Solving multiterm linear matrix equations

order reduction strategies and applications

Valeria Simoncini

Dipartimento di Matematica
Alma Mater Studiorum - Università di Bologna
valeria.simoncini@unibo.it

The matrix equation problem

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

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

The matrix equation problem

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

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

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

Iterative methods: matrix-matrix multiplications and rank truncation
(Benner, Breiten, Bouhamidi, Chehab, Damm, Grasedyck, Jbilou, Kressner, Matthies, Nagy, Onwunta, Raydan, Stoll, Tobler, Wedderburn, Zander, ...)

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

Applications:

- Control
- Deterministic and stochastic, and time dependent PDEs
- Inverse problems and optimization

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

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

Iterative methods: matrix-matrix multiplications and rank truncation
(Benner, Breiten, Bouhamidi, Chehab, Damm, Grasedyck, Jbilou, Kressner, Matthies, Nagy, Onwunta, Raydan, Stoll, Tobler, Wedderburn, Zander, ...)

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

Applications:

- Control
- Deterministic and stochastic, and time dependent PDEs
- Inverse problems and optimization

Multiterm linear matrix equation

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

Alternative approaches to the Kronecker form:

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

A sample of these methodologies on different problems:

- ♣ Stochastic PDE
- ♣ PDEs on polygonal domains
- ♣ All-at-once PDE-constrained optimization problem
- ♣ Bilinear control problems
- ♣ ....

Multiterm linear matrix equation

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

Alternative approaches to the Kronecker form:

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

A sample of these methodologies on different problems:

- Stochastic PDE
- PDEs on polygonal domains
- All-at-once PDE-constrained optimization problem
- Bilinear control problems
- ....

PDEs with random inputs

Stochastic steady-state diffusion eqn: *Find* $u : D \times \Omega \to \mathbb{R}$ *s.t.* $\mathbb{P}$-*a.s.*,

$$\begin{cases} -\nabla \cdot (a(\boldsymbol{x},\omega)\nabla u(\boldsymbol{x},\omega)) = & f(\boldsymbol{x}) \ \text{in D} \\ u(\boldsymbol{x},\omega) = & 0 \quad \text{on } \partial\text{D} \end{cases}$$

f: deterministic;
a: random field, linear function of finite no. of real-valued random variables $\xi_r : \Omega \to \Gamma_r \subset \mathbb{R}$

Common choice: truncated Karhunen–Loève (KL) expansion,

$$a(\boldsymbol{x},\omega) = \mu(\boldsymbol{x}) + \sigma \sum_{r=1}^{\ell} \sqrt{\lambda_r}\phi_r(\boldsymbol{x})\xi_r(\omega),$$

$\mu(\boldsymbol{x})$: expected value of diffusion coef. σ: std dev.
$(\lambda_r, \phi_r(\boldsymbol{x}))$ eigs of the integral operator $\mathcal{V}$ wrto $V(\boldsymbol{x},\boldsymbol{x}') = \frac{1}{\sigma^2}C(\boldsymbol{x},\boldsymbol{x}')$
($\lambda \searrow$ $C : D \times D \to \mathbb{R}$ covariance fun.)

PDEs with random inputs

Stochastic steady-state diffusion eqn: *Find* $u : D \times \Omega \to \mathbb{R}$ *s.t.* $\mathbb{P}$*-a.s.,*

$$\begin{cases} -\nabla \cdot (a(\boldsymbol{x},\omega)\nabla u(\boldsymbol{x},\omega)) = & f(\boldsymbol{x}) \ \text{in D} \\ u(\boldsymbol{x},\omega) = & 0 \quad \text{on } \partial\text{D} \end{cases}$$

f: deterministic;
a: random field, linear function of finite no. of real-valued random variables
$\xi_r : \Omega \to \Gamma_r \subset \mathbb{R}$

Common choice: truncated Karhunen–Loève (KL) expansion,

$$a(\boldsymbol{x},\omega) = \mu(\boldsymbol{x}) + \sigma \sum_{r=1}^{\ell} \sqrt{\lambda_r}\phi_r(\boldsymbol{x})\xi_r(\omega),$$

$\mu(\boldsymbol{x})$: expected value of diffusion coef. σ: std dev.
$(\lambda_r, \phi_r(\boldsymbol{x}))$ eigs of the integral operator $\mathcal{V}$ wrto $V(\boldsymbol{x},\boldsymbol{x}') = \frac{1}{\sigma^2}C(\boldsymbol{x},\boldsymbol{x}')$
($\lambda \searrow$ $C : D \times D \to \mathbb{R}$ covariance fun.)

