

jh250068

# 大規模線形行列方程式に対する数値解法の高速化に関する研究

佐竹祐樹（北海道大学）

## 概要

線形行列方程式は、制御理論や画像処理、物理シミュレーションをはじめとした幅広い分野に現れる。特に、科学技術の発展に伴い、より大規模な問題の求解が求められていることから、大規模行列方程式に対する数値解法の実用的なアルゴリズムの構築、及びその高速化が重要な研究課題である。本研究では、数理及び HPC の両面からのアプローチにより、大規模線形行列方程式向け実用的高速数値解法の構築に取り組む。2025 年度では、昨年度に引き続き、大規模 Sylvester 方程式に対する Krylov 部分空間法及びシフト線形方程式に対する数値解法に関するアルゴリズムの開発・実装を行った。

## 1 共同研究に関する情報

### 1.1 共同研究を実施した拠点名

- 北海道大学 情報基盤センター
- 東京大学 情報基盤センター
- 京都大学 学術情報メディアセンター

### 1.2 課題分野

- 大規模計算科学課題分野

### 1.3 参加研究者の役割分担

- 佐竹祐樹（北海道大学）：課題代表、研究統括、行列方程式に対する数値解法の実装・研究開発
- 深谷猛（北海道大学）：課題副代表、統括補佐、並列計算に関する助言
- 岩下武史（京都大学）：Krylov 部分空間法、並列計算に関する助言
- 曾我部知広（名古屋大学）：シフト線形方程式、Krylov 部分空間法に関する助言

- 萬本遼太郎（名古屋大学）：シフト線形方程式に対する数値解法の並列化

## 2 研究の目的と意義

線形行列方程式は、制御理論や振動解析などの工学分野や、画像処理などの情報学分野に現れる。また、偏微分方程式の離散化においても現れることため、物理シミュレーションなどの幅広い分野においてその求解が重要である。特に、科学技術の発展とともに、各応用分野において、よりサイズの大きい方程式の求解が必要となっており、したがって、大規模な線形行列方程式に対する高速な数値解法の構築は重要な研究課題といえる。

上記のように幅広い応用があるにも関わらず、大規模線形行列方程式に対する数値解法に関する研究は、数値線形代数分野においてあまり注目されてこなかった。その要因の一つとし

て、線形行列方程式はある種の連立 1 次方程式とみなせる点が挙げられる。大規模連立 1 次方程式に対する数値解法に関する研究は、古くから現在に至るまで多くの研究者によって行われており、数値解法の数理に関する基盤的な研究から最新の計算機に合わせた実装に関する研究まで非常に多くの研究が存在する。したがって、連立 1 次方程式に対する実用的な既存数値解法を線形行列方程式に対して利用することが可能である。

その一方で、線形行列方程式のまま求解を行うことで、連立 1 次方程式とみなして求解を行う場合に比べて所要メモリの大幅な削減と計算の高速化が期待できることから、20 年ほど前から行列方程式に対する数値解法や関連する研究が盛んに行われるようになってきた。具体的には、解を密行列で扱う代わりに低ランク行列で表現することで、所要メモリの削減と高速化の実現を目指す手法である。現在も低ランクな近似解を計算する数値解法の研究及びその高速化に関する研究が行われているが、数理的な観点からのアプローチがほとんどであり、実際の計算機上での実装（並列計算等）を考慮した研究はほとんど行われていない。今後応用分野において、さらに大規模な問題の求解が必要になってくると考えられ、また、「ムーアの法則」が限界に近づきつつある現状を踏まえると、数理的視点に加えて HPC の観点からのアプローチによる数値解法の高速化が必要である。本研究では、数理及び HPC の両面の視点から大規模線形行列方程式に対する数値解法の高速化を行うことが目的である。

上述の通り、線形行列方程式を連立 1 次方程式とみなすのではなく、行列方程式の構造を用いて低ランクな近似解を計算する数値解法を用いることで所要メモリを大きく削減できることから、より大規模な問題への対応が可能とな

