Low Tensor-Train Rank Methods to Solve Sylvester Tensor Equations

Tianyi Shi
Cornell University

Alex Townsend
Max Ruth
Sylvester matrix equation

\[ AX + XB^T = F \]

\[ A \in \mathbb{C}^{n_1 \times n_1}, \quad B \in \mathbb{C}^{n_2 \times n_2}, \quad F \in \mathbb{C}^{n_1 \times n_2} \]

3D Sylvester tensor equation

\[ \mathcal{X} \times_1 A + \mathcal{X} \times_2 B + \mathcal{X} \times_3 C = \mathcal{F} \]

\[ A \in \mathbb{C}^{n_1 \times n_1}, \quad B \in \mathbb{C}^{n_2 \times n_2}, \quad C \in \mathbb{C}^{n_3 \times n_3}, \quad \mathcal{F} \in \mathbb{C}^{n_1 \times n_2 \times n_3} \]

k-mode product for a tensor \( \mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d} \) and a matrix \( A \in \mathbb{C}^{n_k \times n_k} \)

\[ (\mathcal{X} \times_k A)_{i_1,...,i_{k-1},j,i_{k+1},...,i_d} = \sum_{i_k=1}^{n_k} \mathcal{X}_{i_1,...,i_{d-1},i_k} A_{j,i_k} \]

Example: Poisson w/ FD

\[ -(u_{xx} + u_{yy} + u_{zz}) = f \quad \text{on} \ \Omega = [-1,1]^3 \]

\[ u |_{\partial \Omega} = 0 \]

\[ \mathcal{X} \times_1 K + \mathcal{X} \times_2 K + \mathcal{X} \times_3 K = \mathcal{F} \]

\[ K = -\frac{1}{h^2} \begin{bmatrix} 2 & -1 \\ -1 & \ddots & \ddots \\ \ddots & \ddots & -1 \\ -1 & 2 \end{bmatrix} \]
Alternating Direction Implicit (ADI) method

“Iterative” method for Sylvester matrix equation $AX - XB^T = F$

1. Select shift parameters $p$ and $q$ each of length $\ell$ based on spectra of $A$ and $B$.

2. For $1 \leq j \leq \ell$
   - Solve $(A - q_jI)X = F + X(B - q_jI)^T$.
   - Solve $X(B - p_jI)^T = (A - p_jI)X - F$. [Wachspress, 2008]
Factored ADI (fADI) [Benner, Li & Truhar, 2009]

“Iterative” method for Sylvester matrix equation $AX - XB^T = UV^T$

$X = ZDY^T$

rank $r$

1. Select shift parameters $p$ and $q$ each of length $\ell$ based on spectra of $A$ and $B$.

2. Solve $(A - q_1 I)Z_1 = U$ and $(B - \bar{p}_1 I)Y_1 = V$. Set $Z = Z_1$ and $Y = Y_1$.

3. Let $D = (q_1 - p_1)I$.

4. For $1 \leq j \leq \ell - 1$
   - Solve $(A - q_{j+1} I)Z_{j+1} = (q_{j+1} - p_j)Z_j$. Set $Z_{j+1} = Z_{j+1} + Z_j$ and $Z = [Z \ Z_{j+1}]$.
   - Solve $(B - \bar{p}_{j+1} I)Y_{j+1} = (\bar{p}_{j+1} - \bar{q}_j)Y_j$. Set $Y_{j+1} = Y_{j+1} + Y_j$ and $Y = [Y \ Y_{j+1}]$.
   - Set $D = \begin{bmatrix} D & (q_{j+1} - p_{j+1})I \ \end{bmatrix}$
fADI as a direct method

- Guaranteed to converge after all iterations
- Quasi-optimal shift parameters $p$ and $q$ are known in many situations [Fortunato & Townsend, 2020]
  [Townsend & Wilber, 2018]
- Zeros and poles of a rational function that can achieve a quasi-optimal Zolotarev number [Zolotarev, 1877]

$$Z_k(\Lambda(A), \Lambda(B)) := \inf_{r \in \mathcal{R}_{k,k}} \sup_{z \in \Lambda(A)} \left| \frac{r(z)}{\inf_{z \in \Lambda(B)} |r(z)|} \right|, \quad k \geq 0$$

$$||X - X_k||_F \leq Z_k(\Lambda(A), \Lambda(B)) ||X||_F \quad \text{[S. & Townsend, 2021]}$$
- Complexity $\mathcal{O}(rlT), \mathcal{O}(T)$ for solving shifted linear systems of $A$ and $B$
Goal: **Solve Sylvester tensor equations with fADI**

Method: **Rewrite into several Sylvester matrix equations**
Tensor-train format [Oseledets, 11]

\[ \mathcal{X}_{i_1, i_2, i_3} \approx \begin{array}{ccc} 1 \times s_1 & s_1 \times s_2 & s_2 \times 1 \\ G_1(i_1) & G_2(i_2) & (G_3) \end{array} \]

\[ G_j \in \mathbb{C}^{s_{j-1} \times n_j \times s_j} \]

\[ s_k \leq \text{rank}(X_k), \quad X_k = \text{reshape}(\mathcal{X}, \prod_{s=1}^{k} n_s, \prod_{s=k+1}^{3} n_s) \]

