Examen: Méthodes de tenseurs pour les problèmes en grande dimension

V. Ehrlacher et M.-S. Dupuy

April 17, 2023

L'examen comporte 2 exercices et 5 pages. Les documents sont autorisés mais les appareils électroniques sont interdits pendant l'examen.

Exercice 1: algorithme PGD orthogonal

Le but de ce problème est d'étudier une variante de l'algorithme PGD vu en cours, appelé algorithme PGD orthogonal. Cet algorithme coûte un peu plus cher en temps de calcul que l'algorithme PGD standard, mais permet en général d'obtenir des approximations de meilleure qualité. Nous allons appliquer cet algorithme à la résolution du problème décrit ci-dessous.

Notations

Dans toute la suite de l'énoncé, si $e \in \mathbb{R}^2$, $|e|$ désigne la norme euclidienne de e. On rappelle que $H^1((0,1)^2)$ est défini comme l'ensemble des fonctions $v : (0,1)^2 \to \mathbb{R}$ mesurables telles que

$$\int_{(0,1)^2} \left(|v|^2 + |\partial_x v|^2 + |\partial_y v|^2\right) = \int_{(0,1)^2} \left(|v|^2 + |\nabla v|^2\right) < +\infty.$$

On admettra que cet espace, muni du produit scalaire

$$\forall v, w \in H^1((0,1)^2), \quad \langle v, w\rangle_{H^1((0,1)^2)} := \int_{(0,1)^2} [vw + \nabla v \cdot \nabla w] = \int_{(0,1)^2} [vw + \partial_x v \partial_x w + \partial_y v \partial_y w]$$

est un espace de Hilbert.

EDP et formulation variationnelle

Soit $f \in L^2((0,1)^2)$. On définit $a : H^1((0,1)^2) \times H^1(0,1)^2) \to \mathbb{R}$ l'application définie comme suit:

$$\forall v, w \in H^1((0,1)^2), \quad a(v,w) := \int_{(0,1)^2} [2vw + \nabla v \cdot \nabla w]\,.$$

On définit également $l : H^1((0,1)^2) \to \mathbb{R}$ l'application définie comme suit:

$$\forall v \in H^1((0,1)^2), \quad l(v) := \int_{(0,1)^2} fv.$$

On peut montrer que le problème: trouver $u \in H^1((0,1)^2)$ tel que

$$\forall v \in H^1((0,1)^2), \quad a(u,v) = l(v), \tag{1}$$

a une unique solution. Le problème (1) est la formulation variationnelle de l'EDP suivante: trouver $u \in H^1((0,1)^2)$ solution de

$$\begin{cases} -\Delta u + 2u = f \text{ sur } (0,1)^2, \\ \nabla u \cdot n = 0 \text{ sur } \partial(0,1)^2, \end{cases}$$

où n désigne la normale unitaire sortante au domaine $(0,1)^2$.

Cette fonction u est aussi solution du problème de minimisation

$$u = \underset{v \in H^1((0,1)^2)}{\operatorname{argmin}} \mathcal{E}(v), \tag{2}$$

où

$$\mathcal{E}(v) := \frac{1}{2} a(v,v) - l(v).$$

Algorithme PGD standard

Pour toute fonction $r : (0,1) \to \mathbb{R}$ et toute fonction $s : (0,1) \to \mathbb{R}$ mesurables, on rappelle que l'on définit la fonction mesurable $r \otimes s$ sur $(0,1)^2$ par

$$r \otimes s : \begin{cases} (0,1)^2 & \to & \mathbb{R} \\ (x,y) & \mapsto & r(x)s(y). \end{cases}$$

1. Montrer que si $r, s \in L^2(0,1)$, alors $r \otimes s \in L^2((0,1)^2)$.
2. Montrer que si $r, s \in H^1(0,1)$, alors $r \otimes s \in H^1((0,1)^2)$.

L'algorithme PGD est un algorithme itératif qui s'écrit comme suit dans notre contexte:

ALGORITHME PGD STANDARD:
Initialisation: Posons $u_0(x,y) = 0$.
Itération $n \geq 1$**:** soit $(r_n, s_n) \in H^1(0,1) \times H^1(0,1)$ solutions de

$$(r_n, s_n) \in \underset{(r,s) \in H^1(0,1) \times H^1(0,1)}{\operatorname{argmin}} \mathcal{E}(u_{n-1} + r \otimes s) \tag{3}$$

On pose $u_n = u_{n-1} + r_n \otimes s_n$ et on passe à l'itération suivante avec $n := n+1$.

On rappelle alors que pour tout $n \geq 1$, (par une récurrence immédiate) on a que $u_n = \sum_{k=1}^n r_k \otimes s_k$.

