Preconditioner for low rank GMRES

Yingda Cheng

Virginia Tech

CHaRMNET conf, Dec 2024

Joint work with Shixu Meng, Daniel Appelö

Outline

1. Introduction

2. A new preconditioner for low rank GMRES

Outline

1. Introduction
2. A new preconditioner for low rank GMRES

Overview of low rank methods for time-dependent problems

Compute low rank solution of matrix differential equations

$$\frac{d}{dt}X(t) = F(X(t), t), \quad X(t) \in \mathbb{R}^{m_1 \times m_2}, \quad X(0) = X_0, \tag{1}$$

Solvers can be cast into three groups:

- Step truncation (ST) method. CHaRMNET work: Qiu, Taitano, Chacon, Hamad.
- Dynamic low rank approximation (DLRA). CHaRMNET work: Hu, Haut, Hauck, Schotthöfer.
- Space-time formalism.

Step truncation method

Idea: evolve the low rank solution for one time step by a traditional time stepping method in an ambient space of higher rank, then performs a truncation (by SVD with given tolerance).

```
Algorithm 1: Forward Euler scheme $t^n \to t^{n+1}$
Input     : numerical solution at $t^n$ : rank $r_n$ matrix $\hat{X}^n$ in its SVD
            form $U^n\Sigma^n(V^n)^T$.
Output    : numerical solution at $t^{n+1}$ : rank $r_{n+1}$ matrix $\hat{X}^{n+1}$ in its
            SVD form $U^{n+1}\Sigma^{n+1}(V^{n+1})^T$.
Parameter: time step $\Delta t$, error tolerance $\epsilon_1, \epsilon_2$
(Evolution). $\hat{X}^{n+1,pre} = \hat{X}^n + \Delta t \mathcal{T}^{sum}_{\epsilon_1}(F(\hat{X}^n, t^n))$.
(Truncation). $\hat{X}^{n+1} = \mathcal{T}^{sum}_{\epsilon_2}(\hat{X}^{n+1,pre})$.
```

Implicit schemes are hard to implement.

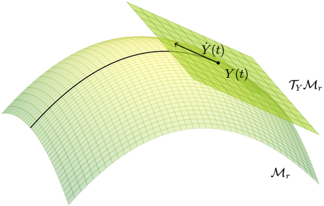
DLRA

DLRA constrains the solution on the rank r matrix manifold $\mathcal{M}_r = \{X \in \mathbb{R}^{m_1 \times m_2}, \mathrm{rank}(X) = r\}$ at $X(t)$.
It solves

$$\frac{d}{dt}X(t) = \Pi_{X(t)}F(X(t), t),$$

where $\Pi_{X(t)}$ is the orthogonal projection onto the tangent space $T_{X(t)}\mathcal{M}_r$.

Figure: From https://www.waves.kit.edu/B9.php



BUG integrator: $K-, L-, S-$ steps, smaller subsystem to solve than the original.

An example of failure of convergence of DLRA

For example, when $X(t)$ is a rank 1 even function in both variables (say $X(t, x_1, x_2) = \exp(-x_1^2)\exp(-x_2^2)$) on a square domain with center $(0, 0)$, and $F(X(t), t) = AXB^T$, with $A = \mathrm{diag}(x_1)$ and $B = \mathrm{diag}(x_2)$.
$\Pi_{X(t)} F(X(t), t) = 0$.

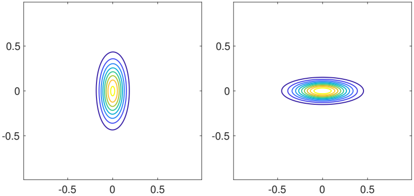


Figure: Left: backward Euler, Right: BUG. Solid body rotation with diffusion

Error estimate

$$C\Delta t + \epsilon + \|(I - \Pi_{X(t)})F(X, t)\|$$

Outline

1. Introduction
2. A new preconditioner for low rank GMRES

Goal

- In practice, DLRA has been applied to many applications, showing satisfactory performance.
- Our goal is to develop an implicit step truncation scheme that combines strength of ST and DLRA.

