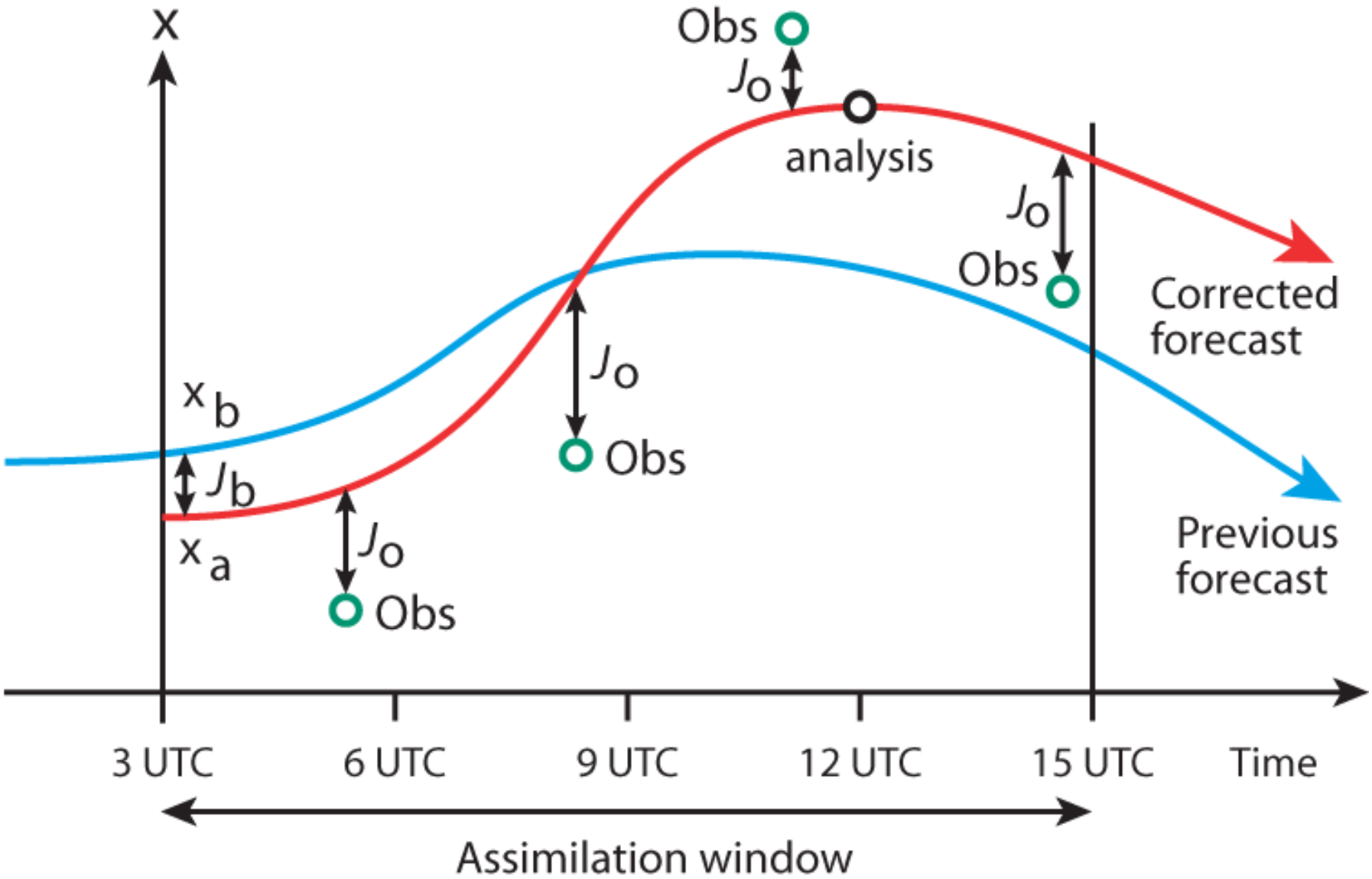
Stein-based Preconditioners for Weak-constraint 4D-Var

Davide Palitta Jemima M. Tabeart

Dipartimento di Matematica, Centro AM^2
Alma Mater Studiorum - Università di Bologna
University of Edinburgh

Sparse Days, 22nd June 2022, Saint-Girons

Data assimilation - what is it?