Trains: “somewhat” mimic the bases for column/row spaces of a matrix

\[ \mathcal{X} \times_1 A + \mathcal{X} \times_2 B + \mathcal{X} \times_3 C = \mathcal{F} \]

\[ F_1 = M_1 N_1^T \quad \text{and} \quad F_2 = M_2 N_2^T \]

\[ \mathcal{F} \text{ with TT cores } G_1, G_2, G_3 \text{ have low TT rank} \]

\[ F_1 = G_1(G_2)_1(G_3 \otimes I) \quad F_2 = (I \otimes G_1)(G_2)_2 G_3 \]
3D Sylvester solver in TT format

\[ \mathcal{X} \times_1 A + \mathcal{X} \times_2 B + \mathcal{X} \times_3 C = \mathcal{F} \]

**Method 1: Combine TTSVD with fADI**

**TTSVD [Oseledets, 11]**

Calculate SVD of \( X_1 \approx U_1 S_1 V_1^T \),
and use \( U_1 \) as first “train”

Let \( W = \text{reshape}(S_1 V_1^T, s_1 n_2, n_3) \),
and calculate SVD of \( W \approx U_2 S_2 V_2^T \).
Use \( \text{reshape}(U_2, s_1, n_2, s_2) \) as second “train”, and \( S_2 V_2^T \) as third “train”

**fADI steps [S. & Townsend, 21]**

\[ AX_1 + X_1 (I \otimes B + C \otimes I)^T = F_1 = M_1 N_1^T \]
Solve only for column space basis \( U_1 \)

\[ (I \otimes (U_1^T A U_1) + B \otimes I) W + WC^T = (I \otimes U_1^T) F_2 = (I \otimes U_1^T) M_2 N_2^T \]

\[ (I \otimes (U_1^T A U_1) + B \otimes I - \alpha I) Z = R \]

\[ (U_1^T A U_1 - \frac{\alpha}{2} I) Z_j + Z_j (B - \frac{\alpha}{2} I)^T = R_j \]

\( Z_j \) and \( R_j \) are reshape of \( j \)th column of \( Z \) and \( R \)
Example: Poisson equation

\[-(u_{xx} + u_{yy} + u_{zz}) = f \text{ on } \Omega = [-1,1]^3, \quad u |_{\partial \Omega} = 0\]

Ansatz \( \tilde{C}^{(3/2)} \): normalized ultraspherical

\[u(x, y, z) = (1 - x^2)(1 - y^2)(1 - z^2) \sum_{p=0}^{n} \sum_{q=0}^{n} \sum_{r=0}^{n} \mathcal{X}_{pqr} \tilde{C}^{(3/2)}_p(x) \tilde{C}^{(3/2)}_q(y) \tilde{C}^{(3/2)}_r(z),\]

\[f(x, y, z) = \sum_{p=0}^{n} \sum_{q=0}^{n} \sum_{r=0}^{n} \mathcal{F}_{pqr} \tilde{C}^{(3/2)}_p(x) \tilde{C}^{(3/2)}_q(y) \tilde{C}^{(3/2)}_r(z)\]

Sylvester equation

\[\mathcal{X} \times_1 A^{-1} + \mathcal{X} \times_2 A^{-1} + \mathcal{X} \times_3 A^{-1} = \mathcal{G}\]

\[\mathcal{G} = \mathcal{F} \times_1 M^{-1} \times_2 M^{-1} \times_3 M^{-1}\]

\[A = D^{-1}M, \text{ } D \text{ diagonal, } M \text{ and } A \text{ symmetric pentadiagonal}\]

\[\Lambda(A) \in [-1, -1/(30n^4)]\]

[Fortunato & Townsend, 20]

\[f = -2(1 - y^2)(1 - z^2) - 2(1 - x^2)(1 - z^2) - 2(1 - x^2)(1 - y^2)\]

\[u = (x^2 - 1)(y^2 - 1)(z^2 - 1)\]

Complexity \( \mathcal{O}(n(\log n)^3(\log(1/\epsilon))^3) \)
3D Sylvester solver in TT format

\[ \mathcal{X} X_1 A + \mathcal{X} X_2 B + \mathcal{X} X_3 C = \mathcal{F} \]

**Method 2: Combine parallel-TTSVD with fADI**

**parallel-TTSVD [S., Ruth & Townsend]**

Calculate SVD of \( X_1 \approx U_1 S_1 V_1^T \),
and use \( U_1 \) as first “train”

Calculate SVD of \( X_2 \approx U_2 S_2 V_2^T \). Use
\( \text{reshape}(U_1^* \text{reshape}(U_2, n_1, n_2 s_2), s_1, n_2, s_2) \)
as second “train” and \( S_2 V_2^T \) as third “train”

**fADI steps [S., Ruth & Townsend]**

\( AX_1 + X_1(I \otimes B + C \otimes I)^T = F_1 = M_1 N_1^T \)

Solve only for column space basis \( U_1 \)

\( (I \otimes A + B \otimes I)X_2 + X_2 C^T = F_2 = M_2 N_2^T \)

Only solve shifted linear systems with \( A, B, \) and \( C \)

Use universal shift parameters for both above equations, then find “trains” almost simultaneously
Example

\[ \mathbf{X} \times_1 D + \mathbf{X} \times_2 D + \mathbf{X} \times_3 D = \mathcal{F} \]

\( D \in \mathbb{R}^{n \times n} \) diagonal with \( D_j \in [-1, -1/(30n)] \)

\( \mathcal{F} \) has TT rank \((1, \lfloor n/4 \rfloor, 2, 1)\) with i.i.d. uniform random numbers in TT cores

Complexity \( \mathcal{O}(n(\log n)^3(\log(1/\epsilon)^3)) \)
Thank you!