

大規模並列環境における多重並列線形計算手法

次世代ペタスケール計算のための線形計算フレームワークの開発

櫻井鉄也¹, 多田野寛人²,
Serge Petiton³, Nahid Emad⁴

¹ 筑波大学大学院 システム情報工学研究科

² 筑波大学 計算科学研究センター

³ INRIA, France

⁴ Univ. Versailles, France

2009 年 5 月 15 日

はじめに

- ① コア数の増加に伴う問題と新しい解法の必要性
大域的通信の少ない解法
耐故障性を備えた解法
- ② 性質の異なる多くの解法
計算対象や計算機環境に応じた解法や前処理法の選択
方法ごとに複数のパラメータ設定が必要
- ③ 並列環境や問題規模に依存するコード
計算環境や手法の複雑さを意識することなく利用可能な高度なツール
コンポーネント型のライブラリ
- ④ オープンソースの活用
INRIA(フランス) で開発している Scilab ベースの並列計算

はじめに

プロトタイプ段階での記述性 — Scilab で表記したアルゴリズム —

Jacobi 法

```
n = size(A,'r');
D = sparse([1:n;1:n]',diag(A));
x = zeros(n,1);
for i=1:itrmax
    r = b - A*x;
    if norm(r)<eps,break,end;
    x = x + D\r;
end
```

共役勾配法

```
n = size(A,'r');
x = zeros(n,1);
r = b; p = r;
for i=1:itrmax
    Ap = A*p;
    alpha = (p'*r)/(p'*Ap);
    x = x + alpha*p;
    r = r - alpha*Ap;
    if norm(r)<eps,break,end;
    beta = -(r'*Ap)/(p'*Ap);
    p = r + beta*p;
end
```

大規模化, 並列化の影響評価ができない.

はじめに

- ① 超並列環境を想定した新しい計算手法開発
大域的通信の少ない方法や計算継続性を備えた方法などの新しい解法
- ② 基盤ソフトウェアの開発
次世代計算環境において効率的に大規模線形計算アプリケーションを開発・実行するための基盤ソフトウェア
- ③ 記述性の高いプロトタイプ実験環境
計算機リソースとその上で実行するアプリケーションの間をつなぐプログラムの開発・実行環境

- 櫻井 鉄也 (筑波大学)
- 多田野寛人 (筑波大学)
- Serge Petiton (INRIA and CNRS/LIFL)
- Nahid Emad (University of Versailles)

海外在住の S. Petiton および N. Emad は利用制限があるため、分散型の固有値解法ライブラリのために必要な技術開発にテーマを絞り込んだ。

- 階層的並列構造を持つ固有値解法
 - 周回積分によるフィルタリング
- 大域的固有値分布の推定
 - プロファイリング
- 計算継続性の検証
 - 一部のノードやネットワークに障害が発生したときの影響

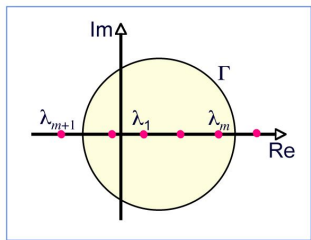
対象とする並列固有値解法

周回積分を用いた固有値解法

複素平面上の周回積分を用いて目的とする部分空間を抽出する.

一般化固有値問題 $Ax = \lambda Bx$ の固有対 : (λ_i, x_i) , $1 \leq i \leq n$

m 個の固有値 $\lambda_1, \dots, \lambda_m$ が Jordan 曲線 Γ 内にあるとする.



零でないベクトル $v \in \mathbb{C}^n$ に対して

$$s_k = \frac{1}{2\pi i} \int_{\Gamma} z^k (zB - A)^{-1} B v dz$$

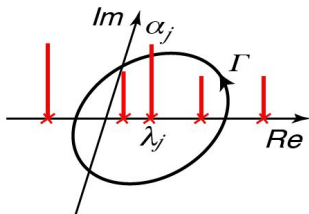
とする. このとき,

$$\text{span}\{s_0, \dots, s_{m-1}\} = \text{span}\{x_1, \dots, x_m\}$$

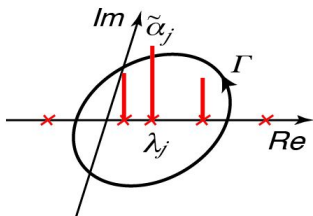
T. Sakurai and H. Tadano, Hokkaido Math. J. 36 (2007).