る。また、数理及び HPC の両面から計算の高速化を実現することで、真に実用的な数値解法の確立につながる。線形行列方程式は、応用分野に加え、非線形行列方程式の求解にも用いられるため、これらの理由から非常に幅広い分野への波及効果が期待できる。

### 3 当拠点公募型研究として実施した意義

低ランクな近似解を計算する数値解法は、行列サイズが大きいほど、よりメモリや計算量の観点での効率性が期待されるため、豊富な計算資源を有する計算機上で実装・評価するのが望ましい。また、先に述べたように、実用性の観点から、実際の計算機上での実装を考慮した数値解法の構築が重要である。特に、現実的な応用分野に現れる大規模問題を考える場合、豊富な計算資源を有する計算機システムを用いる必要がある。したがって、本研究の目的を達成する上で、JHPCN の各拠点の計算機環境での数値解法の開発を行うことは理に適っている。

また、JHPCN のシンポジウム等で多くの応用分野の研究者との交流が可能であり、本研究に対して応用分野の研究者からの知見が得られるとともに、応用分野の研究者に本研究の手法を認知・利用してもらうことで、将来的に幅広い分野への貢献が期待される。

### 4 前年度までに得られた研究成果の概要

前年度までに行った研究は以下のとおりである。

- 行列方程式の一種である Sylvester 方程式を対象として、大規模向け数値解法である低ランク型 Krylov 部分空間法 (truncated BiCGSTAB 法) の実装を行った。本手法

は、Krylov 部分空間法のアルゴリズムに低ランク近似を組み合わせた手法である。反復毎に低ランク近似を行うため、その切り捨て基準パラメータを変化させたときの収束性の変化と解のランク変化の様子を確認した。

- Sylvester 方程式の特殊な場合に相当するシフト線形方程式を対象として、数値解法の MPI と OpenMP を用いたハイブリッド並列化に取り組んだ。具体的には、近年提案された手法である shifted LOP-BiCG 法のハイブリッド並列化を行った。また、LOPBICG 法のアルゴリズムにおける計算順序の工夫により、通信と計算のオーバーラップを行い、通信隠蔽を図った。実装したハイブリッド並列版 LOP-BiCG 法を、東京大学情報基盤センターの Wisteria/BDEC-01 (Odyssey) を用いて性能評価を行った。得られた研究成果については、情報処理学会の HPC 研究会で発表している。
- 行列方程式に対する低ランク型 Krylov 部分空間法において用いられる低ランク近似では、縦長行列の QR 分解が頻出し、また、この計算がボトルネックとなることから、その高速化が重要である。これを踏まえて、分散並列環境を想定した縦長行列向け QR 分解のアルゴリズムの性能評価を行った。

## 5 今年度の研究成果の詳細

本年度に取り組んだ研究内容は以下のとおりである。

### 5.1 大規模 Sylvester 方程式に対する効率的な Krylov 部分空間法の提案・実装

前年度に取り組んだ Sylvester 方程式に対する低ランク型 Krylov 部分空間法の実装を踏まえて、当初その並列化に取り組む予定だったが、Sylvester 方程式の数理構造を活用して Krylov 部分空間法のアルゴリズムを再構築し、新たな手法を開発したことから、その実装および並列化に取り組んだ。具体的には、Krylov 部分空間法の一つである BiCGSTAB 法を対象として、アルゴリズムの再構築を行い、実装・性能評価した。提案手法は、既存手法に比べて既存手法では近似解の低ランク性を保つために反復毎に低ランク近似を必要とし、これが計算のボトルネックとなっていたが、提案手法では理論的な低ランク表現を利用することでこの計算が不要となっており、既存手法に比べて計算の高速化が期待できる。

北海道大学情報基盤センターのスーパーコンピュータシステム Grand Chariot 2 で、行列サイズが 100 万 × 100 万の Sylvester 方程式に対して提案手法のアルゴリズムを適用した収束履歴を図 1 に示す。行列サイズが 100 万 × 100

