Computational methods for large-scale matrix equations and application to PDEs

V. Simoncini

Dipartimento di Matematica

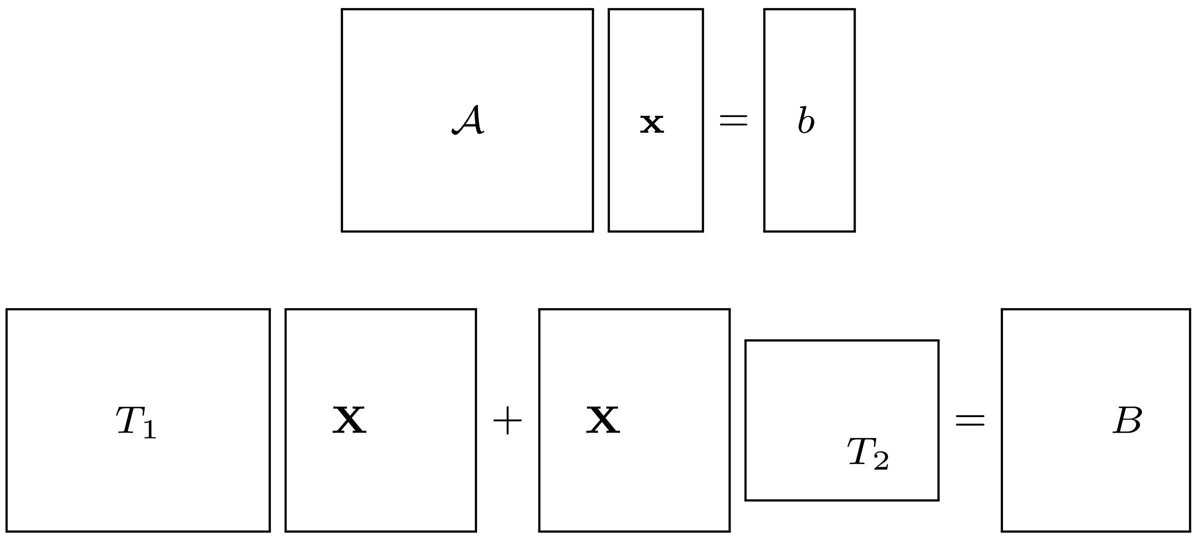
Alma Mater Studiorum - Università di Bologna

valeria.simoncini@unibo.it

Linear (vector) systems and linear **matrix** equations

Problem: solve the linear problem

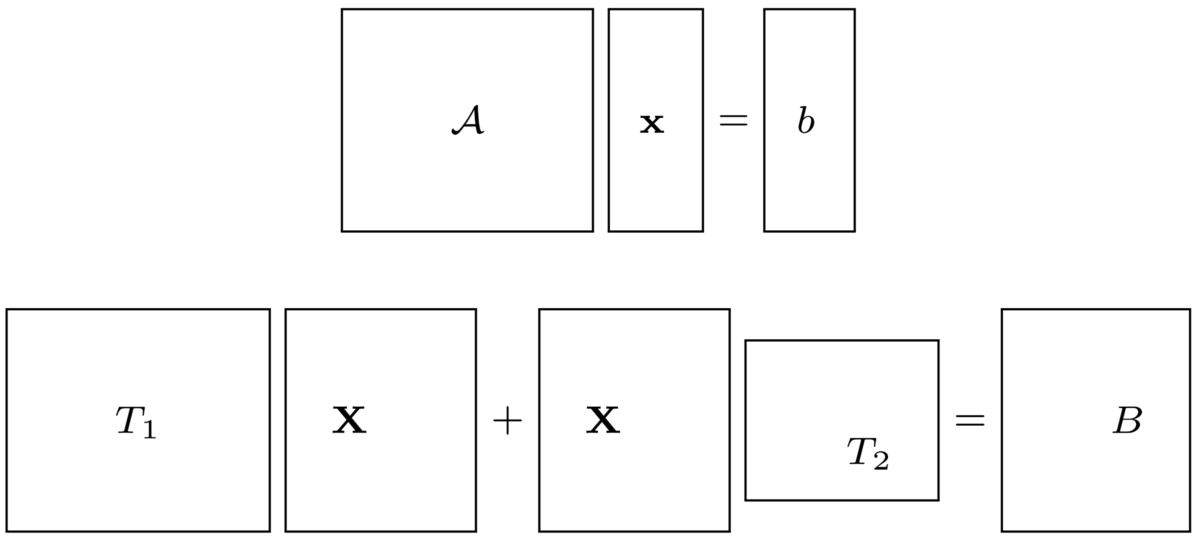
$$\mathcal{A}\mathbf{x} = b \qquad \text{or} \qquad T_1\mathbf{X} + \mathbf{X}T_2 = B$$



Linear (**vector**) systems and linear **matrix** equations

Problem: solve the linear problem

$$\mathcal{A}\mathbf{x} = b \qquad \text{or} \qquad T_1\mathbf{X} + \mathbf{X}T_2 = B$$

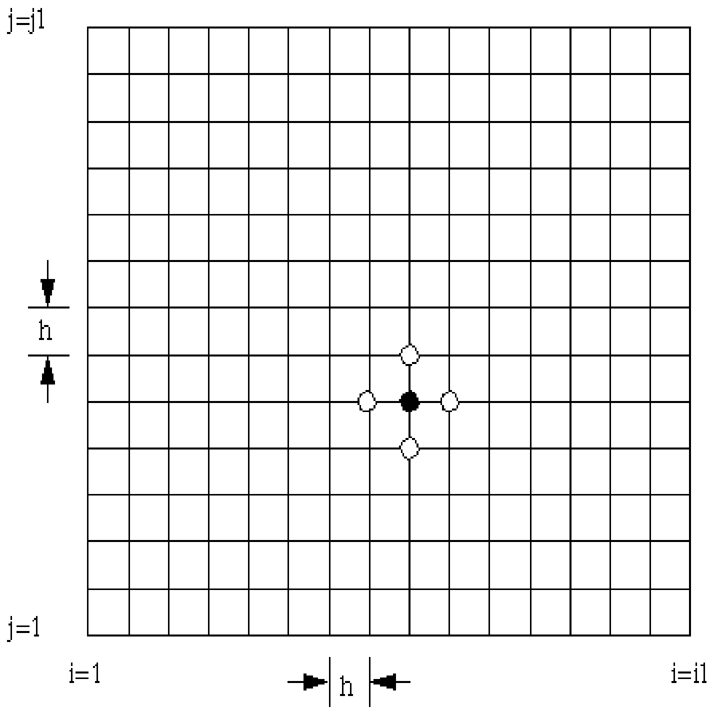


Remark: In discretizing PDEs with tensor bases, the two problems may be mathematically equivalent !

