Matrix oriented methods for dynamical systems

V. Simoncini

Dipartimento di Matematica

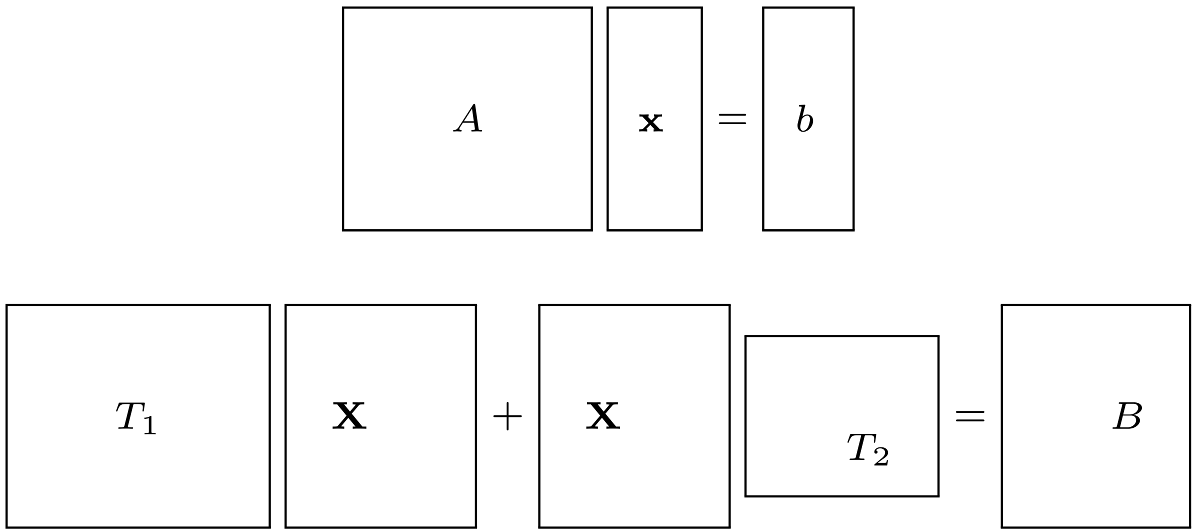
Alma Mater Studiorum - Università di Bologna

valeria.simoncini@unibo.it

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

Problem: solve the linear problem

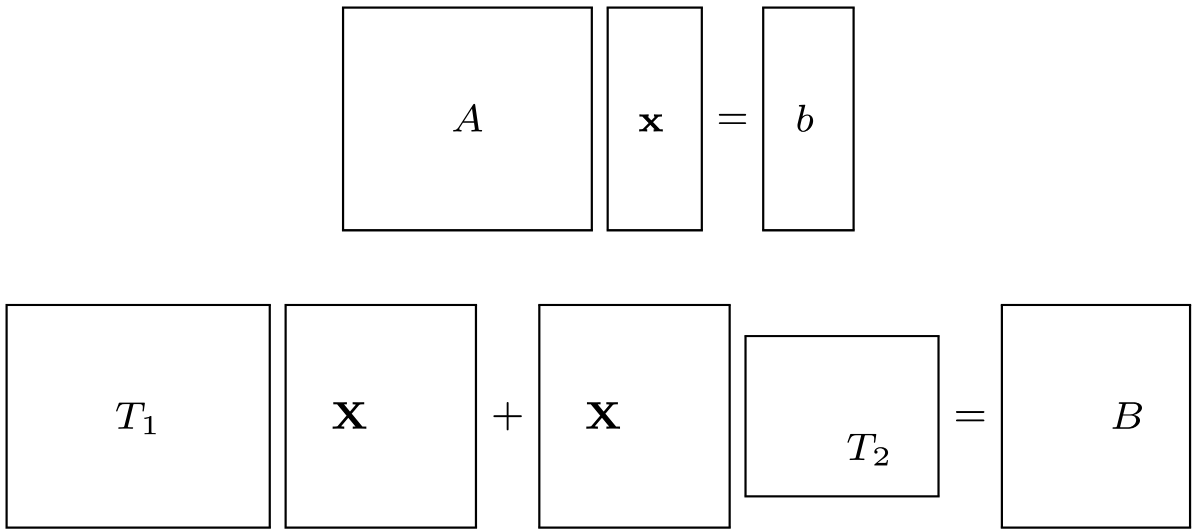
$$A\mathbf{x} = b \qquad \text{or} \qquad T_1\mathbf{X} + \mathbf{X}T_2 = B$$