Stochastic Galerkin discretization. The SPDE-practitioner approach.

Approx with space in tensor product form[1] $\mathcal{X}_h \times S_p$

$$\mathcal{A}\boldsymbol{x} = \boldsymbol{c}, \qquad \mathcal{A} = G_0 \otimes K_0 + \sum_{r=1}^{\ell} G_r \otimes K_r, \quad \boldsymbol{b} = \boldsymbol{g}_0 \otimes \boldsymbol{f}_0,$$

$\boldsymbol{x}$: expansion coef. of approx to u in the tensor product basis $\{\varphi_i \psi_k\}$
$K_r \in \mathbb{R}^{n_x \times n_x}$, FE matrices (sym)
$G_r \in \mathbb{R}^{n_\xi \times n_\xi}$, $r = 0, 1, \ldots, m$ Galerkin matrices associated w/ S_p (sym.)
$\boldsymbol{g}_0$: first column of G_0
$\boldsymbol{f}_0$: FE rhs of deterministic PDE

$$n_\xi = \dim(S_p) = \frac{(\ell + p)!}{\ell! p!} \qquad \Rightarrow \boxed{n_x \cdot n_\xi} \text{ huge}$$

[1] S_p set of multivariate polyn of total degree $\leq p$

The matrix equation formulation

$$(G_0 \otimes K_0 + G_1 \otimes K_1 + \ldots + G_\ell \otimes K_\ell)\,\boldsymbol{x} = \boldsymbol{g}_0 \otimes \boldsymbol{f}_0$$

transforms into

$$K_0 \mathbf{X} G_0 + K_1 \mathbf{X} G_1 + \ldots + K_\ell \mathbf{X} G_\ell = F, \qquad F = \boldsymbol{f}_0 \boldsymbol{g}_0^\top$$

$(G_0 = I)$

Solution strategy. Conjecture:

- $\{K_r\}$ from trunc'd Karhunen–Loève (KL) expansion

$$\Downarrow$$

$$\mathbf{X} \approx \widetilde{X} \text{ low rank, } \widetilde{X} = X_1 X_2^\top$$

Matrix Galerkin approximation of the deterministic part

Approximation space $\mathcal{K}_k$ and basis matrix V_k: $\quad \mathbf{X} \approx X_k = V_k Y$

$$V_k^\top R_k = 0, \qquad R_k := K_0 X_k + K_1 X_k G_1 + \ldots + K_\ell X_k G_\ell - \boldsymbol{f}_0 \boldsymbol{g}_0^\top$$

Computational challenges:

- Generation of $\mathcal{K}_k$ involved $\ell + 1$ different matrices $\{K_r\}$!
- Matrices K_r have different spectral properties
- n_x, n_ξ so large that X_k, R_k should not be formed !

(Powell & Silvester & Simoncini 2017)

Example. $-\nabla \cdot (a\nabla u) = 1, \quad D = (-1,1)^2$. KL expansion

$\mu = 1$, $\xi_r \sim U(-\sqrt{3}, \sqrt{3})$ and $C(\vec{x}_1, \vec{x}_2) = \sigma^2 \exp\left(-\frac{\|\vec{x}_1 - \vec{x}_2\|_1}{2}\right)$, $n_x = 65,025$, $\sigma = 0.3$

ℓ	p	n_ξ	k	inner its	n_k $\mathcal{K}_k$	rank $\widetilde{\boldsymbol{X}}$	time secs	CG time (its)
8 87%	2	45	17	9.8	128	45	32.1	13.4 (8)
	3	165	21	12.2	160	129	41.4	56.6 (10)
	4	495	24	14.5	183	178	51.1	197.0 (12)
	5	1,287	27	16.9	207	207	64.0	553.0 (13)
12 89%	2	91	15	9.9	165	89	47.8	30.0 (8)
	3	455	18	12.2	201	196	61.6	175.0 (10)
	4	1,820	21	15.0	236	236	86.4	821.0 (12)
	5	6,188	25	18.6	281	281	188.0	3070.0 (13)
20 93%	2	231	16	9.4	281	206	111.0	94.7 (8)
	3	1,771	23	12.3	399	399	197.0	845.0 (10)
	4	10,626	26	15.4	454	454	556.0	Out of Mem