The Poisson equation

$$-u_{xx} - u_{yy} = f, \quad \text{in} \quad \Omega = (0,1)^2$$

+ Dirichlet b.c. (zero b.c. for simplicity)



The Poisson equation

$$-u_{xx} - u_{yy} = f, \quad \text{in} \quad \Omega = (0,1)^2 \quad + \text{Dirichlet zero b.c.}$$

FD Discretization: $U_{i,j} \approx u(x_i, y_j)$, with (x_i, y_j) interior nodes, so that

$$u_{xx}(x_i, y_j) \approx \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h^2} = \frac{1}{h^2}[1, -2, 1] \begin{bmatrix} U_{i-1,j} \\ U_{i,j} \\ U_{i+1,j} \end{bmatrix}$$

$$u_{yy}(x_i, y_j) \approx \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{h^2} = \frac{1}{h^2}[U_{i,j-1},\, U_{i,j},\, U_{i,j+1}] \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$-T_1\mathbf{U} - \mathbf{U}T_1^\top = F, \quad F_{ij} = f(x_i, y_j), \quad T_1 = \tfrac{1}{h^2}\text{tridiag}(1, -2, 1)$$

The Poisson equation

$$-u_{xx} - u_{yy} = f, \quad \text{in} \quad \Omega = (0,1)^2 \quad + \text{Dirichlet zero b.c.}$$

FD Discretization: $U_{i,j} \approx u(x_i, y_j)$, with (x_i, y_j) interior nodes, so that

$$u_{xx}(x_i, y_j) \approx \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h^2} = \frac{1}{h^2}[1, -2, 1] \begin{bmatrix} U_{i-1,j} \\ U_{i,j} \\ U_{i+1,j} \end{bmatrix}$$

$$u_{yy}(x_i, y_j) \approx \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{h^2} = \frac{1}{h^2}[U_{i,j-1}, \, U_{i,j}, \, U_{i,j+1}] \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$-T_1\mathbf{U} - \mathbf{U}T_1^\top = F, \quad F_{ij} = f(x_i, y_j), \quad T_1 = \tfrac{1}{h^2}\text{tridiag}(1, -2, 1)$$

Lexicographic ordering:
$\mathbf{U} \to \text{vec}(\mathbf{U}) = \mathbf{u} = [\mathbf{U}_{11}, ..., \mathbf{U}_{n,1}, \mathbf{U}_{1,2}, \ldots, \mathbf{U}_{n,2}, \ldots]^\top$

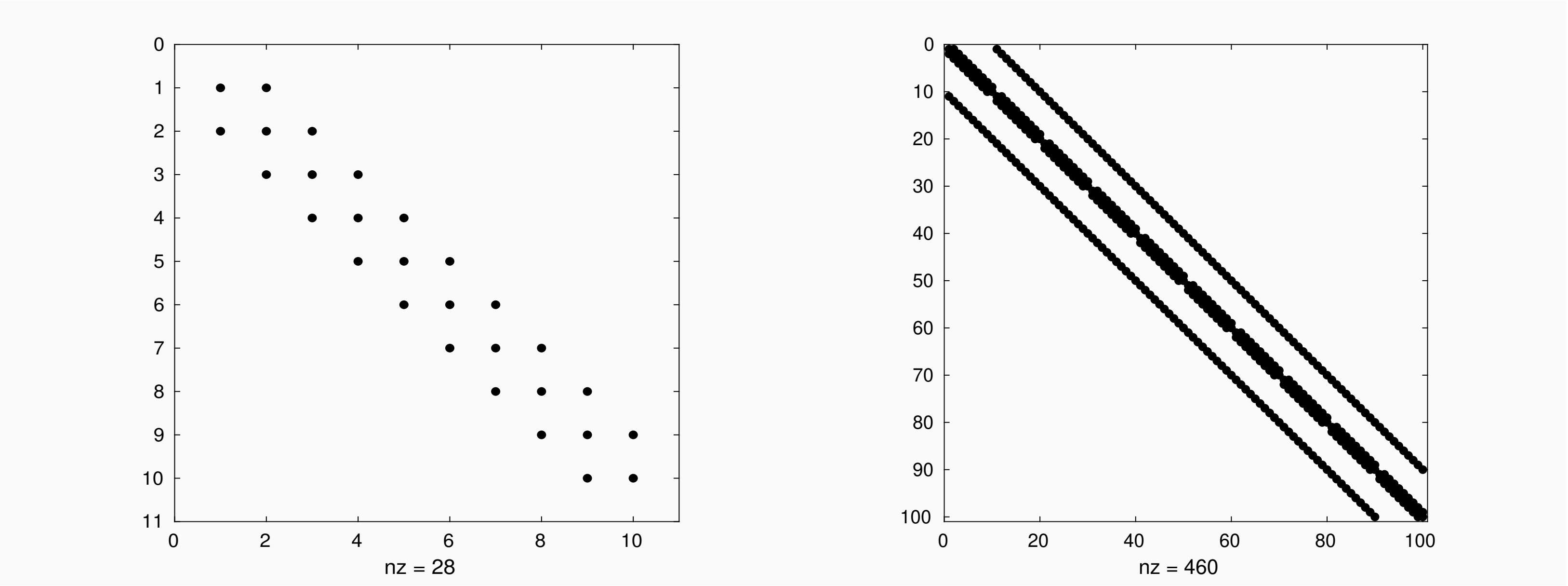
$$\boxed{\mathcal{A}\mathbf{u} = f} \qquad \text{with} \quad \mathcal{A} = -I \otimes T_1 - T_1 \otimes I, \quad f = \text{vec}(F)$$