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

Problem: solve the linear problem

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

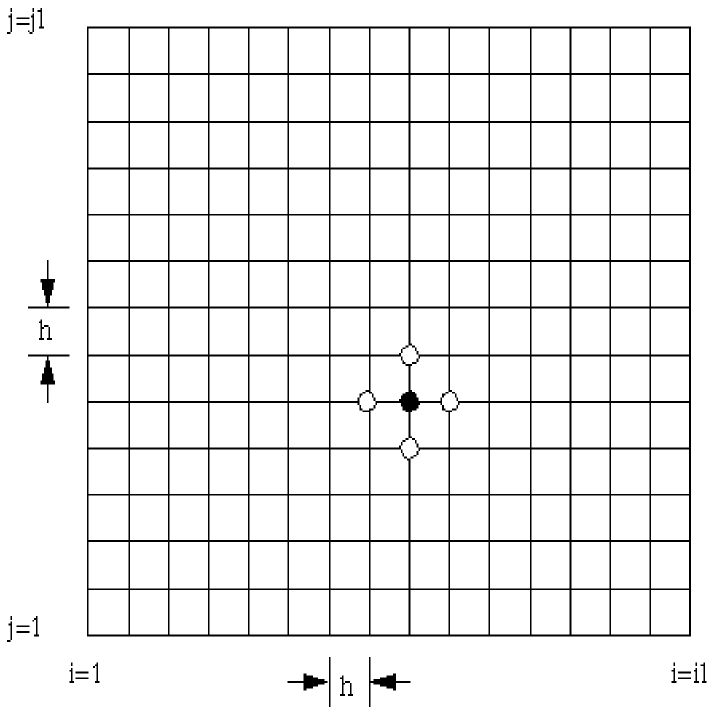


Remark: In discretizing PDEs with tensor bases, the two problems may be mathematically equivalent !

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

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

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

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

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: $\mathbf{U} \to \mathbf{u} = [\mathbf{U}_{11}, \mathbf{U}_{n,1}, \mathbf{U}_{1,2}, \ldots, \mathbf{U}_{n,2}, \ldots]^\top$

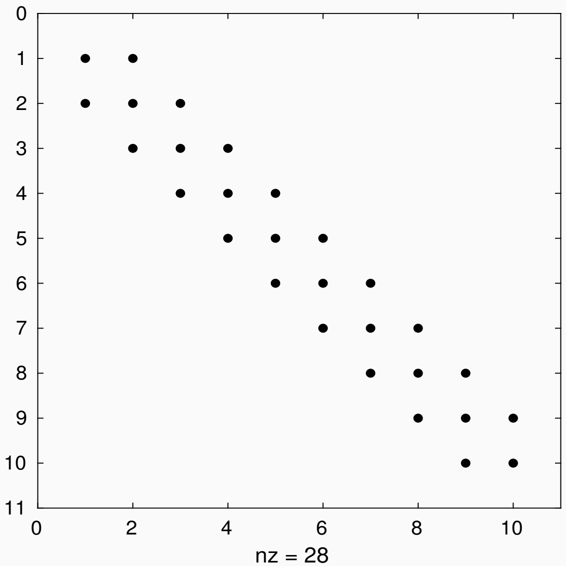
$$A\mathbf{u} = f \qquad A = I \otimes T_1 + T_1 \otimes I, \;\; f = \text{vec}(F),$$