low rank GMRES

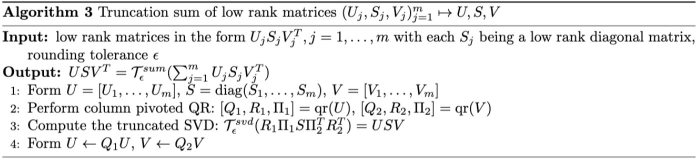
For linear diffusion problem, with an implicit time scheme, we need to solve for

$$C_1XD_1^T + C_2XD_2^T + \cdots C_kXD_k^T := \mathcal{A}X = b, \tag{2}$$

where $\mathcal{A} : \mathbb{R}^{m_1 \times m_2} \mapsto \mathbb{R}^{m_1 \times m_2}$ is the associated linear operator.

Low rank GMRES (lrGMRES)

- is based on GMRES.
- and the iterates are stored in SVD form $X = USV^T$.
- Rank truncation is needed to bound the rank.



Algorithm 3 Truncation sum of low rank matrices $(U_j, S_j, V_j)_{j=1}^m \mapsto U, S, V$

```
Input: low rank matrices in the form $U_jS_jV_j^T, j = 1, \ldots, m$ with each $S_j$ being a low rank diagonal matrix,
    rounding tolerance $\epsilon$
Output: $USV^T = \mathcal{T}_\epsilon^{sum}(\sum_{j=1}^m U_jS_jV_j^T)$
Form $U = [U_1, \ldots, U_m]$, $S = \text{diag}(S_1, \ldots, S_m)$, $V = [V_1, \ldots, V_m]$
Perform column pivoted QR: $[Q_1, R_1, \Pi_1] = \text{qr}(U)$, $[Q_2, R_2, \Pi_2] = \text{qr}(V)$
Compute the truncated SVD: $\mathcal{T}_\epsilon^{svd}(R_1\Pi_1S\Pi_2^TR_2^T) = USV$
Form $U \leftarrow Q_1U$, $V \leftarrow Q_2V$
```

Algorithm 2 lrGMRES

```
Input: linear map $\mathcal{A}$, $b$ in SVD form $b = U_b S_b V_b^T$, initial guess in SVD form $x_0 = U_{x0} S_{x0} V_{x0}^T$, rounding tolerance $\epsilon$, GMRES tolerance $\delta$, maximal number of iterations $m$
Output: $[U_x, S_x, V_x, \text{hasConverged}] = \text{lrGMRES}(\mathcal{A}, U_{x0}, S_{x0}, V_{x0}, U_b, S_b, V_b, \epsilon, \delta, m)$
$[U_{r0}, S_{r0}, V_{r0}] = \mathcal{T}_\epsilon^{sum}\left(b - \mathcal{A}(U_{x0} S_{x0} V_{x0}^T)\right)$, $\beta = \|S_{r0}\|$, $U_1 = U_{r0}$, $S_1 = S_{r0}/\beta$, $V_1 = V_{r0}$.
for $k = 1, \ldots, m$ do
    $[U_w, S_w, V_w] = \mathcal{T}_\epsilon^{sum}\left(\mathcal{A}(U_k S_k V_k^T)\right)$
    for $i = 1, \ldots, k$ do
        $\overline{H}_{i,k} = \langle U_i S_i V_i^T, U_w S_w V_w^T \rangle$
        $[U_w, S_w, V_w] = \mathcal{T}_\epsilon^{sum}\left(U_w S_w V_w^T - \overline{H}_{i,k} U_i S_i V_i^T\right)$
    end for
    $[U_w, S_w, V_w] = \mathcal{T}_\epsilon^{sum}\left(U_w S_w V_w^T\right)$
    $\overline{H}_{k+1,k} = \|U_w S_w V_w^T\|$
    $U_{k+1} = U_w$, $S_{k+1} = S_w / \overline{H}_{k+1,k}$, $V_{k+1} = V_w$
    $y_k = \text{argmin}_{y \in \mathbb{R}^k} \|\beta e_1 - \bar{H}_k y\|$
    $[U_x, S_x, V_x] = \mathcal{T}_\epsilon^{sum}\left(U_{x0} S_{x0} V_{x0}^T + \sum_{j=1}^k y_k(j) U_j S_j V_j^T\right)$
    if $\eta_{\mathcal{A},b}(U_x S_x V_x^T) \le \delta$ then
        hasConverged = True
        Break
    end if
end for
```

low rank Krylov

Low rank Krylov methods have been developed for matrix/tensor.