($(M \otimes N)$ Kronecker product, $(M \otimes N) = (M_{i,j}N)$)

Computational considerations

$$T_1\mathbf{U} + \mathbf{U}T_2 = F, \qquad T_i \in \mathbb{R}^{n_i \times n_i}$$

$$\mathcal{A}\mathbf{u} = f \qquad \mathcal{A} = I \otimes T_1 + T_2 \otimes I \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$$



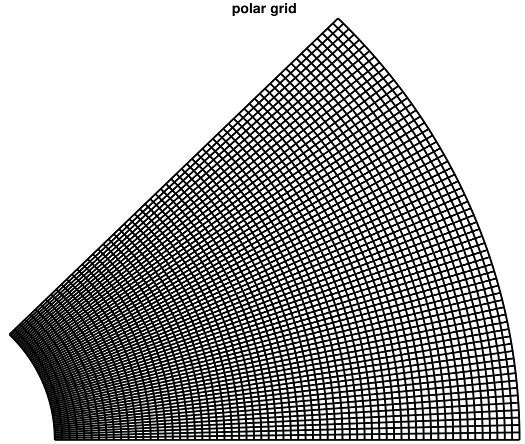
T_1 $\qquad\qquad$ $\mathcal{A}$

Discretization of more complex domains (with Y. Hao)

$$-u_{xx} - u_{yy} = f, \quad \text{in} \quad \Omega$$

$$(x, y) \in \Omega, \quad x = r\cos\theta,\ y = r\sin\theta$$

$$(r, \theta) \in [r_0, r_1] \times [0, \frac{\pi}{4}]$$



♣ Transformed equation in polar coordinates:

$$-r^2\tilde{u}_{rr} - r\tilde{u}_r - \tilde{u}_{\theta\theta} = \tilde{f}, \qquad (r, \theta) \in [r_0, r_1] \times [0, \frac{\pi}{4}]$$

Matrix equation after mapping to the rectangle:

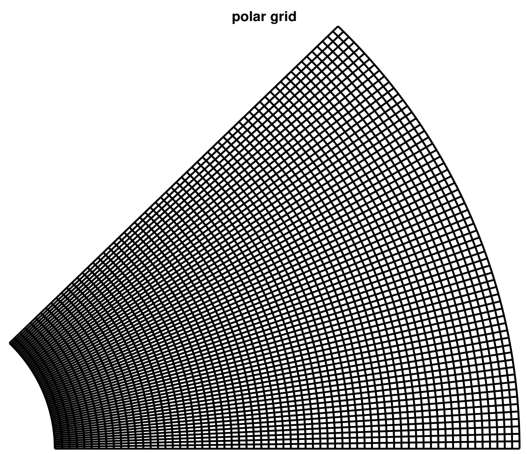
$$\boxed{\Phi^2 T\widetilde{\boldsymbol{U}} + \widetilde{\boldsymbol{U}}T - \Phi B\widetilde{\boldsymbol{U}} = \widetilde{F}} \quad \Leftrightarrow \quad \boxed{(\Phi^2 T - \Phi B)\widetilde{\boldsymbol{U}} + \widetilde{\boldsymbol{U}}T = \widetilde{F}}$$

Discretization of more complex domains (with Y. Hao)

$$-u_{xx} - u_{yy} = f, \quad \text{in} \quad \Omega$$

$$(x, y) \in \Omega, \quad x = r\cos\theta,\ y = r\sin\theta$$

$$(r, \theta) \in [r_0, r_1] \times [0, \frac{\pi}{4}]$$



♣ Transformed equation in log-polar coordinates ($r = e^{\rho}$):

$$-\hat{u}_{\rho\rho} - \hat{u}_{\theta\theta} = \hat{f}, \qquad (\rho, \theta) \in [\rho_0, \rho_1] \times [0, \frac{\pi}{4}]$$

Matrix equation after mapping to the rectangle:

$$\boxed{T\widehat{\boldsymbol{U}} + +\widehat{\boldsymbol{U}}T = \widehat{F}}$$

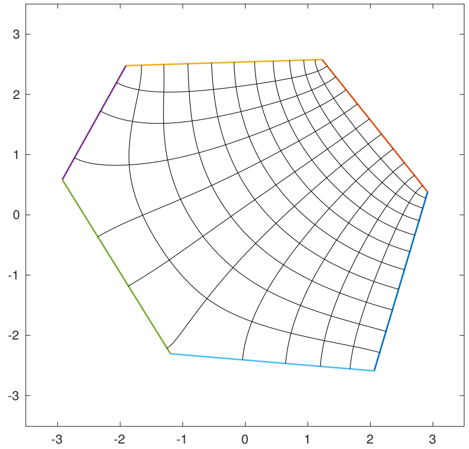
Poisson equation in a polygon with more than 4 edges (with Y. Hao)

♣ Schwarz-Christoffel conformal mappings between polygon Ω and rectangle Π

$$-u_{xx} - u_{yy} = f, \qquad (x, y) \in \Omega$$

$$\boxed{-\widetilde{u}_{\xi\xi} - \widetilde{u}_{\eta\eta} = \mathscr{J}\,\widetilde{f}, \qquad (\xi, \eta) \in \Pi}$$

($\mathscr{J}$ Jacobian det of SC mapping)



With finite difference discretization in Π:

$$\boxed{T_1 U + U T_2 = F}, \qquad F = \widetilde{F} + b.c., \quad \text{and} \quad \widetilde{F}_{i,j} = (\mathscr{J}\,\widetilde{f})(\xi_i, \eta_j),\ 1 \le i \le n_1,\ 1 \le j \le n_2$$

Poisson equation is the ideal setting for SC mappings!