Pour tout $n \geq 1$, on définit

$$\mathcal{J}_{n-1} : \begin{cases} H^1(0,1) \times H^1(0,1) & \to & \mathbb{R} \\ (r,s) & \mapsto & \mathcal{E}(u_{n-1} + r \otimes s). \end{cases}$$

3. Soit $r_*, s_*, r, s \in H^1(0,1)$. Ecrire l'expression de $d\mathcal{J}_{n-1}|_{(r_*,s_*)}(r,s)$, la différentielle de $\mathcal{J}_{n-1}$ pris en l'élément $(r_*, s_*) \in H^1(0,1) \times H^1(0,1)$, appliquée en $(r,s) \in H^1(0,1) \times H^1(0,1)$ en fonction de a, l et u_{n-1}.

4. En déduire que si $(r_n, s_n) \in H^1(0,1) \times H^1(0,1)$ est une solution de (5), alors nécessairement
$$\begin{cases} \forall r \in H^1(0,1), \quad a(r_n \otimes s_n, r \otimes s_n) = l(r \otimes s_n) - a(u_{n-1}, r \otimes s_n), \\ \forall s \in H^1(0,1), \quad a(r_n \otimes s_n, r_n \otimes s) = l(r_n \otimes s) - a(u_{n-1}, r_n \otimes s). \end{cases} \tag{4}$$

5. On supposera dans toute la suite des questions que $f(x,y) = g(x)h(y)$ avec $g, h \in L^2(0,1)$. En déduire que $(r_n, s_n) \in H^1(0,1) \times H^1(0,1)$ sont solutions du problème suivant: pour tout $r \in H^1(0,1)$,
$$\|s_n\|^2_{L^2(0,1)} \left(2\langle r_n, r\rangle_{L^2(0,1)} + \langle r_n', r'\rangle_{L^2(0,1)}\right) + \|s_n'\|^2_{L^2(0,1)} \langle r_n, r\rangle_{L^2(0,1)} = \langle h, s_n\rangle_{L^2(0,1)} \langle g, r\rangle_{L^2(0,1)}$$
$$- \sum_{k=1}^{n-1} \langle s_n, s_k\rangle_{L^2(0,1)} \left(2\langle r_k, r\rangle_{L^2(0,1)} + \langle r_k', r'\rangle_{L^2(0,1)}\right) + \langle s_n', s_k'\rangle_{L^2(0,1)} \langle r_k, r\rangle_{L^2(0,1)}$$
et d'une autre équation (symmétrique en r et s) que l'on écrira.

Algorithme PGD orthogonal

Dans cette section, nous allons étudier une autre version de l'algorithme PGD, appelée algorithme PGD orthogonal. Cet algorithme s'écrit (au niveau continu) comme suit:

ALGORITHME PGD ORTHOGONAL:
Initialisation: Posons $u_0(x,y) = 0$.
Itération $n \geq 1$**:** soit $(r_n, s_n) \in H^1(0,1) \times H^1(0,1)$ solutions de
$$(r_n, s_n) \in \underset{(r,s) \in H^1(0,1) \times H^1(0,1)}{\operatorname{argmin}} \mathcal{E}(u_{n-1} + r \otimes s) \tag{5}$$
Soit ensuite $(c_1^n, \cdots c_n^n) \in \mathbb{R}^n$ solution du problème de minimisation
$$(c_1^n, \cdots c_n^n) \in \underset{(c_1, \cdots, c_n) \in \mathbb{R}^n}{\operatorname{argmin}} \mathcal{E}\left(\sum_{k=1}^n c_k r_k \otimes s_k\right). \tag{6}$$
On pose alors $u_n = \sum_{k=1}^n c_k r_k \otimes s_k$ et on passe à l'itération suivante avec $n := n+1$.

6. Quel est l'intérêt de rajouter l'étape de la résolution du problème (6) dans l'algorithme glouton?

7. Soit $n \geq 1$. Pour tout $c_1, \cdots, c_n \in \mathbb{R}$, on note $C := (c_1, \cdots, c_n) \in \mathbb{R}^n$. Montrer que
$$\mathcal{E}\left(\sum_{k=1}^n c_k r_k \otimes s_k\right) = \frac{1}{2} C^T P_n C - C^T Q_n,$$
où $P_n \in \mathbb{R}^{n \times n}$ est une matrice et $Q_n \in \mathbb{R}^n$ est un vecteur (de taille n) que l'on exprimera en fonction de f et des fonctions r_k, s_k pour $1 \leq k \leq n$.

8. Montrer que si $C_n = (c_1^n, \cdots c_n^n)$ est une solution de (6), alors C_n est solution du problème matriciel
$$P_n C_n = Q_n.$$

Exercice 2: descente de gradient et trains de tenseurs

Dans cet exercice, on considère l'équation de Poisson discrète
$$\mathcal{L}x = b, \tag{7}$$
avec $\mathcal{L} = \sum_{k=1}^{d} \mathcal{L}_k \in \mathbb{R}^{n^d \times n^d}$ où
$$\mathcal{L}_k = \underbrace{\mathrm{id}_n \otimes \cdots \otimes \mathrm{id}_n}_{k-1 \text{ fois}} \otimes h \otimes \underbrace{\mathrm{id}_n \otimes \cdots \otimes \mathrm{id}_n}_{d-k \text{ fois}}$$
et $h \in \mathbb{R}^{n \times n}$ est le Laplacien discret en 1D
$$h = \begin{bmatrix} 2 & -1 & 0 & \dots \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 \end{bmatrix}.$$

Dans cet exercice, on fixe $b_i = 1$ pour tout $1 \leq i \leq n^d$.

