

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 (代表 / 北大), 深谷 猛 (副代表 / 北大), 岩下 武史 (京大), 曾我部 知広 (名大), 萬本 遼太郎 (名大 / 大学院生)

研究背景・目的

背景

- 線形行列方程式は、信号処理、制御理論、振動解析、偏微分方程式の離散化などの多くの分野に現れるため、その求解が重要である。
- 連立1次方程式とみなすことができるため、(十分な計算資源があれば)多くの確立された手法を利用できる。

例. Sylvester方程式

$$AX + XB = C$$

◦ 既知: $A, B, C \in \mathbb{R}^{n \times n}$ ◦ 未知: $X \in \mathbb{R}^{n \times n}$ \Leftrightarrow

$$\mathcal{F}x = c$$

◦ 既知: $\mathcal{F} \in \mathbb{R}^{n^2 \times n^2}, c \in \mathbb{R}^{n^2}$ ◦ 未知: $x \in \mathbb{R}^{n^2}$

- 係数行列が疎行列でも、一般に解行列は密行列となるため、行列サイズが大きいためにメモリの問題がある。
- 近年、近似解行列を低ランク行列として保持する省メモリな数値解法の研究が行われてきた。
- 並列化など、実際の計算機上での実装を考慮した研究はほとんど行われていない。

目的

数理・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」を行うことで、実用的解法の確立を目指す。

手法 1: 低ランク型Krylov部分空間法

行列型のKrylov部分空間法に低ランク近似を組み合わせた手法

概要

- 反復毎に低ランク行列で近似解を更新 (下図: CG法の例)

$$\begin{bmatrix} X_{k+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} X_k \\ \vdots \end{bmatrix} + \alpha_k \begin{bmatrix} P_k \\ \vdots \end{bmatrix}$$

- 小さい特異値を打ち切って圧縮

$$\begin{bmatrix} X_{k+1} \\ \vdots \end{bmatrix} \xrightarrow{\text{QR分解 + SVD}} \begin{bmatrix} X_{k+1} \\ \vdots \end{bmatrix}$$

課題 1: 計算量の削減

QR分解の計算量削減

前の反復の情報を利用することで計算量の削減を行う。

可変打ち切りパラメータの導入

特異値の打ち切りパラメータを反復毎に変化させることでランクの増加を抑制し、計算量を削減する。

課題 2: 前処理の開発

テンソル構造保存型前処理

行列方程式のもつテンソル構造を保存させることで、行列方程式に特化した前処理の開発を行う。

課題 3: 並列化

各計算の並列化

疎行列-ベクトル積, QR分解, 低ランク行列の内積

テンソル構造保存型前処理の並列化

開発した行列方程式向けの前処理の並列化を行い、その有効性を検証する。

手法 2: 射影法

低次元の部分空間に射影して小規模な問題に落とし込む手法

概要

- 適当な部分空間を用いて、正規直交基底を生成
- 直交条件を課して、小規模な行列方程式に帰着させて求解
- 近似解行列を低ランク行列で保持

$$X_k = \begin{bmatrix} v_k & Y_k & w_k^T \\ \vdots & \vdots & \vdots \end{bmatrix} \quad \begin{array}{l} \bullet v_k, w_k: \text{①で生成した列直交行列} \\ \bullet Y_k: \text{②で求めた小規模方程式の解} \end{array}$$

- 圧縮・リスタートの適用

課題 4: Sylvester方程式に対する射影法の高速化

QR分解アルゴリズムの高速化

縦長行列向けのQR分解アルゴリズムを用いて高速化を行う。

各計算の並列化

正規直交基底の生成, 小規模行列方程式の求解アルゴリズム, QR分解

課題 5: T-Sylvester方程式に対する射影法の構築

Lyapunov方程式への変換理論を用いた射影法

Lyapunov方程式への変換理論 ([Oozawa et al., 2018], [S. et al., 2019]) を用いた射影法を構築し、既存手法 [Dopico et al., 2016] との比較を行う。

圧縮・リスタートの適用

所要メモリの増大を防ぐため、提案手法に対して、圧縮・リスタートを適用する。

並列化

提案手法をHPC環境向けに並列化を行い、性能評価を行う。