Examen de Modélisation et calcul scientifique

Jocelyne Erhel
INSA, Rennes

Février 2015

Durée 2 heures. Tous les documents sont autorisés. Les questions ne sont pas par ordre de difficulté et peuvent être traitées dans un ordre quelconque. Il est conseillé de lire tout l'énoncé avant de commencer. Le sujet est long, le barême sera ajusté en conséquence.

On considère deux matrices carrées A et M d'ordre N, creuses, symétriques définies positives. On rappelle que A et M sont inversibles et que

$$\forall x \neq 0, x^T Ax > 0, x^T M^{-1} x > 0 \tag{1}$$

Dans tout ce qui suit, les vecteurs sont de longueur N.

1 Algorithme du Gradient Conjugué Préconditionné

Pour résoudre le système linéaire $Ax = b$, on utilise l'algorithme du Gradient Conjugué Préconditionné par M, algorithme PCG décrit dans l'algorithme 1 ci-dessous.

Le nombre réel $tol > 0$ et le nombre entier $K > 1$ sont donnés. Les mots entre * sont des commentaires.

Les variables r, z, p, q, x sont des vecteurs, les variables $\rho, \alpha, \beta, \gamma, \delta$ sont des scalaires, la variable k est un entier.

1. Montrer que si $r \neq 0$, alors $r^T z \neq 0$, où $z = M^{-1} r$. Indication: utiliser (1).
Réciproquement, montrer que si $r^T z \neq 0$, alors $r \neq 0$.
2. On admet que si $r \neq 0$, alors $p \neq 0$. En déduire que $p^T q \neq 0$, où $q = Ap$.
3. Montrer qu'il n'y a pas de division par zéro dans l'algorithme PCG.
Indication: utiliser le fait que $\rho > tol$.
4. Montrer par récurrence qu'à chaque itération de l'algorithme PCG, on a $r = b - Ax$.
5. On admet que $\|r\| \leq \mu(r^T M^{-1} r)$, où μ est une constante. On suppose que l'algorithme s'arrête parce que $\rho \leq tol$. Montrer que $\|b - Ax\| \leq \mu\, tol$. En déduire que $\|x - A^{-1}b\| \leq \|A^{-1}\|\, \mu\, tol$.
Justifier le test d'arrêt.

Algorithm 1 PCG

* Initialisation *
$x = 0$
$r = b$
$z = M^{-1}r$
$p = z$
$\rho = r^T z$
$k = 0$
* Iterations *
while $\rho > tol$ AND $k < K$ **do**
    $k = k + 1$
    $q = Ap$
    $\gamma = p^T q$
    $\alpha = \rho/\gamma$
    $x = x + \alpha p$
    $r = r - \alpha q$
    $z = M^{-1}r$
    $\delta = r^T z$
    $\beta = \delta/\rho$
    $\rho = \delta$
    $p = z + \beta p$
end while

2 Opérations BLAS1 sur deux cœurs

On exécute l'algorithme PCG sur une machine à deux cœurs avec une mémoire distribuée et on utilise une programmation parallèle, avec communications par messages. Les cœurs sont numérotés 1 et 2.

Un algorithme parallèle est défini par deux processus pouvant s'exécuter simultanément sur les deux cœurs. Pour communiquer, chaque processus utilise des algorithmes SEND et RECV d'envoi et de réception d'un vecteur ou d'un scalaire (vecteur de longueur 1) vers l'autre processus:

- SEND(x, m, c) où x est le vecteur envoyé, m est la longueur du vecteur x, c est le numéro du processus destinataire.
- RECV(x, m, c), où x est le vecteur reçu, m est la longueur du vecteur x, c est le numéro du processus expéditeur.

L'envoi d'un message est non bloquant, la réception est bloquante: le processus attend jusqu'au signal de réception du message correspondant.

Il y a interblocage dans un algorithme parallèle si les deux processus sont tous les deux bloqués sur une opération RECV.

On suppose que N est pair et on définit une partition par blocs des indices: $I_1 = \{1, \ldots, N/2\}$ et $I_2 = \{N/2+1, \ldots, N\}$. Tout vecteur x est partitioné en deux blocs, de sorte que

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

où x_1 contient les coordonnées $x(i)$ avec $i \in I_1$ et x_2 contient les coordonnées $x(i)$ avec $i \in I_2$.

On distribue le vecteur x de sorte que x_1 est dans la mémoire locale du cœur 1 et x_2 est dans la mémoire locale du cœur 2.

L'algorithme parallèle réalisant l'opération de copie $x = y$ s'écrit par exemple de la façon suivante:

Algorithm 2 copie parallèle avec deux processus

```
* processus 1*
x_1 = y_1
* processus 2*
x_2 = y_2
```

Les opérations des processus 1 et 2 sont effectuées simultanément sur les deux cœurs; ici, il n'y a pas de communication.