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

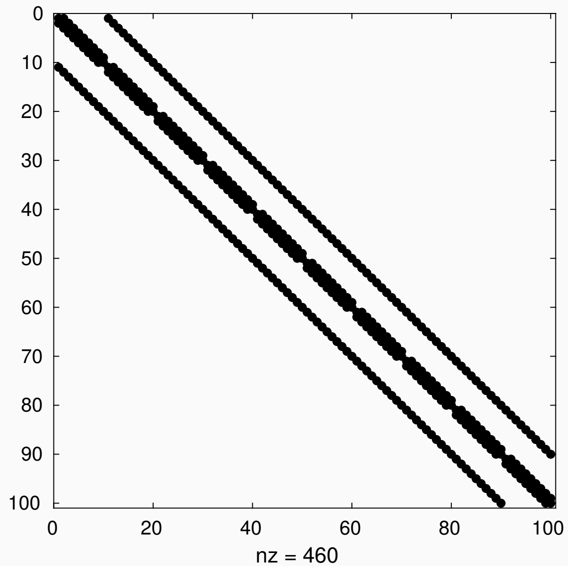
Numerical considerations

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

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



T_1



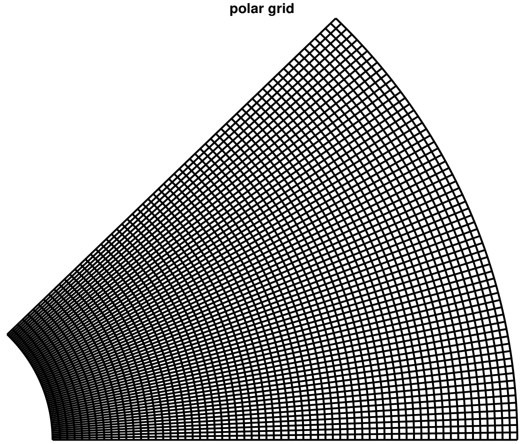
A

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:

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

♣ 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:

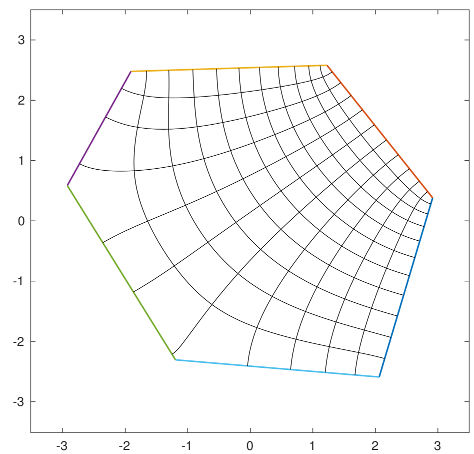
$$\boxed{T\widehat{\boldsymbol{U}} + +\widehat{\boldsymbol{U}}T = \widehat{F}}$$

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

♣ Schwarz-Christoffel conformal mappings between polygon and rectangle

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

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



With finite diff. discretization:

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

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

Poisson equation is the ideal setting for SC mappings!

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)

Numerical solution of the Sylvester equation

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

Various settings:

- <u>Small A and small B:</u> 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 $\boldsymbol{Y}$;
  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 $\boldsymbol{Y}$;
  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 $\boldsymbol{U} = \widehat{\boldsymbol{U}}W^{-1}$

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 $\boldsymbol{Y}$;
  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 $\boldsymbol{U} = \widehat{\boldsymbol{U}}W^{-1}$
- Large A and large B: Iterative solution (G low rank)

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:

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

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

Index k "related" to the approximation rank

Two applications

- Time stepping systems of *Reaction-diffusion PDEs:*

$$\begin{cases} u_t = \ell_1(u) + f_1(u, v), \\ v_t = \ell_2(v) + f_2(u, v), \quad \text{with} \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0, \boldsymbol{\tau}] \end{cases}$$

ℓ_i: diffusion operator linear in u $\qquad$ f_i: nonlinear reaction terms

Two applications

- Time stepping systems of *Reaction-diffusion PDEs:*

$$\begin{cases} u_t = \ell_1(u) + f_1(u,v), \\ v_t = \ell_2(v) + f_2(u,v), \quad \text{with} \quad (x,y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0, \boldsymbol{\tau}] \end{cases}$$

ℓ_i: diffusion operator linear in u $\qquad$ f_i: nonlinear reaction terms

- All-at-once *Heat equation:*

$$u_t + \Delta u = f, \qquad u = u(x,y,z,t) \in \Omega \times \mathcal{I},$$

with $\Omega \subset \mathbb{R}^3$, $\mathcal{I} = (0, \boldsymbol{\tau})$ and zero Dirichlet b.c.