Variational DA can be viewed as a minimization problem

Need to solve

$$\min_{x \in \mathbb{R}^{s(N+1)}} J(x), \quad x = \text{vec}([x_0, \ldots, x_N]) = (x_0^T, \ldots, x_N^T)^T$$

where

$$J(x) = \frac{1}{2}\|x_0 - x_0^B\|^2_{B^{-1}} + \frac{1}{2}\sum_{i=0}^{N} \|y_i - \mathcal{H}_i(x_i)\|^2_{R_i^{-1}} + \frac{1}{2}\sum_{i=0}^{N-1} \|x_{i+1} - \mathcal{M}_i(x_i)\|^2_{Q_{i+1}^{-1}}$$

- $x_i \in \mathbb{R}^s$ model state at time t_i
- $y_i \in \mathbb{R}^p$ observation at time t_i
- $\mathcal{H}_i$ new observation operator, $y_i = \mathcal{H}_i(x_i^t) + \epsilon_i$, x_i^t true state, $\epsilon_i \sim \mathcal{N}(0, R_i)$
- $\mathcal{M}_i$ (inexact) forecast model, $x_{i+1} = \mathcal{M}_i(x_i) + \epsilon_i^M$, $\epsilon_i^M \sim \mathcal{N}(0, Q_i)$
- $x_0^B = x_0^t + \epsilon^B$, $\epsilon^B \sim \mathcal{N}(0, B)$

Reformulate the problem as a saddle-point system

In practice, solve via *incremental approach*: inner (linearised) - outer (update linearisations) to find $\delta x^{\ell} = x^{\ell+1} - x^{\ell}$

Reformulate the problem as a saddle-point system

In practice, solve via *incremental approach*: inner (linearised) - outer (update linearisations) to find $\delta x^{\ell} = x^{\ell+1} - x^{\ell}$

Dropping ℓ, $\delta x = \arg\min \delta J$ can be computed by solving the following saddle-point linear system **[Green, 2019]**

$$\begin{bmatrix} \mathrm{D} & 0 & \mathrm{L} \\ 0 & \mathrm{R} & \mathrm{H} \\ \mathrm{L}^T & \mathrm{H}^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \delta x \end{bmatrix} = \begin{bmatrix} b \\ d \\ 0 \end{bmatrix}$$

- $\mathrm{D} = \mathrm{blkdiag}(B, Q_1, \ldots, Q_N)$
- $\mathrm{R} = \mathrm{blkdiag}(R_0, \ldots, R_N)$
- $\mathrm{H} = \mathrm{blkdiag}(H_0, \ldots, H_N)$

Reformulate the problem as a saddle-point system

In practice, solve via *incremental approach*: inner (linearised) - outer (update linearisations) to find $\delta x^\ell = x^{\ell+1} - x^\ell$

Dropping ℓ, $\delta x = \arg\min \delta J$ can be computed by solving the following saddle-point linear system **[Green, 2019]**

$$\begin{bmatrix} \mathsf{D} & 0 & \mathsf{L} \\ 0 & \mathsf{R} & \mathsf{H} \\ \mathsf{L}^T & \mathsf{H}^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \delta x \end{bmatrix} = \begin{bmatrix} b \\ d \\ 0 \end{bmatrix}$$

- $\mathsf{D} = \text{blkdiag}(B, Q_1, \ldots, Q_N)$
- $\mathsf{R} = \text{blkdiag}(R_0, \ldots, R_N)$
- $\mathsf{H} = \text{blkdiag}(H_0, \ldots, H_N)$
- $\mathsf{L} = \begin{bmatrix} I & & & \\ -M_0 & \ddots & & \\ & \ddots & \ddots & \\ & & -M_{N-1} & I \end{bmatrix}$

Reformulate the problem as a saddle-point system

In practice, solve via *incremental approach*: inner (linearised) - outer (update linearisations) to find $\delta x^\ell = x^{\ell+1} - x^\ell$

Dropping ℓ, $\delta x = \arg\min \delta J$ can be computed by solving the following saddle-point linear system **[Green, 2019]**

$$\begin{bmatrix} \mathsf{D} & 0 & \mathsf{L} \\ 0 & \mathsf{R} & \mathsf{H} \\ \mathsf{L}^T & \mathsf{H}^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \delta x \end{bmatrix} = \begin{bmatrix} b \\ d \\ 0 \end{bmatrix}$$

- $\mathsf{D} = \text{blkdiag}(B, Q_1, \ldots, Q_N)$
- $\mathsf{R} = \text{blkdiag}(R_0, \ldots, R_N)$
- $\mathsf{H} = \text{blkdiag}(H_0, \ldots, H_N)$
- $\mathsf{L} = \begin{bmatrix} I & & & \\ -M_0 & \ddots & & \\ & \ddots & \ddots & \\ & & -M_{N-1} & I \end{bmatrix}$

What if: $Q_i \equiv Q \quad R_i \equiv R \quad H_i \equiv H \quad M_i \equiv M \quad \forall i$?