- Ballani, Grasedyck (2013). Dolgov (2013). Kressner, Sirkovic (2015). Kressner, Tobler 2010, 2011. Coulaud, Giraud, Iannacito (2022). Simoncini, Hao (2023).
- Low rank truncation results in loss of orthogonality in the Krylov subspace which makes analysis of convergence difficult.
- The critical aspect is a good preconditioner, which will control both the iteration number and intermediate rank.
- Preconditioner has to be *friendly* to low rank, i.e. only operating on the low rank factors. Candidate: multigrid, exponential sum.

BUG preconditioner

Due to the good performance of DLRA/BUG in most cases, we propose to use it as a preconditioner for lrGMRES.

Algorithm 8 BUG preconditioner for $\mathcal{A}X = b$

```
Input: linear map $\mathcal{A}$, the right hand side $b$, initial guess in SVD form $USV^T$
Output: $[U_{\text{new}}, S_{\text{new}}, V_{\text{new}}] = \mathcal{M}_{\text{BUG}}^{\mathcal{A},U,S,V}(b)$
K-step: set $K_0 = bV$, solve
    $$\mathcal{A}(K_1 V^T)V = K_0$$
    to obtain $[U, \sim, \sim] = \mathsf{qr}(K_1)$.
L-step: set $L_0 = b^T U$, solve
    $$\mathcal{A}^T(L_1 U^T)U = L_0$$
    to obtain $[V, \sim, \sim] = \mathsf{qr}(L_1)$. Here $\mathcal{A}^T(L_1 U^T) := (\mathcal{A}(U L_1^T))^T)$.
Galerkin: set $\Sigma_0 = U^T bV$, solve
    $$U^T \mathcal{A}(U \Sigma_1 V^T)V = \Sigma_0$$
    and obtain $[U_c, S_c, V_c] = \text{svd}(\Sigma_1)$.
Obtain $U_{\text{new}} = UU_c$, $V_{\text{new}} = VV_c$, $S_{\text{new}} = S_c$.
```

BUG preconditioner

```
Algorithm 9 Implicit Midpoint Method with lrGMRES with BUG preconditioner
Input: A set of second-order finite difference matrices ${A_j, B_j}_{j=1}^s$, time discretization parameters $\Delta t$, $n_t$ and $\theta$, initial condition in SVD form $X_0 = U_0 S_0 V_0^T$, source $G(t)$, rounding tolerance for lrGMRES $\epsilon$, truncation tolerance $\epsilon_2$, lrGMRES stopping criteria $\delta$, maximal number of iterations $m$, restart parameter maxit, preconditioner $\mathcal{M}$
Output: Solution at final time $[U, S, V]$
Set the linear map $\mathcal{A}$ by $\mathcal{A}X = X - \Delta t\theta \sum_{j=1}^s A_j X B_j^T$.
Initialization: $U = U_0$, $S = S_0$, $V = V_0$
for $n = 1, \ldots, n_t$ do
    $t = (n-1)\Delta t$
    $X \leftarrow USV^T$, $G_\theta \leftarrow G(\cdot,\cdot,(1-\theta)t + \theta(t+\Delta t))$, $U_b S_b V_b^T \leftarrow \Delta t(1-\theta)\sum_{j=1}^s A_j X B_j^T + G_\theta^n + X$
    $\mathcal{M} \leftarrow \mathcal{M}_{\text{BUG}}^{\mathcal{A},U,S,V}$
    $[U, S, V, \text{hasConverged}] \leftarrow \text{rplrGMRES}(\mathcal{A}, \mathcal{M}, U, S, V, U_b, S_b, V_b, \epsilon, \delta, m, \text{maxit})$
    $[U, S, V] \leftarrow \mathcal{T}_{\epsilon_2}^{sum}(USV^T)$
end for
```

BUG preconditioner - choice of space

- For the implicit midpoint schemes, $USV^T = X^n$ is the numerical solution at time step n.
- For the BDF scheme, $USV^T = \sum_{j=0}^{l} a_j X^{n-j}$ is a linear combination of computed solutions at previous steps.
- For the DIRK scheme, at the $j-$th inner stage, we take $USV^T = X^{n-1,(j)}$ which is the numerical solution at the $j-$th inner stage of the previous time steps.