More general settings

- Convection-diffusion eqns in a rectangle
  (see, e.g., Palitta & Simoncini, 2016)
- Space-Time discretizations via tensorized high order methods
  (see, e.g., joint wrk w/ Henning, Palitta and Urban, 2020)
- Galerkin FE discretization of Stochastic PDEs
  (see, e.g., joint wrk w/ Powell and Silvester, 2017)
- Isogeometric Analysis
  (see, e.g., Sangalli and Tani, 2016)
- ...

... A classical approach, Bickley & McNamee, 1960, Wachspress, 1963
(Early literature on difference equations)

Numerical solution of the Sylvester equation

$$A\boldsymbol{U} + \boldsymbol{U}B = G$$

Various settings:

- Small A and small B: Bartels-Stewart algorithm

  1. Compute the Schur forms:
     $A^* = URU^*$, $B = VSV^*$ with R, S upper triangular;
  2. Solve $R^*\boldsymbol{Y} + \boldsymbol{Y}S = U^*GV$ for $\boldsymbol{Y}$ (element-wise);
  3. Compute $\boldsymbol{U} = U\boldsymbol{Y}V^*$

Numerical solution of the Sylvester equation

$$A\boldsymbol{U} + \boldsymbol{U}B = G$$

Various settings:

- Small A and small B: Bartels-Stewart algorithm
  1. Compute the Schur forms:
     $A^* = URU^*$, $B = VSV^*$ with R, S upper triangular;
  2. Solve $R^*\boldsymbol{Y} + \boldsymbol{Y}S = U^*GV$ for $\boldsymbol{Y}$ (element-wise);
  3. Compute $\boldsymbol{U} = U\boldsymbol{Y}V^*$
- Large A and small B: Column decoupling
  1. Compute the decomposition $B = WSW^{-1}$, $S = \mathrm{diag}(s_1, \ldots, s_m)$
  2. Set $\widehat{G} = GW$
  3. For $i = 1, \ldots, m$ solve $(A + s_i I)(\widehat{\boldsymbol{U}})_i = (\widehat{G})_i$
  4. Compute $\boldsymbol{U} = \widehat{\boldsymbol{U}}W^{-1}$

Numerical solution of the Sylvester equation

$$A\boldsymbol{U} + \boldsymbol{U}B = G$$

Various settings:

- Small A and small B: Bartels-Stewart algorithm
  1. Compute the Schur forms:
     $A^* = URU^*$, $B = VSV^*$ with R, S upper triangular;
  2. Solve $R^*\boldsymbol{Y} + \boldsymbol{Y}S = U^*GV$ for $\boldsymbol{Y}$ (element-wise);
  3. Compute $\boldsymbol{U} = U\boldsymbol{Y}V^*$.
- Large A and small B: Column decoupling
  1. Compute the decomposition $B = WSW^{-1}$, $S = \mathrm{diag}(s_1, \ldots, s_m)$
  2. Set $\widehat{G} = GW$
  3. For $i = 1, \ldots, m$ solve $(A + s_iI)(\widehat{\boldsymbol{U}})_i = (\widehat{G})_i$
  4. Compute $\boldsymbol{U} = \widehat{\boldsymbol{U}}W^{-1}$
- Large A and large B: Iterative solution (G low rank, or G sparse)

Numerical solution of large scale Sylvester equations

$$A\boldsymbol{U} + \boldsymbol{U}B = G$$

with G low rank

- Projection methods
- ADI (Alternating Direction Iteration)
- Data sparse approaches (structure-dependent)

Projection methods

Seek $\boldsymbol{U}_k \approx \boldsymbol{U}$ of low rank:

$$\boldsymbol{U}_k = \begin{bmatrix} \\ \boldsymbol{U}_k^{(1)} \\ \\ \end{bmatrix} [\ (\boldsymbol{U}_k^{(2)})^\top\ ]$$

with $\boldsymbol{U}_k^{(1)}, \boldsymbol{U}_k^{(2)}$ tall

Index k "related" to the approximation rank

See, Simoncini, SIREV 2016.

Multiterm linear matrix equation

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

$A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{m\times m}$, $\boldsymbol{X}$ unknown matrix

Possibly large dimensions, structured coefficient matrices

The problem in its full generality is far from tractable, although the transformation to a matrix-vector equation [...] allows us to use the considerable arsenal of numerical weapons currently available for the solution of such problems.

Peter Lancaster, SIAM Rev. 1970

Multiterm linear matrix equation. Classical device

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

Kronecker formulation $\boxed{\left(B_1^\top \otimes A_1 + \ldots + B_\ell^\top \otimes A_\ell\right) \boldsymbol{x} = c \;\Leftrightarrow\; \mathcal{A}\boldsymbol{x} = c}$

Iterative methods: matrix-matrix multiplications and rank truncation

(Benner, Breiten, Bouhamidi, Chehab, Damm, Grasedyck, Jbilou, Kressner, Matthies, Nagy, Onwunta, Raydan, Stoll, Tobler, Wedderburn, Zander, ...)

Kronecker product : $M \otimes P = \begin{bmatrix} m_{11}P & \ldots & m_{1n}P \\ \vdots & \ddots & \vdots \\ m_{n1}P & \ldots & m_{nn}P \end{bmatrix}$ and $\text{vec}(AXB) = (B^\top \otimes A)\text{vec}(X)$

Multiterm linear matrix equation. Classical device

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

Kronecker formulation $\boxed{\left(B_1^\top \otimes A_1 + \ldots + B_\ell^\top \otimes A_\ell\right) \boldsymbol{x} = c \;\Leftrightarrow\; \mathcal{A}\boldsymbol{x} = c}$

Iterative methods: matrix-matrix multiplications and rank truncation

(Benner, Breiten, Bouhamidi, Chehab, Damm, Grasedyck, Jbilou, Kressner, Matthies, Nagy, Onwunta, Raydan, Stoll, Tobler, Wedderburn, Zander, ...)

