

M2 Projet en CUDA 2020-21

Dany Uy & Jijun Tang

Sorbonne Université

4 janvier 2021

- 1 Introduction
- 2 Méthode de Thomas
- 3 Réduction cyclique
- 4 Réduction cyclique parallèle
- 5 Performances et applications
- 6 Conclusion

1 Introduction

OBJECTIF :

Résoudre un système de la forme $Ax = d$, avec d un vecteur de \mathbb{R}^n connu et x un vecteur de \mathbb{R}^n à déterminer.

$$A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & 0 \\ & a_3 & b_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ 0 & & & \ddots & \ddots & c_{n-1} \\ & & & & b_n & a_n \end{pmatrix} \in M_n(\mathbb{R}).$$

2 Méthode de Thomas

- Elimination de Gauss-Jordan (n étapes) :

$$c'_1 = \frac{c_1}{b_1}, c'_i = \frac{c_i}{b_i - c'_{i-1}a_i}, i = 2, \dots, n-1$$

$$d'_1 = \frac{d_1}{b_1}, d'_i = \frac{d_i - d'_{i-1}a_i}{b_i - c'_{i-1}a_i}, i = 2, \dots, n-1$$

- Substitution ascendante (n étapes) :

$$x_n = d'_n, x_i = d'_i - c'_i x_{i+1}, i = 2, \dots, n-1.$$

3 Réduction cyclique

PRINCIPE :

- Réduction descendante sur les équations d'indices pairs ($\log_2(n) - 1$ étapes) :

$$k_1 = \frac{a_i}{b_{i-1}}, k_2 = \frac{c_i}{b_{i+1}},$$

$$a'_i = -a_{i-1}k_1, b'_i = b_i - c_{i-1}k_1 - a_{i+1}k_2,$$

$$c'_i = -c_{i+1}k_2, d'_i = d_i - d_{i-1}k_1 - d_{i+1}k_2.$$

- Résolution du système à deux inconnues (par substitution, 1 étape).
- Résolution des équations restantes (par substitution ascendante ($\log_2(n) - 1$ étapes) :

$$x_i = \frac{d'_i - a'_i x_{i-1} - c'_i x_{i+1}}{b'_i}.$$

IMPLÉMENTATION : chaque thread calcule les indices des équations dont il a besoin

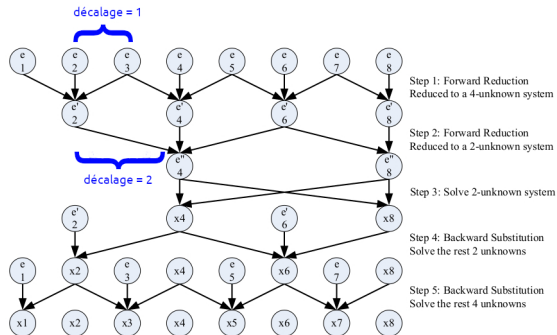


Figure 1. Communication pattern for CR in the 8-unknown case, showing the dataflow between each equation, labeled e_1 to e_8 . Letters e'_i and e''_i stand for updated equation.

4 Réduction cyclique parallèle

PRINCIPE :

- Réduction descendante sur toutes les équations ($\log_2(n) - 1$ étapes) avec les mêmes équations que la réduction cyclique.
- Résolution des systèmes à deux inconnues (1 étape).

IMPLÉMENTATION : chaque thread calcule les indices des équations dont il a besoin

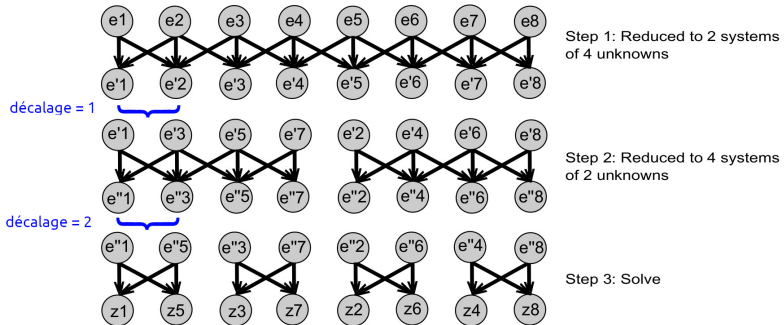


Figure 2: Communication pattern for PCR in the 8-unknown case, showing the dataflow between each equation, labeled e_1 to e_8 . Letters e' and e'' stand for updated equations.

5 Performances et applications

Résultats des tests sur Google Colab :

Méthodes	Matrice 1024x1024	Matrice 1023x1023
Thomas	10.8 microsecondes	
CR	14.3 microsecondes	
PCR	4.4 microsecondes	
PCR_odd		8.2 microsecondes

6 Conclusion

- CR est le plus lent des 3 dans la version actuelle
- PCR permet un gain notable en temps de calcul.