% of variance integral of a

Linear matrix equations for convection-diffusion PDEs

$$-\alpha_1 u_{xx} - \alpha_2 u_{yy} + \boldsymbol{w} \cdot \nabla u + \beta u = f, \quad (x, y) \in \Omega,$$

$\Omega \subset \mathbb{R}^2$ sufficiently regular domains, e.g., polygons, not necessarily convex

Standard procedures giving $\mathcal{A}\mathbf{u} = f$:

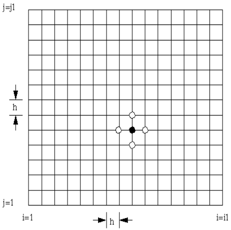
- Large class of finite element methods
- Spectral (element) methods
- Isogeometric Analysis
- Mimetic FD or Virtual element methods
- Classical Finite differences
- ...

Proof of concept:

Explore venues leading to linear matrix equations

The Poisson equation in a square

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



Usual lexicographic ordering $\Rightarrow \qquad Au = b$

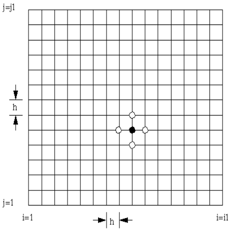
Discretization: $U_{i,j} \approx u_{x_i,y_j}$, with (x_i, y_j) interior nodes, so that

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

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

The Poisson equation in a square

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



Usual lexicographic ordering $\Rightarrow$ $\qquad Au = b$

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

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

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

The Poisson equation - matrix formulation

Let $T = \frac{1}{h^2}\text{tridiag}(-1, \underline{2}, -1)$

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

Collecting all nodes together,

$$-u_{xx} \approx TU, \qquad -u_{yy} \approx UT$$

Therefore, directly from the grid,

$$-u_{xx} - u_{yy} = f \qquad \Rightarrow \qquad TU + UT = F, \qquad F_{ij} = f(x_i, y_j)$$

Convection-diffusion eqns in a rectangle (with D. Palitta)

$$-\varepsilon\Delta u + \phi_1(x)\psi_1(y)u_x + \phi_2(x)\psi_2(y)u_y + \gamma_1(x)\gamma_2(y)u = f$$

$(x, y) \in \Omega \subset \mathbb{R}^2$, ϕ_i, ψ_i, γ_i, $i = 1, 2$ sufficiently regular func's + b.c.

Problem discretization by means of a tensor basis

Multiterm linear matrix equation:

$$-\varepsilon T_1\mathbf{U} - \varepsilon\mathbf{U}T_2 + \Phi_1 B_1\mathbf{U}\Psi_1 + \Phi_2\mathbf{U}B_2^\top\Psi_2 + \Gamma_1\mathbf{U}\Gamma_2 = F$$

Finite Diff.: $\mathbf{U}_{i,j} = \mathbf{U}(x_i, y_j)$ approximate solution at the nodes

but also Isogeometric Analysis (IGA), certain spectral methods, etc.

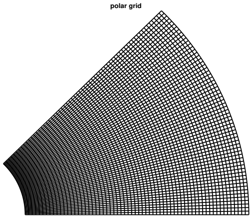
... A classical approach, Bickley & McNamee, 1960, Wachspress, 1963
(Early literature on difference equations)

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:

$$\boxed{\Phi^2 T \widetilde{\mathbf{U}} + \widetilde{\mathbf{U}} T - \Phi B \widetilde{\mathbf{U}} = \widetilde{F}}$$

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

$$-\hat{u}_{\rho\rho} - \hat{u}_{\theta\theta} = \hat{f}, \qquad (r,\theta) \in [r_0, r_1] \times [0, \frac{\pi}{4}]$$

Matrix equation after mapping to the rectangle:

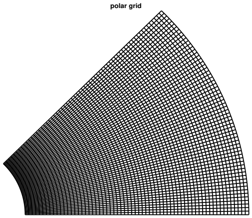
$$\boxed{T\widehat{\mathbf{U}} + +\widehat{\mathbf{U}} T = \widehat{F}}$$

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 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:

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

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

$$-\hat{u}_{\rho\rho} - \hat{u}_{\theta\theta} = \hat{f}, \qquad (r,\theta) \in [r_0, r_1] \times [0, \frac{\pi}{4}]$$

Matrix equation after mapping to the rectangle:

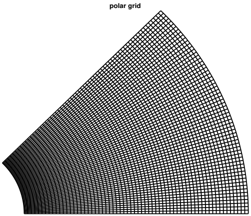
$$\boxed{T \widehat{\boldsymbol{U}} + + \widehat{\boldsymbol{U}} T = \widehat{F}}$$

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 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:

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

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

$$-\hat{u}_{\rho\rho} - \hat{u}_{\theta\theta} = \hat{f}, \qquad (r, \theta) \in [r_0, r_1] \times [0, \frac{\pi}{4}]$$

Matrix equation after mapping to the rectangle:

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

Discretization of polygons with > 4 edges (with Y. Hao)

We need an automatic procedure to map a polygon into a rectangle

Schwarz-Christoffel conformal mappings

$\{z_1, \ldots, z_n\}$: polygon vertices $\{\phi_1\pi, \ldots, \phi_n\pi\}$: vertices interior angles
Pre-images of the vertices (or pre-vertices): $\omega_1, \ldots, \omega_n \in \mathbb{R}$, with

$$\omega_1 < \omega_2 < \cdots < \omega_n = \infty.$$

Schwarz-Christoffel (SC) map g: $$g(\omega) = g(\omega_0) + c \int_{\omega_0}^{\omega} \prod_{j=1}^{n-1} (\zeta - \omega_j)^{\phi_j - 1} \, d\zeta \qquad (*)$$

Practical problems associated with Schwarz-Christoffel maps:

- *SC parameter problem*: determining the pre-vertices ω_j in closed form;
- *quadrature formulas*: Integrating the rhs of (*)

Schwarz-Christoffel Toolbox for Matlab, T. Driscoll, 1996, 2005

Discretization of polygons with > 4 edges (with Y. Hao)

We need an automatic procedure to map a polygon into a rectangle

Schwarz-Christoffel conformal mappings

$\{z_1, \ldots, z_n\}$: polygon vertices $\{\phi_1\pi, \ldots, \phi_n\pi\}$: vertices interior angles
Pre-images of the vertices (or pre-vertices): $\omega_1, ..., \omega_n \in \mathbb{R}$, with

$$\omega_1 < \omega_2 < \cdots < \omega_n = \infty.$$

Schwarz-Christoffel (SC) map g:
$$g(\omega) = g(\omega_0) + c \int_{\omega_0}^{\omega} \prod_{j=1}^{n-1} (\zeta - \omega_j)^{\phi_j - 1} \, d\zeta \qquad (*)$$

Practical problems associated with Schwarz-Christoffel maps:

- *SC parameter problem*: determining the pre-vertices ω_j in closed form;
- *quadrature formulas*: Integrating the rhs of (*)

Schwarz-Christoffel Toolbox for Matlab, T. Driscoll, 1996, 2005

Discretization of polygons with > 4 edges (with Y. Hao)

We need an automatic procedure to map a polygon into a rectangle

Schwarz-Christoffel conformal mappings

$\{z_1, \ldots, z_n\}$: polygon vertices $\{\phi_1\pi, \ldots, \phi_n\pi\}$: vertices interior angles
Pre-images of the vertices (or pre-vertices): $\omega_1, ..., \omega_n \in \mathbb{R}$, with

$$\omega_1 < \omega_2 < \cdots < \omega_n = \infty.$$

Schwarz-Christoffel (SC) map g:

$$g(\omega) = g(\omega_0) + c \int_{\omega_0}^{\omega} \prod_{j=1}^{n-1} (\zeta - \omega_j)^{\phi_j - 1} \, d\zeta \qquad (*)$$