Kronecker product : $M \otimes P = \begin{bmatrix} m_{11}P & \ldots & m_{1n}P \\ \vdots & \ddots & \vdots \\ m_{n1}P & \ldots & m_{nn}P \end{bmatrix}$ and $\mathrm{vec}(AXB) = (B^\top \otimes A)\mathrm{vec}(X)$

Alternatives to Kronecker form:

- Fixed point iterations (an "evergreen"...)
- Projection-type methods ⇒ low rank approximation
- Ad-hoc problem-dependent procedures
- etc.

Current very active area of research

Truncated matrix-oriented CG for Kronecker form

Input: $\mathcal{A}(\boldsymbol{X}) = A_1\boldsymbol{X}B_1 + A_2\boldsymbol{X}B_2 + \ldots + A_\ell\boldsymbol{X}B_\ell$, right-hand side $C \in \mathbb{R}^{n\times n}$ in low-rank format. Truncation operator $\mathcal{T}$.

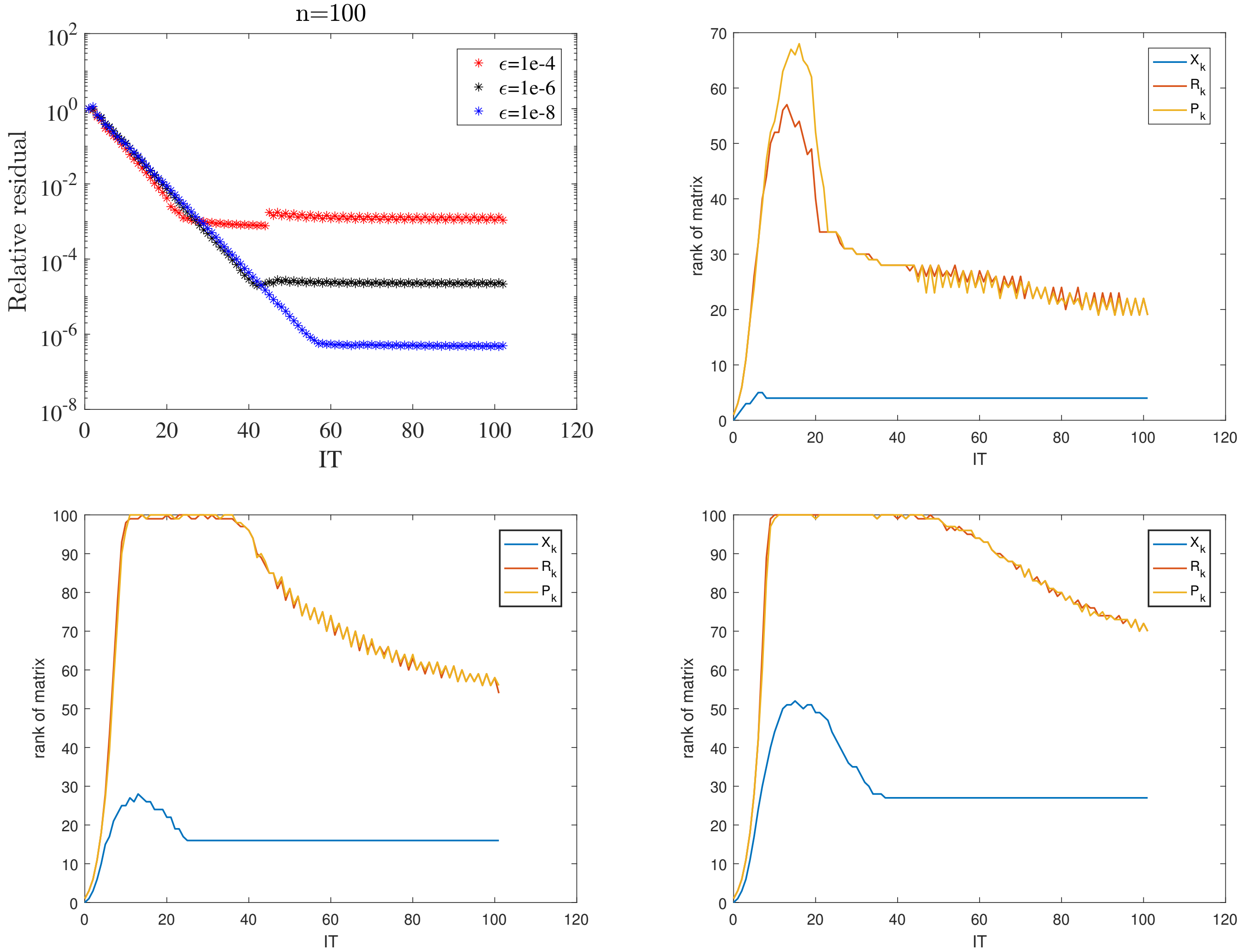
Output: Matrix $X \in \mathbb{R}^{n\times n}$ in low-rank format s.t. $||\mathcal{A}(X) - C||_F/||C||_F \le tol$.