Saddle-point formulation

$$\begin{bmatrix} \mathrm{D} & 0 & \mathrm{L} \\ 0 & \mathrm{R} & \mathrm{H} \\ \mathrm{L}^T & \mathrm{H}^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \delta x \end{bmatrix} = \begin{bmatrix} b \\ d \\ 0 \end{bmatrix}$$

- $\mathrm{D} = \mathrm{blkdiag}(B, Q, \ldots, Q) = e_1 e_1^T \otimes B + (I_{N+1} - e_1 e_1^T) \otimes Q$
- $\mathrm{R} = \mathrm{blkdiag}(R, \ldots, R) = I_{N+1} \otimes R$
- $\mathrm{H} = \mathrm{blkdiag}(H, \ldots, H) = I_{N+1} \otimes H$
- $\mathrm{L} = \begin{bmatrix} I & & & \\ -M & \ddots & & \\ & \ddots & \ddots & \\ & & -M & I \end{bmatrix} = I_{N+1} \otimes I_s - \Sigma \otimes M,\ \Sigma = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}$

Preconditioners

Block preconditioners

$$\mathcal{P}_D = \begin{bmatrix} D & & \\ & R & \\ & & \tilde{S} \end{bmatrix}, \quad \mathcal{P}_T = \begin{bmatrix} D & 0 & L \\ & R & H \\ & & \tilde{S} \end{bmatrix}, \quad \mathcal{P}_C := \begin{bmatrix} D & 0 & \tilde{L} \\ 0 & R & 0 \\ \tilde{L}^T & 0 & 0 \end{bmatrix}$$

$$S = L^T D^{-1} L + H^T R^{-1} H$$

Need to design efficient approximations $\tilde{S}$.

Preconditioners

Block preconditioners

$$\mathcal{P}_D = \begin{bmatrix} \mathsf{D} & & \\ & \mathsf{R} & \\ & & \tilde{\mathsf{S}} \end{bmatrix}, \quad \mathcal{P}_T = \begin{bmatrix} \mathsf{D} & 0 & \mathsf{L} \\ & \mathsf{R} & \mathsf{H} \\ & & \tilde{\mathsf{S}} \end{bmatrix}, \quad \mathcal{P}_{\mathcal{C}} := \begin{bmatrix} \mathsf{D} & 0 & \tilde{\mathsf{L}} \\ 0 & \mathsf{R} & 0 \\ \tilde{\mathsf{L}}^T & 0 & 0 \end{bmatrix}$$

$$\mathsf{S} = \mathsf{L}^T \mathsf{D}^{-1} \mathsf{L} + \mathsf{H}^T \mathsf{R}^{-1} \mathsf{H}$$

Need to design efficient approximations $\tilde{\mathsf{S}}$.

1. $\widehat{\mathsf{S}} = L^T D^{-1} L$, $(\widehat{\mathsf{S}}^{-1} = L^{-1} D L^{-T})$

Preconditioners

Block preconditioners

$$\mathcal{P}_D = \begin{bmatrix} \mathsf{D} & & \\ & \mathsf{R} & \\ & & \tilde{\mathsf{S}} \end{bmatrix}, \quad \mathcal{P}_T = \begin{bmatrix} \mathsf{D} & 0 & \mathsf{L} \\ & \mathsf{R} & \mathsf{H} \\ & & \tilde{\mathsf{S}} \end{bmatrix}, \quad \mathcal{P}_{\mathcal{C}} := \begin{bmatrix} \mathsf{D} & 0 & \tilde{\mathsf{L}} \\ 0 & \mathsf{R} & 0 \\ \tilde{\mathsf{L}}^T & 0 & 0 \end{bmatrix}$$

$$\mathsf{S} = \mathsf{L}^T \mathsf{D}^{-1} \mathsf{L} + \mathsf{H}^T \mathsf{R}^{-1} \mathsf{H}$$

Need to design efficient approximations $\tilde{\mathsf{S}}$.

1. $\widehat{\mathsf{S}} = L^T D^{-1} L$, $(\widehat{\mathsf{S}}^{-1} = L^{-1} D L^{-T})$

$$\mathsf{L}^{-1} = \begin{bmatrix} I & & & & \\ -M & \ddots & & & \\ M^2 & \ddots & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ (-1)^N M^N & \cdots & M^2 & -M & I \end{bmatrix}$$

Preconditioners

Block preconditioners

$$\mathcal{P}_D = \begin{bmatrix} \mathsf{D} & & \\ & \mathsf{R} & \\ & & \tilde{\mathsf{S}} \end{bmatrix}, \quad \mathcal{P}_T = \begin{bmatrix} \mathsf{D} & 0 & \mathsf{L} \\ & \mathsf{R} & \mathsf{H} \\ & & \tilde{\mathsf{S}} \end{bmatrix}, \quad \mathcal{P}_C := \begin{bmatrix} \mathsf{D} & 0 & \tilde{\mathsf{L}} \\ 0 & \mathsf{R} & 0 \\ \tilde{\mathsf{L}}^T & 0 & 0 \end{bmatrix}$$

$$\mathsf{S} = \mathsf{L}^T \mathsf{D}^{-1} \mathsf{L} + \mathsf{H}^T \mathsf{R}^{-1} \mathsf{H}$$

Need to design efficient approximations $\tilde{\mathsf{S}}$.