Practical problems associated with Schwarz-Christoffel maps:

- *SC parameter problem*: determining the pre-vertices ω_j in closed form;
- *quadrature formulas*: Integrating the rhs of (*)

Schwarz-Christoffel Toolbox for Matlab, T. Driscoll, 1996, 2005

Discretization of polygons with > 4 edges (with Y. Hao)

We need an automatic procedure to map a polygon into a rectangle

Schwarz-Christoffel conformal mappings

$\{z_1, \dots, z_n\}$: polygon vertices $\{\phi_1\pi, \dots, \phi_n\pi\}$: vertices interior angles
Pre-images of the vertices (or pre-vertices): $\omega_1, \dots, \omega_n \in \mathbb{R}$, with

$$\omega_1 < \omega_2 < \cdots < \omega_n = \infty.$$

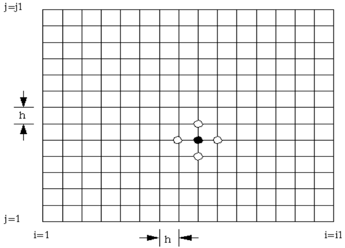
Schwarz-Christoffel (SC) map g:

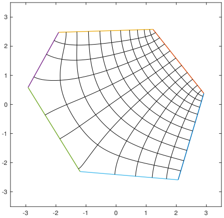
$$g(\omega) = g(\omega_0) + c \int_{\omega_0}^{\omega} \prod_{j=1}^{n-1} (\zeta - \omega_j)^{\phi_j - 1} \, d\zeta \qquad (*)$$

Practical problems associated with Schwarz-Christoffel maps:

- *SC parameter problem*: determining the pre-vertices ω_j in closed form;
- *quadrature formulas*: Integrating the rhs of (*)

Schwarz-Christoffel Toolbox for Matlab, T. Driscoll, 1996, 2005





Canonical rectangular domain Π → $z = g(\xi, \eta)$ → (Convex) physical domain Ω

$$z = g(\omega) = g(\xi + i\eta) = x(\xi, \eta) + iy(\xi, \eta)$$

Jacobian matrix of the conformal map g:

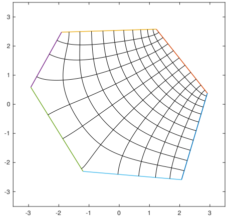
$$\mathcal{J} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \qquad \mathscr{J} = \mathscr{J}(\xi, \eta) = \det(\mathcal{J}) = x_\xi y_\eta - x_\eta y_\xi = x_\xi^2 + x_\eta^2 > 0$$

and $\widetilde{u}_\xi = u_x x_\xi + u_y y_\xi$, $\widetilde{u}_\eta = u_x x_\eta + u_y y_\eta$

Poisson equation in a polygon (with Y. Hao)

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

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



With finite diff. discretization:

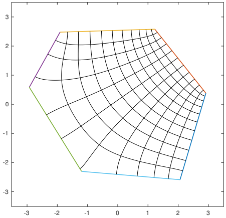
$$\boxed{T_1 U + U T_2 = F}, \qquad \widetilde{F} + b.c., \quad \text{and} \quad \widetilde{F}_{i,j} = (\mathscr{J}\widetilde{f})(\xi_i,\eta_j),\ 1 \le i \le n_1,\ 1 \le j \le n_2$$

Poisson equation is the ideal setting for SC mappings!

Poisson equation in a polygon (with Y. Hao)

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

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



With finite diff. discretization:

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

Poisson equation is the ideal setting for SC mappings!

Adding a reaction term

$$-\Delta u + \beta u = f, \quad (x,y) \in \Omega, \qquad u = 0 \quad \text{on } \partial\Omega.$$

In the canonical (reference) domain Π:

$$-\widetilde{u}_{\xi\xi} - \widetilde{u}_{\eta\eta} + (\mathscr{J}\widetilde{\beta})\widetilde{u} = \mathscr{J}\widetilde{f}, \quad (\xi,\eta) \in \Pi, \qquad \widetilde{u} = 0, \quad (\xi,\eta) \in \partial\Pi.$$

thus giving the following matrix equation:

$$\boxed{T_1 U + U T_2 + G \circ U = F}$$

($\circ$ denotes the (element-wise) Hadamard product)
with $G(i,j) := (\mathscr{J}\widetilde{\beta})(\xi_i, \eta_j)$, with $1 \le i \le n_1$ and $1 \le j \le n_2$