$X_0 = 0$, $R_0 = C$, $P_0 = R_0$, $Q_0 = \mathcal{A}(P_0)$
$\xi_0 = \langle P_0, Q_0\rangle$, $k = 0$ $\qquad\qquad \langle X, Y\rangle = \mathrm{tr}(X^\top Y)$
while $||R_k||_F > tol$ **do**
  $\omega_k = \langle R_k, P_k\rangle/\xi_k$
  $X_{k+1} = X_k + \omega_k P_k$, $\qquad X_{k+1} \leftarrow \mathcal{T}(X_{k+1})$
  $R_{k+1} = C - \mathcal{A}(X_{k+1})$, $\qquad$ Optionally: $R_{k+1} \leftarrow \mathcal{T}(R_{k+1})$
  $\beta_k = -\langle R_{k+1}, Q_k\rangle/\xi_k$
  $P_{k+1} = R_{k+1} + \beta_k P_k$, $\qquad P_{k+1} \leftarrow \mathcal{T}(P_{k+1})$
  $Q_{k+1} = \mathcal{A}(P_{k+1})$, $\qquad$ Optionally: $Q_{k+1} \leftarrow \mathcal{T}(Q_{k+1})$
  $\xi_{k+1} = \langle P_{k+1}, Q_{k+1}\rangle$
  $k = k + 1$
end while
$X = X_k$

♣ Iterates kept in factored form! $\qquad$ Kressner and Tobler, 2011

Threshold based truncated CG. $n = 100$, tol$=\epsilon \in \{10^{-4}, 10^{-6}, 10^{-8}\}$

$A = \frac{1}{h^2}$tridiag$(-1, \underline{2}, -1)$, $M =$ diag(a_1), $N =$ diag(a_2), a_1 and a_2 random vectors



Projection-type methods. 1

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

Given approximation spaces $\mathcal{K}_A$, $\mathcal{K}_B$,

$$\boldsymbol{X} \approx X_m \qquad \text{with} \qquad \text{vec}(X_m) \in \mathcal{K}_B \otimes \mathcal{K}_A$$

Projection-type methods. 1

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

Given approximation spaces $\mathcal{K}_A$, $\mathcal{K}_B$,

$$\boldsymbol{X} \approx X_m \qquad \text{with} \qquad \text{vec}(X_m) \in \mathcal{K}_B \otimes \mathcal{K}_A$$

$\boldsymbol{X}$ is approximated by a low rank matrix !

that is, $X_m := V_m Y_m W_m^\top$, $\qquad \mathcal{K}_A = \text{Range}(V_m),\ \mathcal{K}_B = \text{Range}(W_m)$

Galerkin condition:

$R := A_1 X_m B_1 + A_2 X_m B_2 + \ldots + A_\ell X_m B_\ell - C \quad \perp \quad \mathcal{K}_B \otimes \mathcal{K}_A$

$$V_m^\top R W_m = 0$$

Projection-type methods. 1

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

Given approximation spaces $\mathcal{K}_A$, $\mathcal{K}_B$,

$$\boldsymbol{X} \approx X_m \qquad \text{with} \qquad \text{vec}(X_m) \in \mathcal{K}_B \otimes \mathcal{K}_A$$

$\boldsymbol{X}$ is approximated by a low rank matrix !

that is, $X_m := V_m Y_m W_m^\top$, $\qquad \mathcal{K}_A = \text{Range}(V_m), \; \mathcal{K}_B = \text{Range}(W_m)$

Galerkin condition:

$R := A_1 X_m B_1 + A_2 X_m B_2 + \ldots + A_\ell X_m B_\ell - C \quad \perp \quad \mathcal{K}_B \otimes \mathcal{K}_A$

$$V_m^\top R W_m = 0$$

Projected matrix equation:

$$V_m^\top (A_1 X_m B_1 + \ldots + A_\ell X_m B_\ell - C) W_m = 0$$

$$(V_m^\top A_1 V_m) \boldsymbol{Y} (W_m^\top B_1 W_m) + \ldots + (V_m^\top A_\ell V_m) \boldsymbol{Y} (W_m^\top B_\ell W_m) - V_m^\top C W_m = 0$$

Projection-type methods. 2

Solve for $\boldsymbol{Y}$:

$$(V_m^\top A_1 V_m)\boldsymbol{Y}(W_m^\top B_1 W_m) + \ldots + (V_m^\top A_\ell V_m)\boldsymbol{Y}(W_m^\top B_\ell W_m) - V_m^\top C W_m = 0$$

Then, implicitly generate $\quad X_m := V_m \boldsymbol{Y}_m W_m^\top$

Procedure generalizes the case $\ell = 2$, using the classical *Galerkin projection* methodology

Projection-type methods. 2

Solve for $\boldsymbol{Y}$:

$$(V_m^\top A_1 V_m)\boldsymbol{Y}(W_m^\top B_1 W_m) + \ldots + (V_m^\top A_\ell V_m)\boldsymbol{Y}(W_m^\top B_\ell W_m) - V_m^\top C W_m = 0$$

Then, implicitly generate $\quad X_m := V_m \boldsymbol{Y}_m W_m^\top$

Procedure generalizes the case $\ell = 2$, using the classical *Galerkin projection* methodology

Optimality property: Palitta and Simoncini, 2020

$$\|X_\star - X_m\|_{\mathcal{A}} = \min_{\substack{Z = V_m Y W_m^\top \\ Y \in \mathbb{R}^{m\times m}}} \|X_\star - Z\|_{\mathcal{A}},$$

where $\|X\|_{\mathcal{A}}^2 = \mathrm{trace}\left(\sum_{j=1}^{\ell} X^\top A_j X B_j\right)$.

Projection-type methods. 2

Solve for $\boldsymbol{Y}$:

$$(V_m^\top A_1 V_m)\boldsymbol{Y}(W_m^\top B_1 W_m) + \ldots + (V_m^\top A_\ell V_m)\boldsymbol{Y}(W_m^\top B_\ell W_m) - V_m^\top C W_m = 0$$

Then, implicitly generate $\quad X_m := V_m \boldsymbol{Y}_m W_m^\top$

Procedure generalizes the case $\ell = 2$, using the classical *Galerkin projection* methodology

Optimality property: Palitta and Simoncini, 2020

$$\|X_\star - X_m\|_{\mathcal{A}} = \min_{\substack{Z = V_m Y W_m^\top \\ Y \in \mathbb{R}^{m\times m}}} \|X_\star - Z\|_{\mathcal{A}},$$

where $\|X\|_{\mathcal{A}}^2 = \mathrm{trace}\left(\sum_{j=1}^{\ell} X^\top A_j X B_j\right)$.