1. $\widehat{\mathsf{S}} = L^T D^{-1} L$
   $\widehat{\mathsf{S}}^{-1} = L^{-1} D L^{-T}$
2. $\tilde{\mathsf{L}} \approx \mathsf{L} = I_{N+1} \otimes I_s - \Sigma \otimes M$

$$\tilde{\mathsf{L}}_1 = I_{N+1} \otimes I_s - \Sigma_1 \otimes I_s \quad \text{or} \quad \tilde{\mathsf{L}}_2^{-1} = \sum_{k=0}^{\bar{k}} (-1)^k \Sigma^k \otimes M^k$$

Instead of parallelism, we exploit the Kronecker structure

L is a Stein operator!

$$\mathsf{L} = I_{N+1} \otimes I_s - \Sigma \otimes M$$

i.e., given $x = \mathrm{vec}(X)$, $v = \mathrm{vec}(V)$

$$x = \mathsf{L}^{-1}(v) \quad \Longleftrightarrow \quad X - MX\Sigma^T = V$$

Many different efficient methods for solving Stein matrix equations - can use the exact L^{-1} in a preconditioner.

Caveat:

Our method relies on the eigendecomposition of M: we need moderate values of s

Matrix-oriented GMRES

Let's exploit the Kronecker product to solve our problem

$$\begin{bmatrix} \mathsf{D} & 0 & \mathsf{L} \\ 0 & \mathsf{R} & \mathsf{H} \\ \mathsf{L}^T & \mathsf{H}^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \delta x \end{bmatrix} = \begin{bmatrix} b \\ d \\ 0 \end{bmatrix}$$

- Exploit the Kronecker product in the "matrix-vector" products
- Represent the basis "vectors" by matrices
- Use the matrix-inner product in the orthogonalization
- Equivalent in exact arithmetic to standard GMRES.

Matrix-oriented GMRES

$$\begin{bmatrix} \mathsf{D} & 0 & \mathsf{L} \\ 0 & \mathsf{R} & \mathsf{H} \\ \mathsf{L}^T & \mathsf{H}^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \delta x \end{bmatrix} = \begin{bmatrix} b \\ d \\ 0 \end{bmatrix}$$

- $b = \mathrm{vec}([b_0, c_1, \ldots, c_N]) = \mathrm{vec}(V_1^{(1)}) \in \mathbb{R}^{(N+1)s}$
- $d = \mathrm{vec}([d_0, \ldots, d_N]) = \mathrm{vec}(V_1^{(2)}) \in \mathbb{R}^{(N+1)p}$

$$\begin{bmatrix} \mathsf{D} & 0 & \mathsf{L} \\ 0 & \mathsf{R} & \mathsf{H} \\ \mathsf{L}^T & \mathsf{H}^T & 0 \end{bmatrix} \begin{bmatrix} b \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} \mathsf{D}b \\ \mathsf{R}d \\ \mathsf{L}^T b + \mathsf{H}^T d \end{bmatrix} = \mathrm{vec}\left(\begin{bmatrix} BV_1^{(1)} e_1 e_1^T + QV_1^{(1)}(I - e_1 e_1^T) \\ RV_1^{(2)} \\ V_1^{(1)} - M^T V_1^{(1)} \Sigma + H^T V_1^{(2)} \end{bmatrix} \right)$$

Stein-based preconditioners - $\mathcal{P}_{\mathcal{D}}$

Let's use $\tilde{\mathrm{L}} = \mathrm{L}$, $\tilde{\mathrm{S}} = \mathrm{L}^T \mathrm{D}^{-1} \mathrm{L}$, at k-th iteration

$$\mathcal{P}_D^{-1} v_k = \begin{bmatrix} \mathrm{D}^{-1} & & \\ & \mathrm{R}^{-1} & \\ & & \tilde{\mathrm{S}}^{-1} \end{bmatrix} \mathrm{vec}\left(\begin{bmatrix} V_k^{(1)} \\ V_k^{(2)} \\ V_k^{(3)} \end{bmatrix} \right) = \begin{bmatrix} \mathrm{D}^{-1}\mathrm{vec}\left(V_k^{(1)}\right) \\ \mathrm{R}^{-1}\mathrm{vec}\left(V_k^{(2)}\right) \\ \tilde{\mathrm{S}}^{-1}\mathrm{vec}\left(V_k^{(3)}\right) \end{bmatrix}$$

Exploit the structure of $\mathrm{D} = e_1 e_1^T \otimes B + (I - e_1 e_1^T) \otimes Q$ and $\mathrm{R} = I \otimes R$

- $\mathrm{D}^{-1}\mathrm{vec}\left(V_k^{(1)}\right) = \mathrm{vec}(W)$ where

$$W(:,1) = B^{-1} V_k^{(1)}(:,1), \quad W(:,2:N+1) = Q^{-1} V_k^{(1)}(:,2:N+1)$$

- $\mathrm{R}^{-1}\mathrm{vec}\left(V_k^{(2)}\right) = \mathrm{vec}\left(R^{-1} V_k^{(2)}\right)$