BUG preconditioner - parameter study

We study the details of the following parameters: restart parameter, rounding tolerance ϵ and stopping criteria.

- Frequent restart benefits convergence.
- Truncation tolerance should be chosen according to local truncation error of the underlying scheme (next page).
- Stopping:

$$\eta_{\mathcal{A},b}(x_k) = \frac{\|\mathcal{A}x_k - b\|}{\|\mathcal{A}\|_2\|x_k\| + \|b\|} \leq \delta = \epsilon,$$

BUG preconditioner - truncation tolerance

Example (implicit midpoint+second order FD). $\Delta t = h$, $LTE = h^3$. For diffusion operator $h\epsilon(1 + \Delta t/h^2) = O(h^3)$, we obtain $\epsilon = O(h^3)$

Theorem (Convergence)

Suppose the matrix differential equation (1) *satisfies one-sided Lipschitz condition*

$$\langle F(X,t) - F(Y,t), X - Y\rangle \leq \alpha \|X - Y\|^2.$$

The implicit midpoint method with low rank GMRES scheme, if iteration terminates, applied to a linear diffusion type problem with the property $\|\mathcal{A}\|_2 \leq \frac{C\Delta t}{h^2}$, *and mesh size* $\Delta t = O(h)$, *tolerance* $\epsilon = O(h^3), \epsilon_2 = O(h^2)$ *is convergent of second order.*

BUG preconditioner - truncation tolerance

Example (implicit midpoint+second order FD). $\Delta t = h$, $LTE = h^3$. For diffusion operator $h\epsilon(1 + \Delta t/h^2) = O(h^3)$, we obtain $\epsilon = O(h^3)$

Theorem (Convergence)

Suppose the matrix differential equation (1) *satisfies one-sided Lipschitz condition*

$$\langle F(X,t) - F(Y,t), X - Y \rangle \leq \alpha \|X - Y\|^2.$$

The implicit midpoint method with low rank GMRES scheme, if iteration terminates, applied to a linear diffusion type problem with the property $\|\mathcal{A}\|_2 \leq \frac{C\Delta t}{h^2}$, *and mesh size* $\Delta t = O(h)$, *tolerance* $\epsilon = O(h^3), \epsilon_2 = O(h^2)$ *is convergent of second order.*

Further, we propose a hybrid preconditioner which alternates between BUG and ES preconditioner.

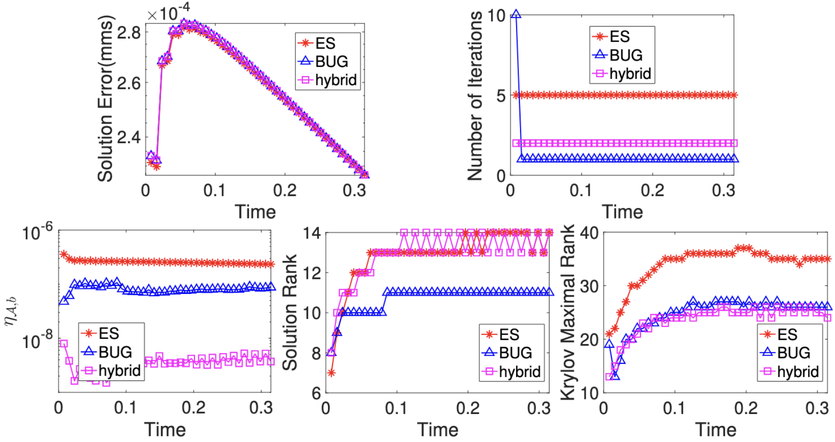
Numerical results

$$\begin{aligned} \frac{\partial X}{\partial t} &= b_1(y)\frac{\partial}{\partial x}\left(a_1(x)\frac{\partial X}{\partial x}\right) + b_2(y)\frac{\partial^2(a_2(x)X)}{\partial x \partial y} \\ &+ a_3(x)\frac{\partial^2(b_3(y)X)}{\partial x \partial y} + a_4(x)\frac{\partial}{\partial y}\left(b_4(y)\frac{\partial X}{\partial y}\right) + G(x,y,t) \quad (3) \end{aligned}$$