Adding a reaction term

$$-\Delta u + \beta u = f, \quad (x,y) \in \Omega, \qquad u = 0 \quad \text{on } \partial\Omega.$$

In the canonical (reference) domain Π:

$$-\widetilde{u}_{\xi\xi} - \widetilde{u}_{\eta\eta} + (\mathscr{J}\widetilde{\beta})\widetilde{u} = \mathscr{J}\widetilde{f}, \quad (\xi,\eta) \in \Pi, \qquad \widetilde{u} = 0, \quad (\xi,\eta) \in \partial\Pi.$$

thus giving the following matrix equation:

$$\boxed{T_1 U + U T_2 + G \circ U = F}$$

($\circ$ denotes the (element-wise) Hadamard product)
with $G(i,j) := (\mathscr{J}\widetilde{\beta})(\xi_i, \eta_j)$, with $1 \le i \le n_1$ and $1 \le j \le n_2$

A simple convection-diffusion problem

$$-u_{xx} - u_{yy} + \boldsymbol{w} \cdot \nabla u = f, \quad (x, y) \in \Omega$$

Transformed SC-mapped problem:

$$-\varepsilon(\widetilde{u}_{\xi\xi} + \widetilde{u}_{\eta\eta}) + x_\xi \widetilde{u}_\xi + x_\eta \widetilde{u}_\eta = \mathscr{J} \widetilde{f}, \quad (\xi, \eta) \in \Pi, \widetilde{u} = 0, \quad (\xi, \eta) \in \partial\Pi,$$

where we used $\boldsymbol{w} = (1, 0)$.

This yields the matrix equation

$$\boxed{\varepsilon T_1 U + \varepsilon U T_2 + X_\xi \circ (B_1 U) + X_\eta \circ (U B_2) = F,}$$

where $(X_\xi)_{i,j} = x_\xi(\xi_i, \eta_j)$, $(X_\eta)_{i,j} = x_\eta(\xi_i, \eta_j)$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$

Conclusions. 1

- Linear matrix equations can be obtained for general domains with different discretization procedures (IGA, FD, conformal mappings, ...)
- Structural properties should be exploited (different from Kronecker formulations)
- 3D case leads to linear **tensor** equations: a new research area

Solution strategies: 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 $\mathcal{K}_A$, $\mathcal{K}_B$,

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

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

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

$$V_m^\top R W_m = 0 \qquad \mathcal{K}_A = \mathrm{Range}(V_m), \mathcal{K}_B = \mathrm{Range}(W_m)$$

Let $X_m := V_m Y_m W_m^\top$.
Projected matrix equation:

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

Solution strategies: 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 !

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

$$V_m^\top R W_m = 0 \qquad \mathcal{K}_A = \text{Range}(V_m), \mathcal{K}_B = \text{Range}(W_m)$$

Let $X_m := V_m Y_m W_m^\top$.

Projected matrix equation:

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

Solution strategies: 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 !

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

$$V_m^\top R W_m = 0 \qquad \mathcal{K}_A = \text{Range}(V_m), \mathcal{K}_B = \text{Range}(W_m)$$

Let $X_m := V_m Y_m W_m^\top$.

Projected matrix equation:

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

Solution strategies: 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 $X_m := V_m Y_m W_m^\top$

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

Crucial issues for effectiveness:

- Choice of spaces $\mathcal{K}_A, \mathcal{K}_B$
- Generation of the two spaces $\mathcal{K}_A, \mathcal{K}_B$. 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

Solution strategies: 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 $X_m := V_m Y_m W_m^\top$

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

Crucial issues for effectiveness:

- Choice of spaces $\mathcal{K}_A, \mathcal{K}_B$
- Generation of the two spaces $\mathcal{K}_A, \mathcal{K}_B$. 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 particular case[2]:

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

with $A \in \mathbb{R}^{n\times n}$, **and the M_is having 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

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

[2] 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[2]:

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

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

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

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

that is

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

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

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

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

Computational methods for certain structured problems

A particular case[2]:

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

with $A \in \mathbb{R}^{n\times n}$, **and the M_is having 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

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

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

Matrix-oriented Sherman-Morrison-Woodbury formula

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

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 $H = I + \mathcal{V}^T\mathcal{P} \in \mathbb{R}^{s^2 \times s^2}$
4. Solve $Hg = \mathcal{V}^T w$
5. Compute $x = w - \mathcal{P}g$.