Stein-based preconditioners - $\mathcal{P}_{\mathcal{D}}$

Let's use $\tilde{\mathsf{L}} = \mathsf{L}$, $\tilde{\mathsf{S}} = \mathsf{L}^T\mathsf{D}^{-1}\mathsf{L}$, at k-th iteration

$$\mathcal{P}_D^{-1} v_k = \begin{bmatrix} \mathsf{D}^{-1} & & \\ & \mathsf{R}^{-1} & \\ & & \tilde{\mathsf{S}}^{-1} \end{bmatrix} \mathrm{vec}\left(\begin{bmatrix} V_k^{(1)} \\ V_k^{(2)} \\ V_k^{(3)} \end{bmatrix} \right) = \begin{bmatrix} \mathsf{D}^{-1}\mathrm{vec}\left(V_k^{(1)}\right) \\ \mathsf{R}^{-1}\mathrm{vec}\left(V_k^{(2)}\right) \\ \tilde{\mathsf{S}}^{-1}\mathrm{vec}\left(V_k^{(3)}\right) \end{bmatrix}$$

Exploit the structure of $\mathsf{L} = I_{N+1} \otimes I_s - \Sigma \otimes M$

- $\tilde{\mathsf{S}}^{-1}\mathrm{vec}\left(V_k^{(3)}\right) = \mathsf{L}^{-1}\mathsf{D}\mathsf{L}^{-T}\mathrm{vec}\left(V_k^{(3)}\right) = \mathrm{vec}\,(Z)$ where
  1. Solve the Stein matrix equation
  $$X - M^T X \Sigma = V_k^{(3)}$$
  2. Set $Y = [BX(:,1), QX(:,2:N+1)]$
  3. Solve the Stein matrix equation
  $$Z - MZ\Sigma^T = Y$$

Eigenvalue bounds for $\mathcal{P}_D$

Can we employ better approximation $\tilde{S}$ to $S = L^T D^{-1} L + H^T R^{-1} H$?

If we use $\tilde{S} = S$

$$\lambda(\mathcal{P}_D^{-1}\mathcal{A}) \in \left\{ \frac{1-\sqrt{5}}{2}, 1, \frac{1+\sqrt{5}}{2} \right\}$$

More performing $\tilde{S}$

Original idea: Include low-rank approximation to second term in S

$$\mathrm{H}^T\mathrm{R}^{-1}\mathrm{H} = I_{N+1} \otimes (H^T R^{-1} H) \approx I_{N+1} \otimes (K_r K_r^T) = \mathrm{K}_r\mathrm{K}_r^T,\ \mathrm{K}_r = I_{N+1} \otimes K_r$$

and define

$$\tilde{\mathrm{S}} = \mathrm{L}^T\mathrm{D}^{-1}\mathrm{L} + \mathrm{K}_r\mathrm{K}_r^T$$

so that (Sherman-Morrison-Woodbury)

$$\begin{aligned}\tilde{\mathrm{S}}^{-1} =& \mathrm{L}^{-1}\mathrm{D}\mathrm{L}^{-T} \\ &- \mathrm{L}^{-1}\mathrm{D}\mathrm{L}^{-T}\mathrm{K}_r \left(I_{r(N+1)} + \mathrm{K}_r^T\mathrm{L}^{-1}\mathrm{D}\mathrm{L}^{-T}\mathrm{K}_r\right)^{-1} \mathrm{K}_r^T\mathrm{L}^{-1}\mathrm{D}\mathrm{L}^{-T}\end{aligned}$$

- Exploit Kronecker structure of D, L, and K_r
- Main issue: $\left(I_{r(N+1)} + \mathrm{K}_r^T\mathrm{L}^{-1}\mathrm{D}\mathrm{L}^{-T}\mathrm{K}_r\right)^{-1}$

More performing $\tilde{S}$

$$\left(I_{r(N+1)} + K_r^T L^{-1} D L^{-T} K_r\right) x = b \tag{1}$$

- We cannot compute $I_{r(N+1)} + K_r^T L^{-1} D L^{-T} K_r$: large and dense
- We can efficiently perform matrix-vector products with it
- $I_{r(N+1)} + K_r^T L^{-1} D L^{-T} K_r$ is SPD $\rightarrow$ matrix-oriented CG for (1)
- We can afford $r = p$ (potentially $\tilde{S} = S$)
- The iterative solution of (1) implies $\tilde{S} \approx S$ even when $r = p$
- The overall method is a inner-outer scheme $\rightarrow$ flexible (outer) Krylov solver

Numerical results

Example 2 from **[Pearson, Tabeart (2021)]** with $s = 1000$, $p = 500$

- B and Q generated by adapted SOAR correlation function **[Pearson, Tabeart (2021)]** with $L_B = 0.6, L_Q = 0.75$, $\sigma_B = 0.5$, $\sigma_Q = 0.2$, 100 non-zero entries per row for B and 120 for Q

$$\mathrm{D} = e_1 e_1^T \otimes B + (I_{N+1} - e_1 e_1^T) \otimes Q \in \mathbb{R}^{1000(N+1)\times 1000(N+1)}$$