Crucial issues for effectiveness:

- Choice of spaces $\mathcal{K}_A, \mathcal{K}_B$ and their construction. Ideally,

$$\mathrm{range}(V_m) \subseteq \mathrm{range}(V_{m+1}), \qquad \mathrm{range}(W_m) \subseteq \mathrm{range}(W_{m+1})$$

- Solution of the reduced multiterm equation

A "simple" example

$$AX + XA + MXM = ff^\top, \qquad A, M \,\text{spd},\; f \,\text{vector}$$

♣ No available *direct* methods for the generic case, except Kronecker form

A “simple” example

$$A\boldsymbol{X} + \boldsymbol{X}A + M\boldsymbol{X}M = ff^\top, \qquad A, M \,\text{spd},\; f \,\text{vector}$$

♣ No available *direct* methods for the generic case, except Kronecker form

Matrix-oriented CG: $X^{(k)} = X_1^{(k)} G^{(k)} (X_1^{(k)})^\top$

$\text{range}(X_1^{(k)}) \subset \mathbb{Q}_k = \text{span}\{f, Af, Mf, A^2f, AMf, MAf, M^2f, ...\}, \dim(\mathbb{Q}_{k+1}) \le \dim(\mathbb{Q}_k) + 2^k$

A "simple" example

$$A\boldsymbol{X} + \boldsymbol{X}A + M\boldsymbol{X}M = ff^\top, \qquad A, M \,\text{spd}, \; f \,\text{vector}$$

♣ No available *direct* methods for the generic case, except Kronecker form

Matrix-oriented CG: $\qquad X^{(k)} = X_1^{(k)} G^{(k)} (X_1^{(k)})^\top$

$\text{range}(X_1^{(k)}) \subset \mathbb{Q}_k = \text{span}\{f, Af, Mf, A^2 f, AMf, MAf, M^2 f, ...\}, \dim(\mathbb{Q}_{k+1}) \leq \dim(\mathbb{Q}_k) + 2^k$

Galerkin method: Choose $\mathcal{K}_m = \text{range}(V_m)$ with

$$V_0 = f =: \underline{v_1} \qquad V_1 = [v_1, Av_1, Mv_1] =: [v_1, \underline{v_2}, v_3]$$

$$V_2 = [V_1, Av_2, Mv_2] =: [v_1, v_2, \underline{v_3}, v_4, v_5]$$

$$V_3 = [V_2, Av_3, Mv_3] =: [v_1, v_2, v_3, \underline{v_4}, v_5, v_6, v_7]$$

etc.

$\Rightarrow \mathcal{K}_m = \text{span}\{f, Af, Mf, A^2 f, AMf, MAf, M^2 f, ...\}, \; \dim(\mathcal{K}_m) = 2m + 1$

$$\boxed{\mathbb{Q}_k = \text{range}(V_{2^{k-1}})}$$

Hao and Simoncini, work in progress

Computational methods for certain structured problems

A particular case[a]:

$$A\boldsymbol{X} + \boldsymbol{X}A^\top + M_1\boldsymbol{X}M_1 + \ldots + M_\ell\boldsymbol{X}M_\ell = F,$$

with $A \in \mathbb{R}^{n\times n}$, M_i**s with very low rank** s_i**,** $M_i = U_iV_i^\top$

[a]In fact, terms in the form $M_i\boldsymbol{X}N_i$ can also be treated

Computational methods for certain structured problems

A particular case[a]:

$$A\boldsymbol{X} + \boldsymbol{X}A^\top + M_1\boldsymbol{X}M_1 + \ldots + M_\ell \boldsymbol{X} M_\ell = F,$$

with $A \in \mathbb{R}^{n\times n}$, M_i**s with very low rank** s_i**,** $M_i = U_i V_i^\top$

Using the Kronecker form ($\ell = 1$):

$$(A \otimes I + I \otimes A + (U_1 \otimes U_1)(V_1 \otimes V_1)^\top)\boldsymbol{x} = f$$

that is

$$(\mathcal{A} + \mathcal{U}\mathcal{V}^\top)\boldsymbol{x} = f$$

with $\mathcal{U} = U_1 \otimes U_1$, $\mathcal{V} = V_1 \otimes V_1$ again of low rank s_1^2

[a] In fact, terms in the form $M_i\boldsymbol{X}N_i$ can also be treated

Computational methods for certain structured problems

A particular case[a]:

$$AX + XA^\top + M_1 X M_1 + \ldots + M_\ell X M_\ell = F,$$

with $A \in \mathbb{R}^{n\times n}$, M_i**s with very low rank** s_i**,** $M_i = U_i V_i^\top$

Using the Kronecker form ($\ell = 1$):

$$(A \otimes I + I \otimes A + (U_1 \otimes U_1)(V_1 \otimes V_1)^\top)\boldsymbol{x} = f$$

that is

$$(\mathcal{A} + \mathcal{U}\mathcal{V}^\top)\boldsymbol{x} = f$$

with $\mathcal{U} = U_1 \otimes U_1$, $\mathcal{V} = V_1 \otimes V_1$ again of low rank s_1^2

Solution method: Sherman-Morrison-Woodbury formula

$$\boldsymbol{x} = (\mathcal{A} + \mathcal{U}\mathcal{V}^\top)^{-1} f = \mathcal{A}^{-1} f - \mathcal{A}^{-1}\mathcal{U}(I + \mathcal{V}^\top \mathcal{A}^{-1}\mathcal{U})^{-1}\mathcal{V}^\top \mathcal{A}^{-1} f$$

---

[a] In fact, terms in the form $M_i X N_i$ can also be treated

Matrix-oriented Sherman-Morrison-Woodbury formula

$$\boldsymbol{x} = \mathcal{A}^{-1} f - \mathcal{A}^{-1}\mathcal{U}(I + \mathcal{V}^\top \mathcal{A}^{-1}\mathcal{U})^{-1}\mathcal{V}^\top \mathcal{A}^{-1} f$$

1. Solve $\mathcal{A}w = f$
2. Solve $\mathcal{A}\mathtt{p}_j = \mathtt{u}_j$ where $\mathcal{U} = [\mathtt{u}_1, \ldots, \mathtt{u}_{s^2}]$ to give $\mathcal{P} = [\mathtt{p}_1, \ldots, \mathtt{p}_{s^2}]$;
3. Compute $H = I + \mathcal{V}^\top \mathcal{P} \in \mathbb{R}^{s^2 \times s^2}$
4. Solve $Hg = \mathcal{V}^\top w$
5. Compute $x = w - \mathcal{P}g$.

