u_{\mu^{(i)}}(x). \quad (12)$$

Sous Scilab, des appels à la fonction `rand()` permettent de tirer une variable aléatoire suivant la loi  $\mathcal{U}([0, 1])$ . On tracera sur un même graphique les fonctions  $\mathbb{E}^{\text{MC}}(u_\mu(x))$  (avec  $M = 1000$ ) et  $u_{\bar{\mu}}(x)$  où  $\bar{\mu}$  est la valeur moyenne de  $\mu$  (qui ici vaut 0.505). Commenter. Reprendre l'étude lorsque  $\mu$  suit une loi log-uniforme avec

$$\log_{10} \mu \sim \mathcal{U}([-2, 0]). \quad (13)$$

### 3 Étude du cas bidimensionnel

On se place maintenant sur le carré unité  $\Omega = ]0, 1[ \times ]0, 1[$ . On considère pour terme source  $f(x, y) = \sin(2\pi x) \sin(2\pi y)$  sur  $\Omega$  et on impose des conditions de Dirichlet homogènes sur le bord  $\partial\Omega$  de  $\Omega$ . Le coefficient de diffusion  $\kappa_\mu$  est toujours défini par (2), mais cette fois le sous-ensemble  $\Omega_2$  est composé de 16 inclusions circulaires sous la forme

$$\Omega_2 := \bigcup_{1 \leq i \leq 4} \bigcup_{1 \leq j \leq 4} C_{i,j}, \quad (14)$$

où  $C_{i,j}$  est le disque de rayon 0.075 et de centre  $(x_i, y_j) = (0.2i, 0.2j)$ . Ainsi, pour chaque  $\mu \in \mathcal{P}$  fixé, on cherche la fonction  $u_\mu(x) : \Omega \rightarrow \mathbb{R}$  telle que

$$\begin{cases} -\nabla \cdot (\kappa_\mu(x) \nabla u_\mu(x)) + u_\mu(x) = f(x), & x \in \Omega, \\ u_\mu|_{\partial\Omega} = 0. \end{cases} \quad (15)$$

**Question 8** Mise en œuvre sous FreeFem++. On organisera le script comme à la question 6 ci-dessus. On utilisera les valeurs proposées à la question 6 pour l'application numérique.

Conseils FreeFem++.

- Pour définir le domaine  $\Omega$  et son maillage (les ... signifient qu'il faut faire des copié-collé)

```
border B1(t=0,1) {x=t;y=0 ;label =1;};
...
real Rc = 0.075;
border C11(t=0,2*pi){x=0.2+Rc*cos(t); y=0.2+Rc*sin(t); label=2;};
...
int Nb=50;
int Nc=20;
mesh Th=buildmesh(B1(Nb)+...+C11(Nc)+...);
```

où les paramètres Nb et Nc permettent de contrôler la finesse du maillage.

- Espaces d'éléments finis

```
fespace Vh(Th, P1);
fespace Xh(Th, P0);
```

- Matrices  $A_0$  et  $A_1$

```
Xh coeffB = (region==16);
Xh coeffC = (region<16);
varf A0p(u,v,solver=UMFPACK) =
int2d (Th) ( coeffB*(dx(u)*dx(v)+dy(u)*dy(v)) ) + on(1,u=0);
matrix A0 = A0p(Vh,Vh);
...

```

- Choix de  $\mathcal{P}_0$  et initialisation des matrices réduites  $A_0^{\text{RB}}$  et  $A_1^{\text{RB}}$

```
int nrb=3;
real [int] select(nrb);
select[0]=.1;select[1]=.5;select[2]=1.;
real [int,int] A0red(nrb,nrb);
real [int,int] A1red(nrb,nrb);
```

- Calcul des fonctions de la base réduite

```
Vh [int] solutions(nrb);
for(int k=0;k<nrb;k++){
    mu = select(k);
    A = A0+mu*A1;
    set(A,eps=1e-10);
    u[]=A^-1*F[];
    solutions[k]=u;
}
```

- Obtention de  $N$

```
int nddl= Vh.ndof;
```

- Le calcul des matrices pourra être effectué selon

```
real [int] buffer(nddl);
for(int i=0;i<nrb;i++){
    for(int j=0;j<nrb;j++){
        buffer = (A0*solutions[j] []);
        A0red(i,j) =solutions[i] []'*buffer;
        ...
    }
}
```