- $R \in \mathbb{R}^{500\times 500}$ is produced using the block approach of **[Pearson, Tabeart (2021)]**

$$\mathrm{R} = I_{N+1} \otimes R \in \mathbb{R}^{500(N+1)\times 500(N+1)}$$

- $H \in \mathbb{R}^{500\times 1000}$ has a single unit entry per row, arranged in ascending column order

$$\mathrm{H} = I_{N+1} \otimes H \in \mathbb{R}^{500(N+1)\times 1000(N+1)}$$

Numerical results

Example 2 from **[Pearson, Tabeart (2021)]** with $s = 1000$, $p = 500$

- $\mathsf{L} = I_{N+1} \otimes I_s - \Sigma \otimes M \in \mathbb{R}^{1000(N+1)\times 1000(N+1)}$ where

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1-2r & r & 0 & \cdots & 0 & 0 \\ 0 & r & \ddots & \ddots & & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & r & 0 \\ 0 & \cdots & 0 & 0 & r & 1-2r & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad r = 4e-1$$

M corresponds to a discretized heat equation (Forward Euler + 2nd order centered FD)

Numerical results

Solver: (F)GMRES (tol: 10^{-8})

Different preconditioning frameworks: $\mathcal{P}_{\mathcal{D}}$, $\mathcal{P}_{\mathcal{T}}$, and $\mathcal{P}_{\mathcal{C}}$

Within all of them we use different $\tilde{\mathrm{L}}$ (and $\tilde{\mathrm{S}}$ if needed)

- $\tilde{\mathrm{L}}$ from **[Pearson, Tabeart (2021)]** with $k = 3$ ($\tilde{\mathrm{L}} \neq \mathrm{L}$)
- $\tilde{\mathrm{L}} = \mathrm{L}$ and $\tilde{\mathrm{S}} = \mathrm{L}^T\mathrm{D}^{-1}\mathrm{L}$ ($r = 0$)
- $\tilde{\mathrm{L}} = \mathrm{L}$ and $\tilde{\mathrm{S}} \approx \mathrm{L}^T\mathrm{D}^{-1}\mathrm{L} + \mathrm{H}^T\mathrm{R}^{-1}\mathrm{H}$ ($r = p$, $tol_{CG} = 10^{-8}$)