Example

$$\begin{aligned} a_1(x) &= 1 = b_1(y), \quad a_2(x) = 0.8 = b_3(y), \\ b_2(y) &= 1 = a_3(x), \quad a_4(x) = 1 = b_4(y), \quad (4) \end{aligned}$$

Numerical results



Numerical results

$$\begin{aligned} a_1(x) &= 1, \qquad b_1(y) = 1 + 0.1\sin(\pi y), \\ a_2(x) &= 1, \qquad b_2(y) = 1/\eta(1 + 0.1\sin(\pi y)), \\ a_3(x) &= 1, \qquad b_3(y) = 1/\eta(1 + 0.1\sin(\pi y)), \\ a_4(x) &= 1, \qquad b_4(y) = 1/\eta^2(1 + 0.1\sin(\pi y)), \end{aligned} \tag{5}$$

with $\eta = 1/10$.

Numerical results

	error	order	error	order	error	order	error	order
h	$\epsilon = h^3$(ES)	–	$\epsilon = \eta h^3$(ES)	–	$\epsilon = \eta^2 h^3$(ES)	–	$\epsilon = h^3$(BUG)	–
3.12(-2)	1.42(-2)	–	1.17(-3)	–	9.67(-4)	–	9.13(-4)	–
1.56(-2)	4.77(-3)	1.57	3.10(-4)	1.92	2.35(-4)	2.03	2.40(-4)	1.92
7.81(-3)	4.71(-3)	0.017	9.71(-5)	1.67	6.06(-5)	1.95	6.03(-5)	1.99

Table 3: Example 5.6. Solving diffusion equation with high contrast variable coefficients (16) and manufactured solution (17). Preconditioner: ES and BUG. For $h = h_x = h_y \in \{3.12(-2), 1.56(-2), 7.81(-3)\}$, this table displays the solution error at the final time for different rounding tolerances and order of convergence.

Numerical results -BDF4

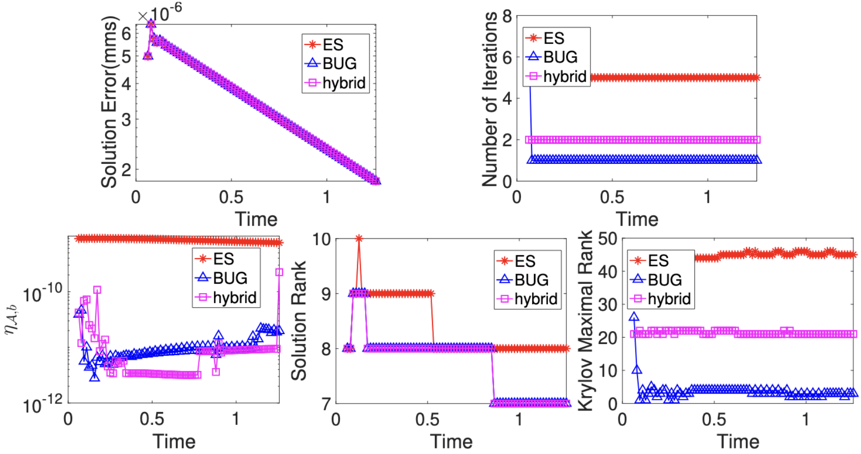


Figure 8: Example 5.7. Solving diffusion equation with variable coefficient (18) and manufactured solution (19). BDF4 and forth order finite difference in space. Preconditioner: ES, BUG, and hybrid. Fourth-order scheme with BDF. Rounding tolerance $\epsilon = h^5$. For $h = h_x = h_y = 1.56(-2)$, this figure displays the history of solution error, iteration number, $\eta_{\mathcal{A},b}$, solution rank, and maximal Krylov rank.

Conclusions and outlook

- We developed low rank methods for discretizing stiff equations.
- Future work: nonlinear problem, multiscale problem.
- Some other ongoing work: reduced order model (ROM) for kinetic equations. postprocessing for ROM. low rank for quantum mechanics.

The END! Thank You!