6. On considère l'opération $x = x + a * y$, où x et y sont des vecteurs, a est un scalaire. On suppose que a est stocké dans les mémoires locales des deux cœurs. Ecrire un algorithme parallèle réalisant cette opération sur les deux cœurs. Y a-t-il des communications ?

7. On considère l'opération $a = x^T x$, où x et y sont des vecteurs, a est un scalaire. Le scalaire résultat a sera stocké dans les mémoires locales des deux cœurs. Ecrire un algorithme parallèle réalisant cette opération sur les deux cœurs. Préciser les communications utilisées et la longueur des vecteurs envoyés. Vérifier que l'algorithme se termine sans interblocage.

3 Produit matrice vecteur en parallèle

On définit une partition par blocs de la matrice A de la façon suivante

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

où $A_{21} = A_{12}^T$, les blocs A_{11} et A_{22} sont des matrices carrées d'ordre $N/2$.

On distribue la matrice A par colonnes de sorte que les $N/2$ premières colonnes (donc les blocs A_{11} et A_{21}) sont dans la mémoire locale du cœur 1 et les $N/2$ dernières colonnes (donc les blocs A_{12} et A_{22}) sont dans la mémoire locale du cœur 2.

On considère le produit matrice vecteur $y = Ax$. Les vecteurs x et y sont distribués sur les deux cœurs comme précédemment.

8. Ecrire les formules permettant de calculer y_1 et y_2, en fonction de x_1 et x_2 et des blocs matriciels.

9. Ecrire un algorithme parallèle réalisant cette opération sur les deux cœurs. Préciser les communications utilisées et la longueur des vecteurs envoyés.
Est-il possible de recouvrir les communications par les calculs ?
Vérifier que l'algorithme se termine sans interblocage.

4 Préconditionnement bloc-Jacobi en parallèle

On définit une matrice M avec une partition par blocs de la façon suivante

$$M = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}$$

10. Montrer que A_{11} est symétrique définie positive. Indication: utiliser (1) avec le vecteur $x = \begin{pmatrix} x_1 \\ 0 \end{pmatrix}$.
Montrer que A_{22} est aussi symétrique définie positive.
En déduire que M est symétrique définie positive.

On considère l'opération $y = M^{-1}x$. Les vecteurs x et y sont distribués sur les deux cœurs comme précédemment.

11. Montrer que

$$M^{-1} = \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & A_{22}^{-1} \end{pmatrix}$$

12.Ecrire les formules permettant de calculer y_1 et y_2, en fonction de x_1 et x_2 et des blocs matriciels.
Ecrire un algorithme parallèle réalisant cette opération sur les deux cœurs.
Y a-t-il des communications?

5 Algorithme parallèle du Gradient Conjugué Préconditionné

On applique l'algorithme PCG à la matrice A ainsi partitionnée et à la matrice M définie ci-dessus.

13. Ecrire une version parallèle de l'algorithme PCG, sur deux cœurs, en utilisant les opérations parallèles des questions précédentes. Préciser les communications.

6 Cas d'une matrice issue d'un problème de diffusion

On considère un problème de diffusion stationnaire (indépendant du temps) dans un domaine 2D carré. On utilise un maillage régulier, avec n mailles dans chaque direction, soit un total de $N = n^2$ mailles. Après discrétisation, on obtient un système linéaire d'ordre N, avec une matrice A creuse, symétrique définie positive.

On numérote les cellules par lignes. La matrice A a une structure régulière, avec les éléments non nuls répartis sur cinq diagonales: $i - j = -n, -1, 0, 1, +n$.

14. Montrer que la partition des vecteurs correspond à une partition du carré en deux sous-domaines et préciser laquelle.

15. Montrer que le bloc A_{21} a la structure suivante:

$$A_{21} = \begin{pmatrix} 0 & B \\ 0 & 0 \end{pmatrix}$$

où B est une matrice diagonale d'ordre n.

16. En déduire que le vecteur $A_{21}x_1$ a la structure suivante:

$$A_{21}x_1 = \begin{pmatrix} z \\ 0 \end{pmatrix}$$

où z est un vecteur de longueur n.
Utiliser cette propriété pour réduire les communications dans le produit matrice vecteur. Quelle est maintenant la longueur des vecteurs envoyés ?

7 Partition en C sous-domaines

On considère maintenant une machine avec C cœurs, toujours à mémoire distribuée, et on veut définir un algorithme PCG parallèle avec C processus, pour la matrice issue d'un problème de diffusion.

17. Définir une partition du domaine carré en C sous-domaines. Indiquer les structures par blocs des vecteurs et des matrices A et M.

18. Préciser les communications dans les opérations BLAS1, en supposant que les scalaires sont stockés dans toutes les mémoires locales. On pourra définir une opération de communication globale pour le produit scalaire.

19. Préciser les communications dans l'opération produit matrice vecteur. Indiquer la longueur des vecteurs envoyés.

20. Ecrire l'algorithme PCG parallèle avec C processus. On écrira un seul algorithme paramétré par le numéro c du processus.