Numerical results

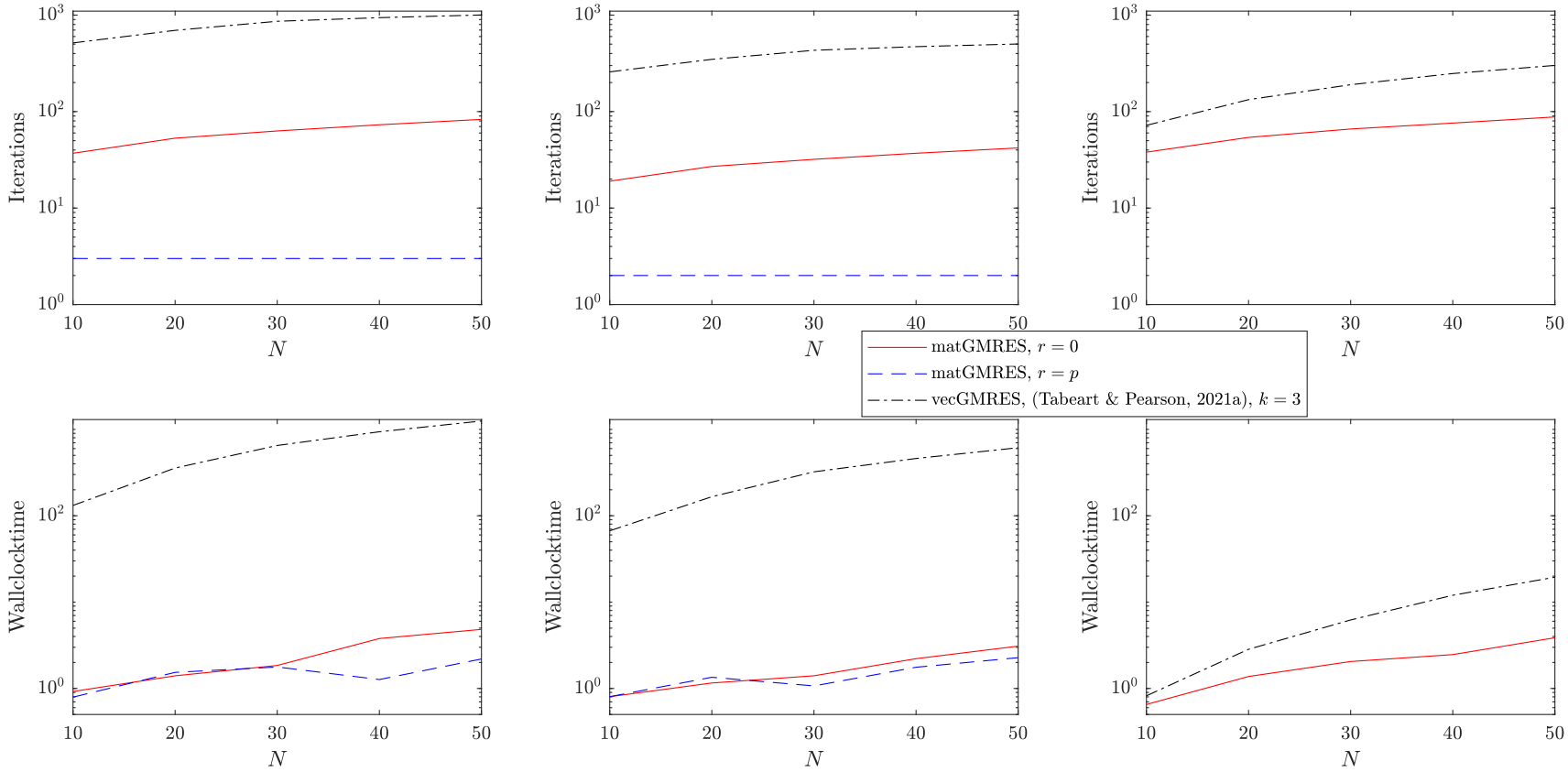


Figure: Top: Number of iterations. Bottom: Wallclock time. Left: $\mathcal{P}_{\mathcal{D}}$. Center: $\mathcal{P}_{\mathcal{T}}$. Right: $\mathcal{P}_{\mathcal{C}}$

More challenging problems: time-dependent M_i's

$$\mathsf{L} = \begin{bmatrix} I & & & \\ -M_1 & \ddots & & \\ & \ddots & \ddots & \\ & & -M_N & I \end{bmatrix}$$

L is no longer a Stein operator!

- Approximate L with a Stein operator at the preconditioning level

$$\tilde{\mathsf{L}} = I_{N+1} \otimes I_s - \Sigma \otimes \widehat{M} \approx \mathsf{L}$$

- $\widehat{M}$ some representative values of the M_i's

Numerical results

Lorenz 96: the M_i's stem from the discretization of following problem

- s equally spaced points on the unit line

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + 8, \quad i = 1, \ldots, s$$

  w/ periodic boundary conditions ($x_{-1} = x_{s-1}, x_0 = s_s$)

- Forward in time integration by fourth-order Runge-Kutta scheme **[El-Said, 2015]**
- $s = 1000$, $\Delta t = 10^{-6}$

Same setting as before for D, R, and H

Numerical results

$N = 100$

	$\mathcal{P}_{\mathcal{D}}$	$\mathcal{P}_{\mathcal{T}}$	$\mathcal{P}_{\mathcal{C}}$
matGMRES, $\widehat{M} = M_{100}$, $r = 0$	229	115	242
matGMRES, $\widehat{M} = M_{100}$, $r = p$	5	3	-
vecGMRES [Pearson, Tabeart, 2021], $k = 3$	-	922	506
matGMRES, $\widehat{M} = M_{100}$, $r = 0$	56.2311	19.9438	64.2279
matGMRES, $\widehat{M} = M_{100}$, $r = p$	11.0382	11.0639	-
vecGMRES [Pearson, Tabeart, 2021], $k = 3$	2886.6	2639.4	283.9

Conclusions

Conclusions

- Exploiting Kronecker and matrix equation structures often leads to remarkable computational gains
- New preconditioners with very competitive computational records and appealing theoretical properties
- Optimal results for time-independent forecast models
- Very good results for time-dependent M_i's as well

Outlooks

- Larger spatial dimension s
- Relax Kronecker structure assumption on D, R, and H

Reference: *Stein-based Preconditioners for Weak-constraint 4D-Var*
D. Palitta and J. M. Tabeart
ArXiv: 2203.17184

Same preconditioning frameworks

Block preconditioners

$$\mathcal{P}_D = \begin{bmatrix} \mathsf{D} & & \\ & \mathsf{R} & \\ & & \tilde{\mathsf{S}} \end{bmatrix}, \quad \mathcal{P}_T = \begin{bmatrix} \mathsf{D} & 0 & \tilde{\mathsf{L}} \\ & \mathsf{R} & \mathsf{H} \\ & & \tilde{\mathsf{S}} \end{bmatrix}, \quad \mathcal{P}_{\mathcal{C}} := \begin{bmatrix} \mathsf{D} & 0 & \tilde{\mathsf{L}} \\ 0 & \mathsf{R} & 0 \\ \tilde{\mathsf{L}}^T & 0 & 0 \end{bmatrix}$$

- $\tilde{\mathsf{L}} = I_{N+1} \otimes I_s - \Sigma \otimes \widehat{M}$
- $\tilde{\mathsf{S}} = \tilde{\mathsf{L}}^T \mathsf{D}^{-1} \tilde{\mathsf{L}}$ $(r = 0)$
- $\tilde{\mathsf{S}} \approx \tilde{\mathsf{L}}^T \mathsf{D}^{-1} \tilde{\mathsf{L}} + \mathsf{H}^T \mathsf{R}^{-1} \mathsf{H}$ $(r = p)$