1. Donner une représentation en trains de tenseurs de rang 1 de b.

2. Donner une représentation en trains de tenseurs de l'opérateur $\mathcal{L}$ de rang 2.

3. Rappeler et justifier la formule d'une représentation en trains de tenseurs de la somme de deux vecteurs X, Y donnés comme des trains de tenseurs $(x_1, \dots, x_d)$ et $(y_1, \dots, y_d)$, où $x_k \in \mathbb{R}^{n \times r_{k-1}^x \times r_k^x}$ et $y_k \in \mathbb{R}^{n \times r_{k-1}^y \times r_k^y}$.

4. Faire de même pour le produit matrice-vecteur AX où $(A_1, \dots, A_d)$ est une représentation en trains de tenseurs de la matrice A, avec $A_k \in \mathbb{R}^{n \times n \times R_{k-1}^A \times R_k^A}$ et $(x_1, \dots, x_d)$, $x_k \in \mathbb{R}^{n \times r_{k-1}^x \times r_k^x}$, une représentation en trains de tenseurs de X.

On considère une résolution du système linéaire (7) par un algorithme de descente de gradient.

Algorithm 1 Descente de gradient

```
function DG(A, b, α, ε_DG)
    x = 0
    p = b
    while ‖p‖ > ε_DG do
        x = x + αp
        p = b − Ax
    end while
    return x
end function
```

5. Donner la formule récursive du rang TT de x à chaque itération. Quel est l'inconvénient de cet algorithme ?

On considère une variante 2 de l'algorithme précédent où `TTrounding(x,`$\varepsilon_{\mathrm{TT}}$`)` retourne un train de tenseurs compressé via l'algorithme HSVD.

Algorithm 2 Descente de gradient

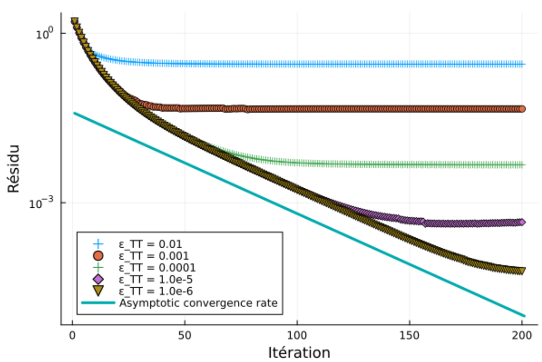
---

function DG-TT$(A, b, \alpha, \varepsilon_{\mathrm{DG}}, \varepsilon_{\mathrm{TT}})$
  $x = 0$
  $p = b$
  while $\|p\| > \varepsilon_{\mathrm{DG}}$ **do**
    $x = x + \alpha p$
    $x =$ `TTrounding`$(x, \varepsilon_{\mathrm{TT}})$
    $p = b - Ax$
    $p =$ `TTrounding`$(p, \varepsilon_{\mathrm{TT}})$
  end while
  return x
end function

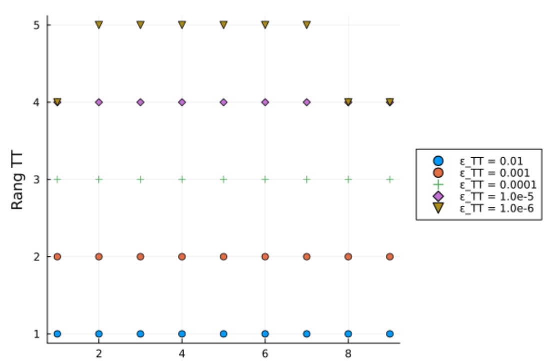
On applique l'algorithme 2 au problème (7) pour $d = 10$ et $n = 10$, $\varepsilon_{DG} = 10^{-6}$.

6. D'après la Figure 1b, estimer le coût de stockage de x en train de tenseurs. Quel est le gain par rapport au stockage du vecteur x ?

7. Interpréter la courbe de convergence 1a, en particulier, discuter des niveaux des plateaux.



(a) Résidu $Ax_k - b$ à chaque itération de l'algorithme 2 pour différentes valeurs de $\varepsilon_{\mathrm{TT}}$



(b) Rangs TT de x pour les reshapes $(x_{i_1 \dots i_k}^{i_{k+1} \dots i_d})$ lors de la 200e itération de l'algorithme 2 pour différentes valeurs de $\varepsilon_{\mathrm{TT}}$