

シフト線型方程式に対するMINRES法の適用と性能評価

日高 俊太郎*

* 電気通信大学 情報理工学研究科 情報・ネットワーク工学専攻 山本有作研究室



研究目的

本研究では、標準シフト線型方程式

$$(A + \sigma^{(k)}I)\mathbf{x}^{(k)} = \mathbf{b}, \quad (k = 1, \dots, M).$$

に対する解法である shifted MINRES法[1]の効率的な並列計算モデルの検討を行う。 A は**実対称・エルミート**行列で σ_k は**複素数**, I は**単位行列**であるとする。

研究背景

標準シフト線型方程式 (1) は、量子力学や電子構造計算などに現れる。特に近年では、行列関数の計算における部分問題としても重要性が増している。こうした問題では、 **$10^7 \sim 10^8$ 次元を超える超大規模行列**が登場することもあり、従来の逐次的な解法では対応が困難である。

このような制約下では、次のような特性を持つアルゴリズムが求められる：

- Matrix-free（行列全体を保持しない）
- 複数シフトに対する同時解法
- 並列計算への適用可能性

このような要請に応える手法として、Krylov部分空間法が注目されている。

shifted MINRES法

MINRES法は、実対称またはエルミートな係数行列に対して有効な Krylov 部分空間法であり、**Lanczos過程**に基づいて最小残差解を反復的に求める。これをシフト線形方程式に拡張したのが shifted MINRES法（sminres）である。アルゴリズムの概要：

- A に対して Lanczos過程を実行し、正規直交基底を構成

$$AV_n = V_{n+1}\hat{T}_n, \quad V_n = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_n], \quad T_n = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix}, \quad \hat{T}_n = \begin{bmatrix} T_n \\ \beta_n \mathbf{e}_n^T \end{bmatrix} \quad (2)$$

- $k = 1, 2, \dots, M$

- $\hat{T}_n^{(k)} = \hat{T}_n + \sigma^{(k)} \begin{bmatrix} I \\ \mathbf{0}^T \end{bmatrix}$ とおく（ $(A + \sigma^{(k)}I)V_n = V_{n+1}\hat{T}_n^{(k)}$ が成り立つ）

- $\hat{T}_n^{(k)}$ のQR分解を計算する（ $T_n^{(k)} = Q_n R_n$ ）

- $\mathbf{y}_n^{(k)} = \|\mathbf{b}\| R_n^{-1} Q_n^H \mathbf{e}_1$ を求める

$$\begin{aligned} \|\mathbf{r}_n^{(k)}\| &= \|\mathbf{b} - (A + \sigma_k I)\mathbf{x}_n\| = \left\| V_{n+1} \left(\|\mathbf{b}\| \mathbf{e}_1 - \hat{T}_n^{(k)} \mathbf{y}_n^{(k)} \right) \right\| \\ &= \left\| \|\mathbf{b}\| \mathbf{e}_1 - \hat{T}_n^{(k)} \mathbf{y}_n^{(k)} \right\| \end{aligned} \quad (3)$$

- 最小残差解 $\mathbf{x}_n^{(k)} = V_n \mathbf{y}_n^{(k)}$ を求める

特徴：

- 1回**のLanczos過程で **M 個**の方程式に必要な基底が得られる
- 残差のノルムの単調減少性**と**無破綻性**を持つ
- シフトごとに**独立に**最小残差解が求められる