Same machinery as before

Numerical results

$$\tilde{\mathsf{L}} = I_{N+1} \otimes I_s - \Sigma \otimes \widehat{M}$$

Different options for $\widehat{M}$

- $\widehat{M} = M_1$
- $\widehat{M} = M_N$
- $\widehat{M} = karch(M_i)$, Karcher mean of $\{1/2(M_i + M_i^T)\}_i$

Numerical results

$N = 10$

	$\mathcal{P}_{\mathcal{D}}$	$\mathcal{P}_{\mathcal{T}}$	$\mathcal{P}_{\mathcal{C}}$
matGMRES, $\widehat{M} = M_1$, $r = 0$	45	29	46
matGMRES, $\widehat{M} = M_{10}$, $r = 0$	45	29	46
matGMRES, $\widehat{M} = karch(M_i)$, $r = 0$	45	29	46
matGMRES, $\widehat{M} = M_1$, $r = p$	7	6	-
matGMRES, $\widehat{M} = M_{10}$, $r = p$	7	6	-
matGMRES, $\widehat{M} = karch(M_i)$, $r = p$	7	6	-
vecGMRES **[Pearson, Tabeart, 2021]**, $k = 3$	493	262	84
matGMRES, $\widehat{M} = M_1$, $r = 0$	11.1453	7.2486	10.3794
matGMRES, $\widehat{M} = M_{10}$, $r = 0$	11.5554	7.5041	10.4503
matGMRES, $\widehat{M} = karch(M_i)$, $r = 0$	11.3662	7.4592	10.3609
matGMRES, $\widehat{M} = M_1$, $r = p$	3.8272	4.0534	-
matGMRES, $\widehat{M} = M_{10}$, $r = p$	4.3331	4.1532	-
matGMRES, $\widehat{M} = karch(M_i)$, $r = p$	3.847	4.1523	-
vecGMRES **[Pearson, Tabeart, 2021]**, $k = 3$	152.8040	80.8425	4.7303

Computing L^{-1}

$X - MX\Sigma_1^T = V$ w/ $\Sigma_1 = C_1 - e_1 e_N^T$, $C_1 = \mathcal{F}^{-1}\Pi\mathcal{F}$, $\Pi = \mathrm{diag}(\mathcal{F}(C_1 e_1))$, $\mathcal{F}$ DFT matrix

$$X - MXC_1^T + MXe_N e_1^T = V$$

If $M = S\Lambda S^{-1}$ and $\widetilde{X} := S^{-1}X\mathcal{F}^T$, we solve

$$\widetilde{X} - \Lambda\widetilde{X}\Pi + \Lambda\widetilde{X}(\mathcal{F}^{-T}e_N)(\mathcal{F}e_1)^T = S^{-1}V\mathcal{F}^T$$

By applying the SMW formula to the Kronecker form + some work (properties of the inner matrix product and the Hadamard product)

$$\Downarrow$$

$$X = S(Z - W)\mathrm{F}^{-T}$$

$Z = H \odot \left(S^{-1}V\mathcal{F}^T\right)$

$W = H \odot \left(\left(\left(I + \Lambda \cdot \mathrm{diag}\left(H\left(\mathcal{F}^{-T}e_N \odot \mathcal{F}e_1\right)\right)\right)^{-1} Z\mathcal{F}^{-T}e_N\right) e_1^T\mathcal{F}^T\right)$

$H_{i,j} = 1/(1 - \lambda_i \cdot e_j^T(\mathcal{F}(C_1 e_1)))$, $\odot$ Hadamard product

A good $\widehat{M}$?

Proposition

Let $D_i = \widehat{M} - M_i$. The eigenvalues of $\tilde{\mathrm{L}}^{-T}\mathrm{L}^T\mathrm{L}\tilde{\mathrm{L}}^{-1}$ can be bounded above by

$$1 + \frac{N}{2}\left(\rho_N + \sqrt{\rho_N^2 + 4\rho_N}\right)$$

where

$$\rho_N = \begin{cases} N \cdot \max_{m=1,\dots,N} \lambda_{max}(D_m^T D_m), & \text{if } \lambda_{\max}(\widehat{M}^T\widehat{M}) = 1, \\ \frac{1-\lambda_{\max}^N(\widehat{M}^T\widehat{M})}{1-\lambda_{\max}(\widehat{M}^T\widehat{M})} \cdot \max_{m=1,\dots,N} \lambda_{max}(D_m^T D_m), & \text{otherwise.} \end{cases}$$

- The smaller $\max_{m=1,\dots,N} \lambda_{max}(D_m^T D_m)$, and $\lambda_{\max}(\widehat{M}^T\widehat{M})$, the better

A good $\widehat{M}$?

Lorenz 96, $N = 100$

$\widehat{M}$	M_1	M_{100}	$Karch(M_i)$
$\|\widehat{M}\|$	1.000	1.0023	1.0011
$\max_m \lambda_{\max}(D_m^T D_m)$	1.7641×10^{-5}	1.7641×10^{-5}	1.3981×10^{-5}