

疎行列ガウス掃出し法の効率化について

田尻 太郎¹

¹システムデザイン開発株式会社 テクニカルシステム事業部

e-mail : taro_tajiri@sddgrp.co.jp, Tel : 011-281-6770

1 概要

疎行列の掃出法において Active 列の零要素を Skip する処理を前提とすれば (Fillin Skip と呼ぶ) 未処理ブロックの非零要素 (Fillin と呼ぶ) 数最小の列を Active 列と交換する事で処理時間の短縮を期待できる^[1]. しかし Fillin 最小の列の探索は完全 Pivot と同程度の CPU 負荷があるため, Fillin Skip のみの処理の 2 倍以上の時間を要する.

ここでは掃出 Step ごとに列交換を行わず, 数値結果に基づいた適切なタイミングで未処理ブロックの列を Fillin 数の昇順に並べ直す事によって掃出法を効率化する, 簡易な方法を検討した.

なおここでは, 掃出 Step ごとの列交換を m-Ordering, Fillin 数昇順の並べかえを f-Ordering と呼ぶ.

2 f-Ordering による前処理

数値計算用のサンプルとしてほぼ一様ランダムな Fillin 配置を持つ疎行列を使用した. 掃出法の前処理として f-Ordering を行った結果を図 1. に示す. 行列規模は 1000×1000 , 非零率は Fillin 数/全要素数, 横軸は掃出法の処理率 (Step 数/行寸法), 縦軸は各 Step の処理時間 (s) である. 同一非零率を持つ 5 個の行列の平均値に長さ 20Step の移動平均をかけ平滑化した. 前処理による結果は黒で, Fillin Skip のみ (赤) と m-Ordering (青) による結果を併記する. m-Ordering に列探索時間は含まれない. また A, B のラインは, Active 列非零率が 5% の時の処理時間を示す.

A, B ラインは前処理と Fillin Skip の 1

~3% のグラフでは変曲点を通り, 変曲点以降は処理に伴う Fillin 増加で急速に行列は密化する. さらに係数行列の非零率 5% で前処理効果は殆どない. 以上から, ほぼ一様ランダムな疎行列の密化は, 列非零率 5% の Active 列出現のタイミングで起こると判断した.

3 f-Ordering 実行の制御

f-Ordering の CPU 負荷は完全 Pivot の 1.5 程度のため毎回実行はできないが, 係数行列の非零率が小さい時は図 1. より 1 回の実行で, Fillin Skip のみの時よりかなり処理時間を短縮できる. そこで先の結果を考慮した, f-Ordering の実行制御を試みた.

1) f-Ordering の開始条件

非零率 5% を越える Active 列の出現を開始条件とした. ただし 20Step の移動平均に基づく結果なので, トリガー値は行列規模 n の 2% の長さ ($20/1000=0.02$) の平均とする. 平均値計算は行方向 Pivot 選択時に並行して行う.

2) f-Ordering の終了条件

係数行列の未処理ブロックに開始条件のトリガー値以下の非零率の未処理列が残っていなければ f-Ordering の効果は持続しないので, それを打ち切り条件とした.

結果を図 2., 図 3. に示す. 図中の f-Ordering には列並べかえ時間も含まれるが, 係数行列非零率 5% 以内では m-Ordering の掃出時間に近い結果を与える. また 1) 2) のコーディングは極めて容易であり, 任意の疎行列にアルゴリズム上は適用できる. 非一様な場合の処理時間短縮の事前評価は難しいが, 計算時間短縮の効果は期待できると考える.