Algorithm 1 shifted MINRES method

```
1:  $\mathbf{x}_0^{(\sigma)} = \mathbf{0}, \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ 
2:  $\beta_0 = 0, \mathbf{v}_0 = \mathbf{0}; \mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|_2$ 
3:  $f_1^{(\sigma)} = 1, h_0^{(\sigma)} = \|\mathbf{r}_0\|_2, \mathbf{p}_{-1}^{(\sigma)} = \mathbf{p}_0^{(\sigma)} = \mathbf{0}$ 
4: for  $j = 1, 2, \dots$  do
5:    $\mathbf{v}_{j+1}'' = A\mathbf{v}_j$ 
6:    $\alpha_j = \langle \mathbf{v}_j, \mathbf{v}_{j+1}'' \rangle$ 
7:    $\mathbf{v}_{j+1}' = \mathbf{v}_{j+1}'' - \alpha_j \mathbf{v}_j - \beta_{j-1} \mathbf{v}_{j-1}$ 
8:    $\beta_j = \|\mathbf{v}_{j+1}'\|_2$ 
9:    $\mathbf{v}_{j+1} = \mathbf{v}_{j+1}' / \beta_j$ 
10:  for  $k = 1, 2, \dots, M$  do ↩ 並列化可能
11:     $\hat{T}_{j-2,j}^{(k)} = 0$ 
12:     $\hat{T}_{j,j}^{(k)} = \beta_{j-1}, \hat{T}_{j,j}^{(k)} = \alpha_j + \sigma_k, \hat{T}_{j+1,j}^{(k)} = \beta_j$ 
13:    if  $j \geq 3$  then  $\begin{bmatrix} \hat{T}_{j,j-2}^{(k)} & \hat{T}_{j,j-1}^{(k)} \end{bmatrix}^T = G_{j-2}^{(k)} \begin{bmatrix} \hat{T}_{j,j-2}^{(k)} & \hat{T}_{j,j-1}^{(k)} \end{bmatrix}^T$  end if
14:    if  $j \geq 2$  then  $\begin{bmatrix} \hat{T}_{j,j-1}^{(k)} & \hat{T}_{j,j}^{(k)} \end{bmatrix}^T = G_{j-1}^{(k)} \begin{bmatrix} \hat{T}_{j,j-1}^{(k)} & \hat{T}_{j,j}^{(k)} \end{bmatrix}^T$  end if
15:     $\left( \hat{T}_{j,j}^{(k)}, G_j^{(k)} \right) = \text{GIVENS} \left( \hat{T}_{j,j}^{(k)}, \hat{T}_{j+1,j}^{(k)} \right)$ 
16:     $\mathbf{p}_j^{(k)} = \left( \mathbf{v}_j - \hat{T}_{j-2,j}^{(k)} \mathbf{p}_{j-2}^{(k)} - \hat{T}_{j-1,j}^{(k)} \mathbf{p}_{j-1}^{(k)} \right) / \hat{T}_{j,j}^{(k)}$ 
17:     $\mathbf{x}_j^{(k)} = \mathbf{x}_{j-1}^{(k)} + \|\mathbf{r}_0\| c_j^{(k)} f_j^{(k)} \mathbf{p}_j^{(k)}$ 
18:     $f_{j+1}^{(k)} = -\bar{s}_j f_j^{(k)}$ 
19:     $h_j^{(k)} = |-\bar{s}_j^{(k)}| \cdot h_{j-1}^{(k)}$ 
20:    Determine convergence by  $\|\mathbf{r}_j^{(k)}\| = h_j^{(k)}$ 
21:  end for
22: end for
```

shifted MINRES法の並列化モデル

sminresでは、すべてのシフトに対して共通の基底を用いて反復計算を行う。この構造は、シフトごとの処理が独立であることから、高い並列性を有する。本研究では、以下の2種類の並列化モデルを実装・比較した：

モデル1：ベクトル分割(MPI)＋シフト分割(OpenMP)

- MPI によるベクトルの**行方向分割**（ドメイン分割）
- OpenMP による各プロセス内でシフトループを並列化（ M 個の方程式）
- 行列ベクトル積、内積、ノルム演算で通信を必要とする

モデル2：シフト分割(MPI)＋シフト分割(openMP)

- MPI によりプロセスに対してシフトを割り当て、完全に**独立なシフト分散**
- OpenMP により担当シフト内でループを並列化
- 通信を必要としない（行列は各プロセスが全体を保持）