Steps 1. and 2.:

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

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

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

Step 3.

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

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

Matrix-oriented Sherman-Morrison-Woodbury formula

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

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 $H = I + \mathcal{V}^T\mathcal{P} \in \mathbb{R}^{s^2 \times s^2}$
4. Solve $Hg = \mathcal{V}^T w$
5. Compute $x = w - \mathcal{P}g$.

Steps 1. and 2.:

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

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

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

Step 3.

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

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

Matrix-oriented Sherman-Morrison-Woodbury formula

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

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 $H = I + \mathcal{V}^T\mathcal{P} \in \mathbb{R}^{s^2 \times s^2}$
4. Solve $Hg = \mathcal{V}^T w$
5. Compute $x = w - \mathcal{P}g$.

Steps 1. and 2.:

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

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

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

Step 3.

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

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

Numerical examples. 1

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	3.817934e-11	0.195	2.292375e-10
	6/10	0.017	9.051326e-10	0.657	4.987161e-10
	12/20	0.035	5.259884e-09	2.333	1.357709e-08
80	3/5	0.022	2.152743e-10	5.283	1.228423e-09
	6/10	0.033	8.380606e-09	15.408	1.849484e-08
	12/20	0.074	2.502003e-08	56.347	3.467476e-08
160	3/5	0.043	1.291839e-09	129.957	6.891372e-09
	6/10	0.070	1.102578e-08	281.946	2.691323e-08
	12/20	0.220	2.907566e-07	1030.242	1.202511e-06

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

Numerical examples. 2

n	Matrix form CPU time	Matrix form Err	Vector Form CPU time	Vector Form Err
40	0.012	6.452267e-10	0.037	7.231068e-10
80	0.013	2.750854e-10	0.124	2.012480e-09
160	0.024	3.253562e-09	0.581	7.208432e-09
320	0.056	4.615180e-08	2.763	1.710614e-07

Table: Numerical results for symmetric and tridiagonal banded matrix A and U_1, U_2 random with $s_1 = 3$, $s_2 = 5$ columns, resp.

n	Matrix form CPU time	Matrix form Err	Vector Form CPU time	Vector Form Err
40	0.063	6.582486e-11	0.361	1.410898e-10
80	0.093	1.184547e-08	6.116	1.867875e-08
160	0.430	2.691697e-07	278.895	1.301593e-06

Table: Numerical results for nonsymmetric and full matrix A, with $s_1 = 3$, $s_2 = 5$ columns, resp.

Conclusions. 2

- First examples where structure can be exploited

(Not reported) This approach can be used for solving linear *tensor* equations

- Devise more general "direct" solvers, to be used 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 on regular domains* April 2020. HAL archive: hal-02902456
2. Yue Hao and V. S., *The Sherman-Morrison-Woodbury formula for generalized linear matrix equations and applications*, July 2020. HAL archive hal-02902455.
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), pp. 377-441.

Conclusions. 2

- First examples where structure can be exploited

(Not reported) This approach can be used for solving linear *tensor* equations

- Devise more general “direct” solvers, to be used 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 on regular domains* April 2020. HAL archive: hal-02902456
2. Yue Hao and V. S., *The Sherman-Morrison-Woodbury formula for generalized linear matrix equations and applications*, July 2020. HAL archive hal-02902455.
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), pp. 377-441.

Conclusions. 2

- First examples where structure can be exploited

(Not reported) This approach can be used for solving linear *tensor* equations

- Devise more general "direct" solvers, to be used 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 on regular domains* April 2020. HAL archive: hal-02902456
2. Yue Hao and V. S., *The Sherman-Morrison-Woodbury formula for generalized linear matrix equations and applications*, July 2020. HAL archive hal-02902455.
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), pp. 377-441.