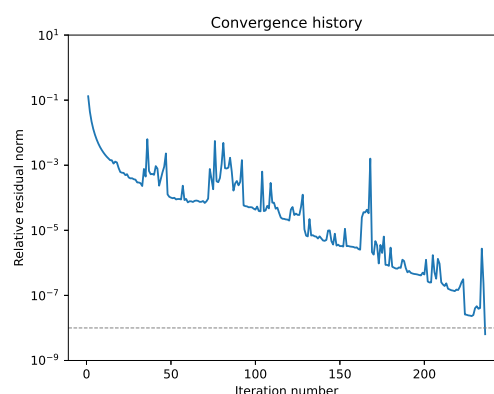


図 1: 提案アルゴリズム（大規模 Sylvester 方程式に対する BiCGSTAB 法）の相対残差履歴

万の場合、1 兆次元の連立 1 次方程式と等価で

あり、解行列の全成分を計算する場合に 1 ノード（メモリ容量: 512GiB）ではメモリ不足となるが、低ランク形式で計算することでこれを可能としている。同サイズの問題を既存のアルゴリズムを用いた場合、同様の収束履歴となることが予想されるものの、（比較的小規模な問題で試した結果から）現実的な計算時間で解が得られない可能性が高いと判断し、ここでは既存手法との比較は行わずに、提案手法のみ並列化を行った。

提案手法において、近似解は縦長行列・（小規模な）正方行列・横長行列の 3 つの行列の積として低ランク形式で表現される（図 2）。こ

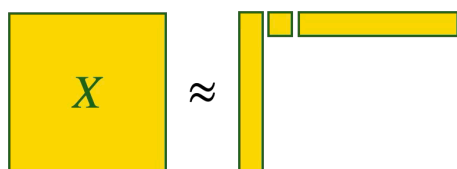


図 2: 低ランク形式

の縦長行列および横長行列を一次元データ分散することで提案手法の MPI 並列化（フラット MPI）を行い、Grand Chariot 2 で実装・性能評価を実施した。図 3 は、プロセス数を変化させた時の BiCGSTAB 法の計算時間の変化を示している。ただし、縦軸は逐次版（プロセス数=1）の計算時間で正規化して対数でプロットしている。

図から、プロセス数を増やした時に 16 プロセスまでは綺麗にスケールしているが、32 以上では速度向上率が下がり、512 から 1024 にすると通信時間の増加により遅くなることを確認した。右辺行列（のランク）を変えて同様に実験を行ったが、ほぼ同様の結果が得られた。

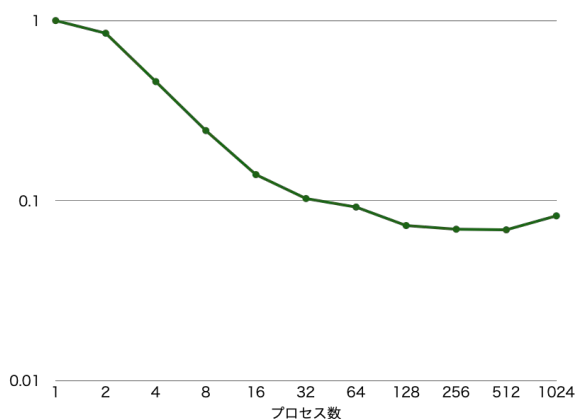


図 3: 大規模 Sylvester 方程式に対する BiCGSTAB 法（フラット MPI）のプロセス数に対する計算時間の変化

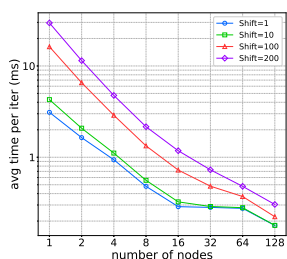
## 5.2 シフト線形方程式に対する shifted LOP-BiCG 法の実装

本年度は、昨年度に実施したシフト線形方程式

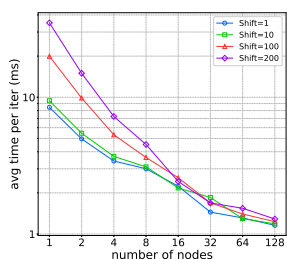
$$(A + \sigma_i I)x^{(i)} = b, \quad i = 1, \dots, m \quad (1)$$