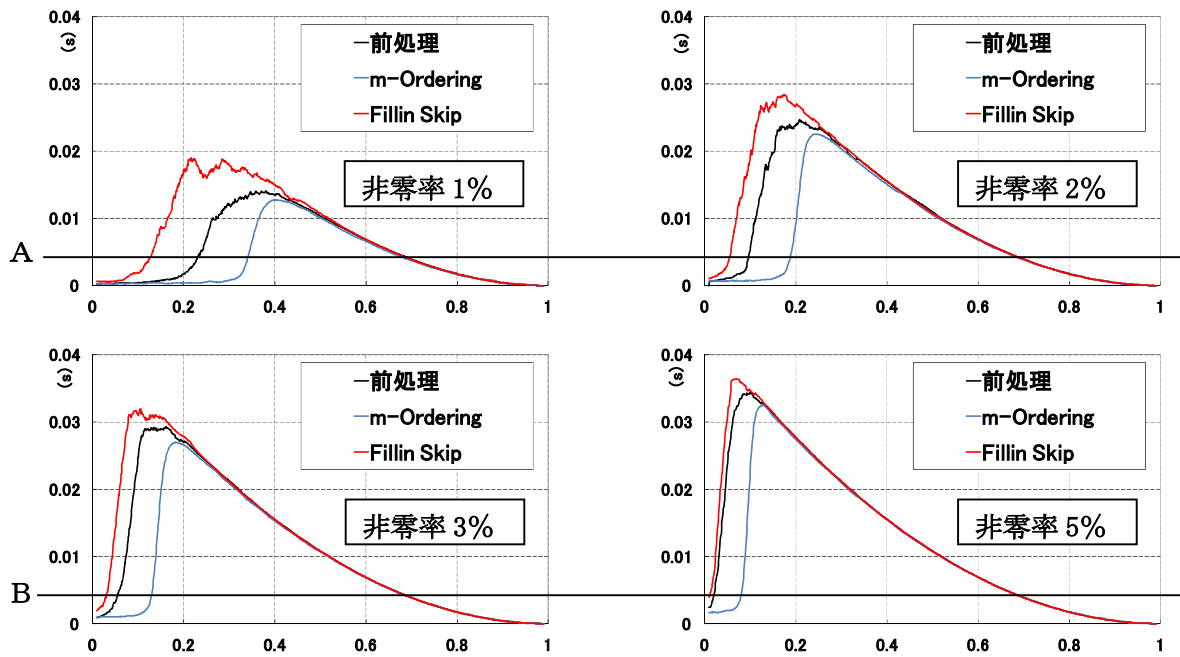


図 1. f-Ordering による前処理結果

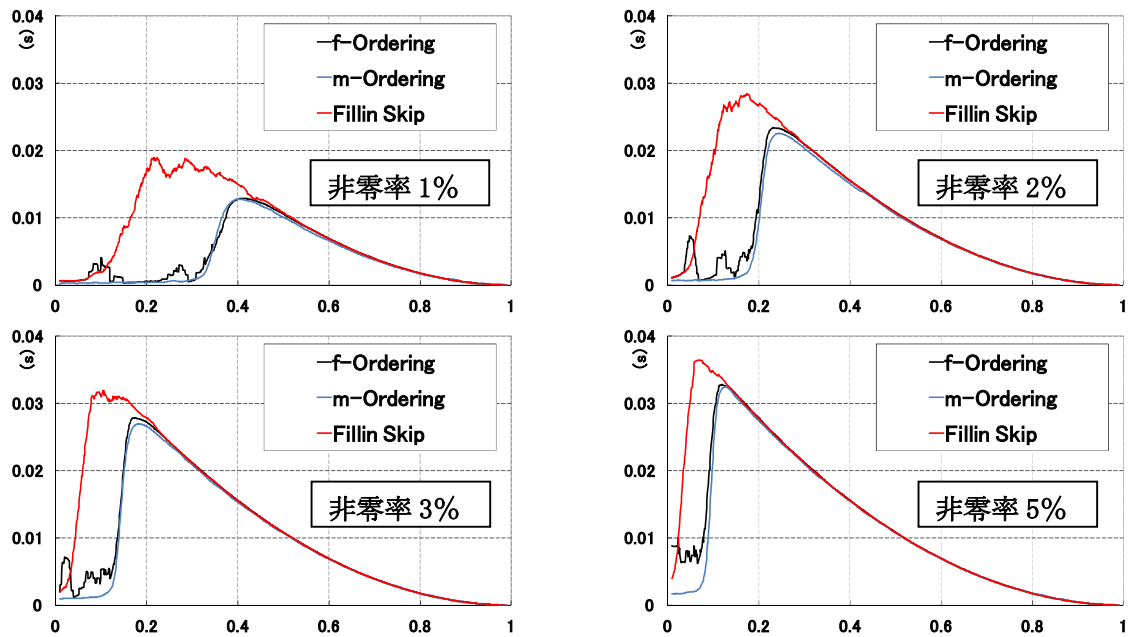


図 2. f-Ordering 実行制御による結果

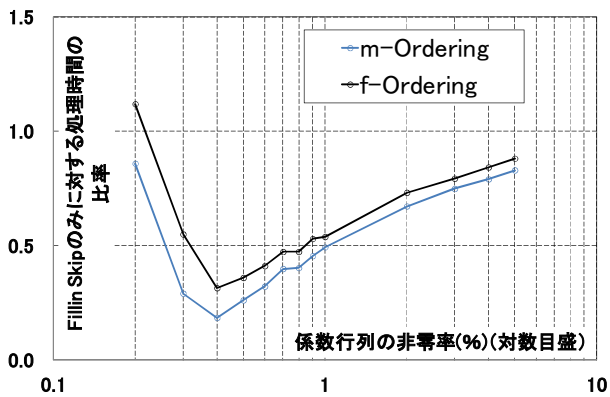


図 3. 各種方法による処理時間比較

参考文献

- [1] 小国 力, 村田健郎, 三好俊郎, ドンガラ, 長谷川秀彦, 行列計算ソフトウェア (WS, スーパーコン, 並列計算機), 丸善株式会社, 1991 年.