入力ベクトルに対するフィルター

周回積分は入力ベクトル v に対するフィルターと見なせる。



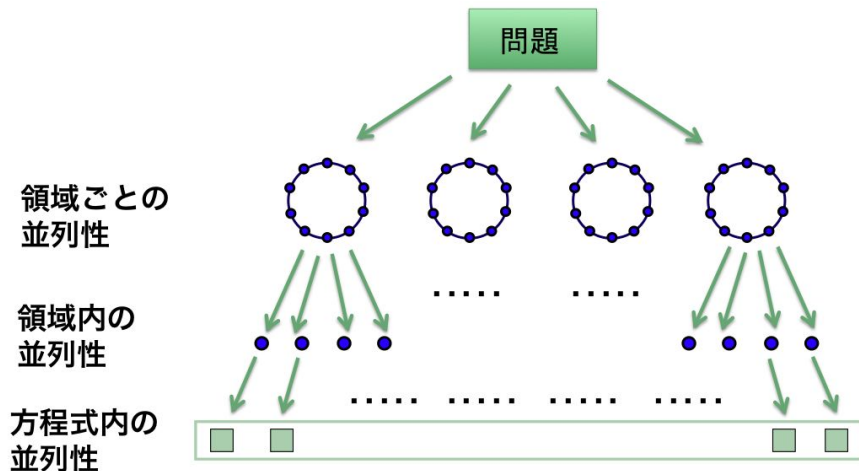
$$v = \sum_{j=1}^n \alpha_j x_j$$



$$s_k = \frac{1}{2\pi i} \int_{\Gamma} z^k (zB - A)^{-1} Bv \, dz$$

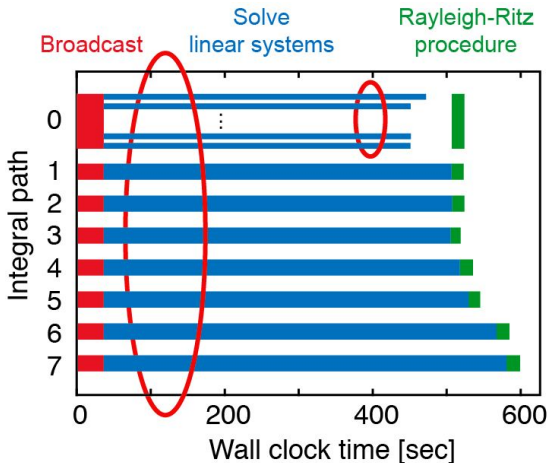
$$s_k = \sum_{j=1}^m \alpha_j \lambda_j^k x_j$$

階層的な並列構造



階層的な並列構造

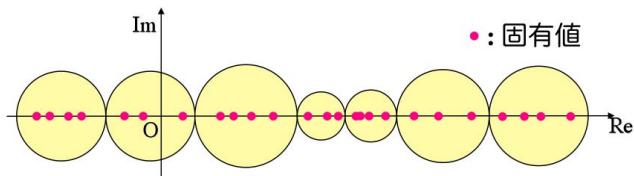
1つの領域に16ノードを割り当てた実験結果. 8つの領域を計算.



各計算ノードで起きた障害の影響は限定的.

固有値分布にあわせた領域設定

分散して計算を行うためには、最初に領域を設定する必要がある。



代数的部分構造 (AS) 法

行列を階層的な部分構造に順序付けし，部分空間を再帰的に求める

$$A = \begin{bmatrix} A_{11} & A_{13} \\ & A_{22} & A_{23} \\ A_{13}^T & A_{23}^T & A_{33} \end{bmatrix}, B = \begin{bmatrix} B_{11} & B_{13} \\ & B_{22} & B_{23} \\ B_{13}^T & B_{23}^T & B_{33} \end{bmatrix}$$