Matrix-oriented Sherman-Morrison-Woodbury formula

$$\boldsymbol{x} = \mathcal{A}^{-1} f - \mathcal{A}^{-1}\mathcal{U}(I + \mathcal{V}^\top \mathcal{A}^{-1}\mathcal{U})^{-1}\mathcal{V}^\top \mathcal{A}^{-1} f$$

1. Solve $\mathcal{A}w = f$
2. Solve $\mathcal{A}\mathrm{p}_j = \mathrm{u}_j$ where $\mathcal{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_{s^2}]$ to give $\mathcal{P} = [\mathrm{p}_1, \ldots, \mathrm{p}_{s^2}]$;
3. Compute $H = I + \mathcal{V}^\top \mathcal{P} \in \mathbb{R}^{s^2 \times s^2}$
4. Solve $Hg = \mathcal{V}^\top w$
5. Compute $x = w - \mathcal{P}g$.

Steps 1. and 2.:

$$w = \mathcal{A}^{-1} f \quad \Leftrightarrow \quad AW + WA^\top = F, \quad f = \mathrm{vec}(F)$$

Analogously for each $\mathrm{p}_j = \mathrm{vec}(P_j)$ in step 2

$AW + WA^\top = P_j$ Lyapunov equations, with the same A - cheap "direct" solution

Matrix-oriented Sherman-Morrison-Woodbury formula

$$\boldsymbol{x} = \mathcal{A}^{-1}f - \mathcal{A}^{-1}\mathcal{U}(I + \mathcal{V}^\top\mathcal{A}^{-1}\mathcal{U})^{-1}\mathcal{V}^\top\mathcal{A}^{-1}f$$

1. Solve $\mathcal{A}w = f$
2. Solve $\mathcal{A}\mathtt{p}_j = \mathtt{u}_j$ where $\mathcal{U} = [\mathtt{u}_1, \ldots, \mathtt{u}_{s^2}]$ to give $\mathcal{P} = [\mathtt{p}_1, \ldots, \mathtt{p}_{s^2}]$;
3. Compute $H = I + \mathcal{V}^\top\mathcal{P} \in \mathbb{R}^{s^2 \times s^2}$
4. Solve $Hg = \mathcal{V}^\top w$
5. Compute $x = w - \mathcal{P}g$.

Steps 1. and 2.:

$$w = \mathcal{A}^{-1}f \quad \Leftrightarrow \quad AW + WA^\top = F, \quad f = \mathrm{vec}(F)$$

Analogously for each $\mathtt{p}_j = \mathrm{vec}(P_j)$ in step 2

$AW + WA^\top = P_j$ Lyapunov equations, with the same A - cheap “direct” solution

Step 3.

$$\mathtt{v}_j^\top\mathcal{A}^{-1}\mathtt{u}_t = v_i^\top P_t v_k, \quad j = (k-1)s + i$$

Analogously for $\mathcal{V}^\top w$ in step 4

A numerical example

Let $X_\star$ be a ref. soln (uniformly distr.random), and rhs computed explicitly

We monitor: $Err := \frac{\|X - X_\star\|_F}{\|X_\star\|_F}$

		Matrix form		Vector Form	
n	s_1/s_2	CPU time	Err	CPU time	Err
40	$3/5$	0.013	3.81e-11	0.195	2.29e-10
	$6/10$	0.017	9.05e-10	0.657	4.98e-10
	$12/20$	0.035	5.25e-09	2.333	1.35e-08
80	$3/5$	0.022	2.15e-10	5.283	1.22e-09
	$6/10$	0.033	8.38e-09	15.408	1.84e-08
	$12/20$	0.074	2.50e-08	56.347	3.46e-08
160	$3/5$	0.043	1.29e-09	129.957	6.89e-09
	$6/10$	0.070	1.10e-08	281.946	2.69e-08
	$12/20$	0.220	2.90e-07	1030.242	1.20e-06

Table 1: Symmetric and dense matrix A and U_1, U_2 ($\ell = 2$) for various ranks s_1, s_2

Hao and Simoncini, 2021. See also Damm, 2008, Massei etal 2018.

Conclusions

- Rich setting for new algorithmic strategies
- Certain approaches appropriate for solving linear *tensor* equations
- Devise more general "direct" solvers, to be used (also) in the projection phase!

Visit: `www.dm.unibo.it/~simoncin`

Email address: `valeria.simoncini@unibo.it`

REFERENCES

1. Yue Hao and V. S., *Matrix equation solving of PDEs in polygonal domains using conformal mappings.* Journal of Numerical Mathematics, vol. 29, no. 3, 2021

2. Yue Hao and V. S., *The Sherman-Morrison-Woodbury formula for generalized linear matrix equations and applications*, Numer. Linear Algebra w/Appl. 28 (5), 2021

3. Catherine E. Powell, David Silvester and V. S., *An efficient reduced basis solver for stochastic Galerkin matrix equations*, SIAM J. Scientific Computing, 39 (1), (2017).

4. Davide Palitta and V. S., *Matrix-equation-based strategies for convection-diffusion equations*, BIT Numerical Mathematics, 56-2, (2016).

5. V.S., *Computational methods for linear matrix equations*,SIAM Review, 58-3(2016)