に対する shifted LOPBiCG 法の実装の改善に取り組んだ。具体的には、行列ベクトル積に関して、昨年度は汎用性を考慮して（行列の疎構造に依存しない）集団通信（Allgather）を用いて実装していたが、通信量削減の観点から 1 対 1 通信を用いた実装を行った。

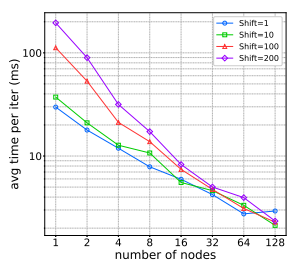
東京大学情報基盤センターの Wisteria / BDEC-01 (Odyssey) を用いて分散並列版 shifted LOPBiCG 法の実装・性能評価を行った結果を図 4 に示す。5 つのグラフはそれぞれ SuiteSparse Matrix Collection から取得した 5 種類の行列（atmosmodl、cage14、cage15、ss、Transport）に対する結果を示している。また、各行列に対して方程式の数をそれぞれ  $m = 1, 10, 100, 200$  の 4 通り行い、ノード数を増やした際の 1 反復あたりの計算時間の平均をそれぞれプロットしている。



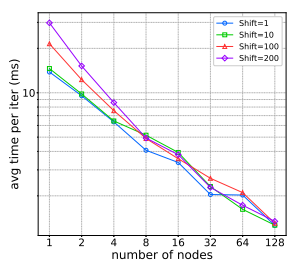
(a) atmosmodl



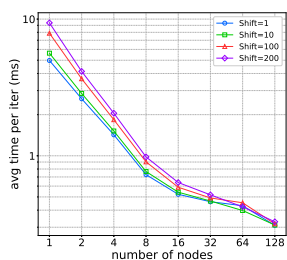
(b) cage14



(c) cage15



(d) ss



(e) Transport

結果から、各テスト行列に対して、ほぼ全ての場所でノード数を(128ノードまでの範囲で)増やした際に高速化できることを確認した。また、Transport などでは、ノード数を増やした時に方程式の数が多い場合 ( $m = 200$ ) と少ない場合 ( $m = 1$ ) の1反復あたりの計算時間がほとんど変わらないことが確認できた。

## 6 進捗状況の自己評価と今後の展望

当初の計画と変更点があったことで、全体的に進捗が遅くなり、計算資源の有効活用が十分でなかったことは反省すべき点である。ただし、中でも、より効率的なアルゴリズムの開発やアルゴリズムの改善ができたことは評価できるのではないかと考えている。

大規模 Sylvester 方程式に対する低ランク Krylov 部分空間法に関しては、当初の予定では、昨年度実施した既存手法の実装の並列化に取り組むつもりであったが、(理論的に)より効率的なアルゴリズムを再構築することができたため、その実装・並列化に取り組んだ。当初の計画とは異なる方向性の内容となったものの、既存手法のボトルネックであった計算を回避し、より実用的なアルゴリズムの開発に成功したことから、有意義であったと考えている。ただし、並列化は現状フラット MPI のみであることと並列化においてアルゴリズム改善の余地があることから、成果としてはまだ足りない部分が多く残ってしまったのは反省すべき点である。これについては、翌年度以降に引き続き取り組んでいく予定である。

シフト方程式に対する LOPBiCG 法の並列化については、昨年度の実装に比べてより実用的な実装を行うことができたと考えている。特に、昨年度の実装ではノード数を128まで増やした際に計算時間が遅くなっていたのに対して、本年度の実装では128ノードでもほぼ高速

図 4: ノード数を変化させた時の LOPBiCG 法の1反復あたりの計算時間

化ができているという点で並列性能の改善ができたと言える。今後は、異なる計算環境での実装・性能評価と実問題に適用した際の有効性の検証に取り組んでいく必要がある。