Block Gaussian Elimination

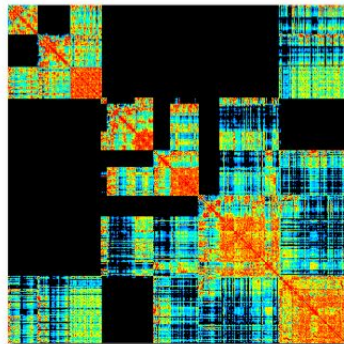
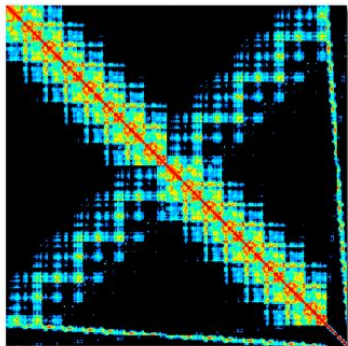
$$\hat{A} = \begin{bmatrix} A_{11} & & \\ & A_{22} & \\ & & \hat{A}_{33} \end{bmatrix}, \hat{B} = \begin{bmatrix} B_{11} & & \hat{B}_{13} \\ & B_{22} & \hat{B}_{23} \\ \hat{B}_{13}^T & \hat{B}_{23}^T & \hat{B}_{33} \end{bmatrix}$$

Rayleigh-Ritz Procedure

近似固有値

部分構造化による行列要素配置

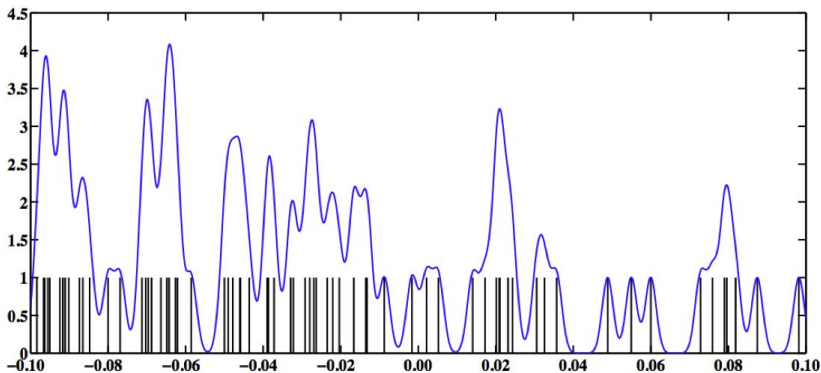
Nested Dissection Ordering による部分構造化



AS 法は広範囲の固有値分布を低精度で求めることに向いた方法
他の方法の初期設定のために用いる

AS 法による固有値分布推定

AS 法で得られた固有値分布から固有値の密集度を判定し領域を分割する



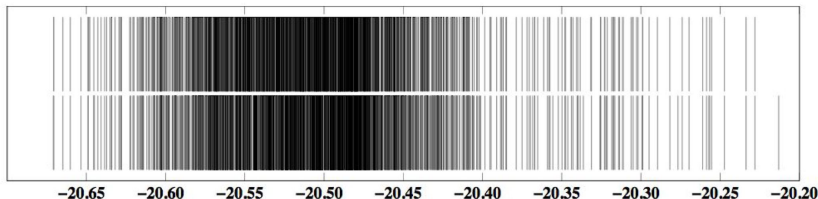
AS 法による固有値分布推定

EGFR dimmer(約 10 万次元) : 1 つのシフトあたり 1 ノードで約 700 秒で計算.

—20.5 付近の固有値計算の結果

上段 : AS 法の結果を利用して Block SS 法を適用して得られた結果 1,682 個

下段 : AS 法による推定



EGFR dimer には 0 が 1,682 個含まれ, 関係する固有値が -20.5 付近に現れる.

階層的並列固有値解法の開発：

- 代数的部分構造化法による大域的固有値分布の推定
複数シフトでの分布推定を同時に計算する
- 周回積分を用いた固有値解法による通信の少ない解法
分割した領域間の通信が発生しない
1 ノードでの障害が他に与える影響が限定的
- オーダリング後のブロック行列に対する BLAS の適用
疎行列処理にできるだけ密行列向きの処理を取り入れる

今後の課題：

- 超並列環境を想定した数値計算ライブラリの開発
- 再利用性・記述性の高いアルゴリズム開発環境の提供
- 計算の局所性による耐障害性の検証