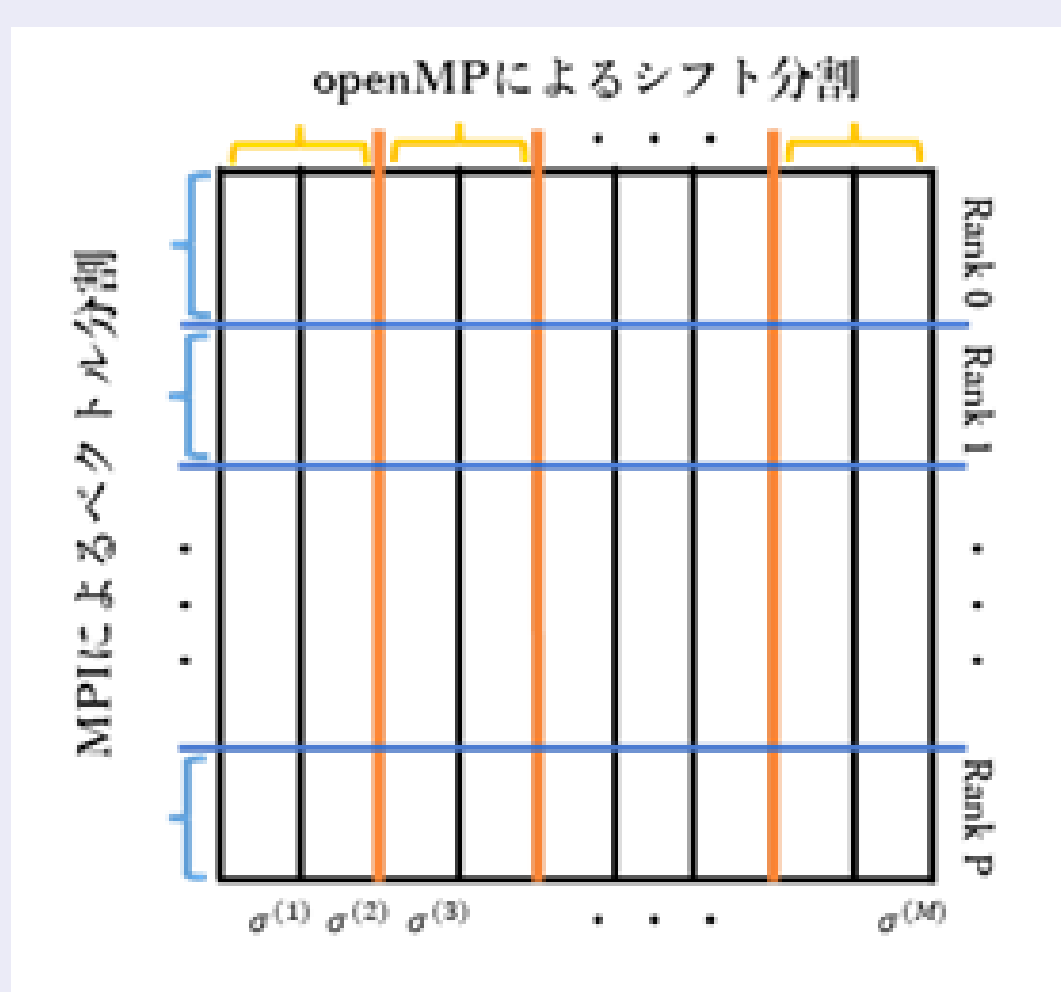


図 1: 並列化モデル 1

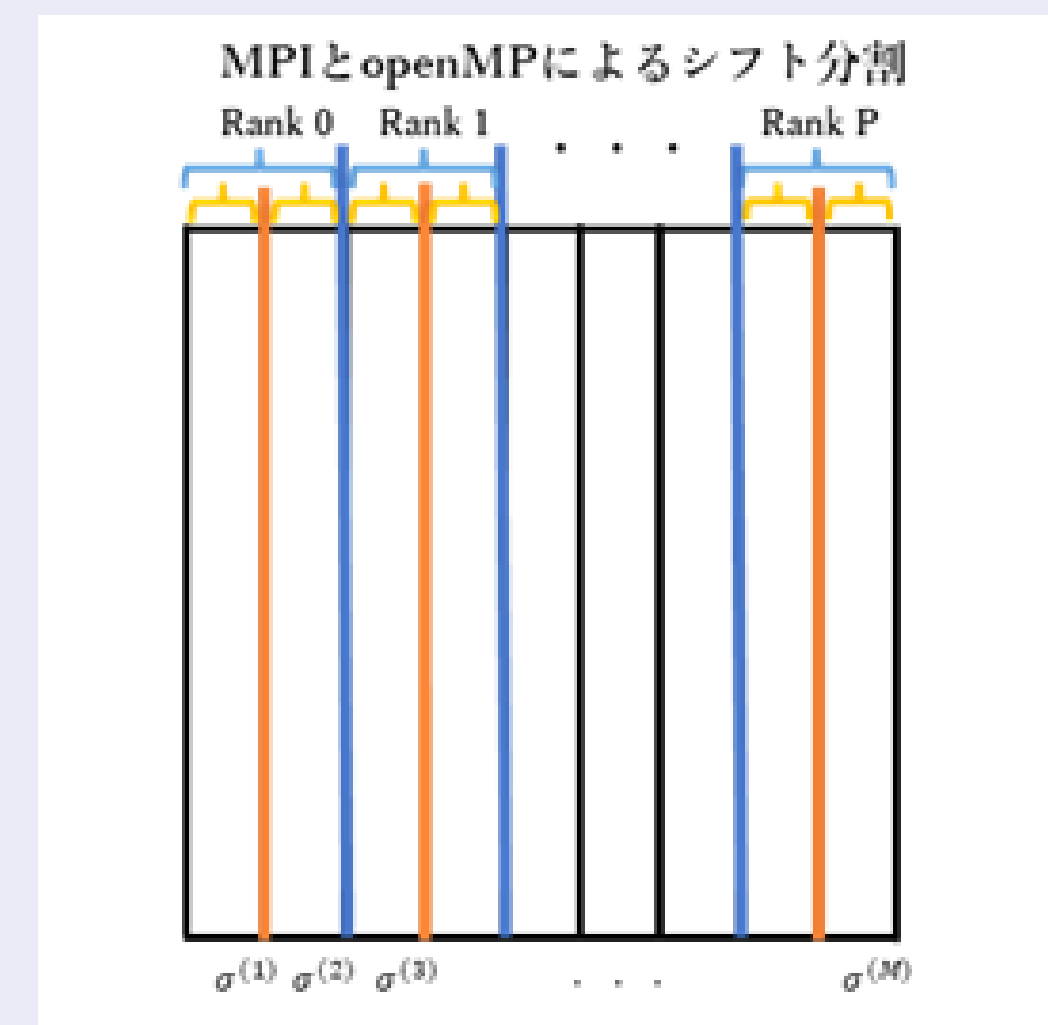


図 2: 並列化モデル 2

モデル1における通信の詳細：

行番号	MPI関数	目的	通信量
5	MPI Allgatherv	部分ベクトルを集約し、全体ベクトルを構成	$O(N)$
6	MPI Allreduce	局所内積の合計を求め、全プロセスに共有	$O(1)$
8	MPI Allreduce	局所平方和の合計を求め、全プロセスに共有	$O(1)$

数値実験

CED上で2種類の並列化モデルおよび逐次実装の実行時間（秒）を比較した。

- 問題設定

- Intel Core i7-12700 (12 cores), GNU C++ Compiler, Netlib BLAS/LAPACK
- VCNT4000std (4000次実対称), VCNT40000std (40000次実対称) [2]
- $\sigma^{(k)} = 0.01 \exp(2\pi(k - 0.5)/M)$, $M = 50$ （実験1）
- $\sigma^{(k)} = -0.1 + k * 0.2/M + 0.01i$, $M = 100$ （実験2）
- 収束条件: 相対残差 $\|\mathbf{r}_j\| / \|\mathbf{r}_0\| \leq 10^{-13}$
- MPI: 4並列, OpenMP: 指定なし(最大20スレッド)

- 実験1

VCNT4000std			VCNT40000std		
逐次	モデル1	モデル2	逐次	モデル1	モデル2
3.44726	13.5996	0.988548	676.268	156.539	912.709

- 実験2

VCNT4000std			VCNT40000std		
逐次	モデル1	モデル2	逐次	モデル1	モデル2
2.55095	3.44825	0.267222	160.371	32.1898	207.159

- 大規模行列では、通信の相対負荷が低下し、モデル1が最も高速だった
- 小規模行列では、通信のないモデル2が最も高速に動作した
- 超大規模行列では、モデル1がスケーラブルな解法として期待できる

まとめと今後の展望

- shifted MINRES法に対する2種類の並列化モデルの検討をおこなった
 - 大規模行列にはモデル1が、小規模行列にはモデル2が有効
- 大規模問題、実問題での性能検証
- その他の並列化モデル・最適化の検討

参考文献

- S. Hidaka, S. Kudo, T. Hoshi, Y. Yamamoto, Performance of the shifted minimal residual method for multiply shifted linear systems with real symmetric or complex Hermitian coefficient matrices, Comput. Phys. Comm., **314** (2025), 109679.
- T. Hoshi, ELSEs matrix library, 2019, <http://www.elses.jp/matrix/>. (accessed 22 Jun. 2025)
- S. Hidaka, 修士課程中間発表リポジトリ, <https://github.com/ShunHidaka/masters-interim-poster>.