Systems of Reaction-diffusion PDEs

$$\begin{cases} u_t = \ell_1(u) + f_1(u,v), \\ v_t = \ell_2(v) + f_2(u,v), \quad \text{with} \quad (x,y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0,T] \end{cases}$$

with $u(x,y,0) = u_0(x,y)$, $v(x,y,0) = v_0(x,y)$, and appropriate b.c. on Ω

ℓ_i: diffusion operator linear in u $\qquad$ f_i: nonlinear reaction terms

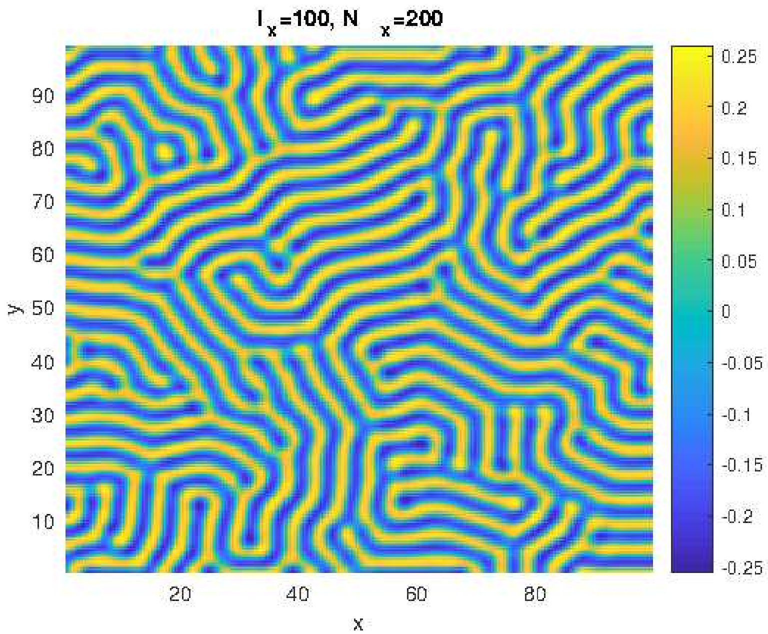
Applications:

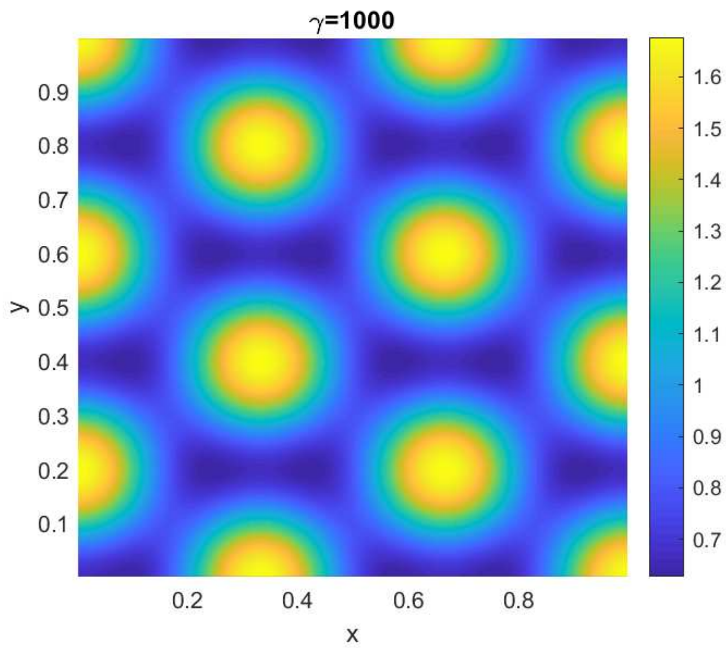
chemistry, biology, ecology, and more recently in metal growth by electrodeposition, tumor growth, biomedicine and cell motility

$\Rightarrow$ spatial patterns such as labyrinths, spots, stripes

Joint work with M.C. D'Autilia & I. Sgura, Università di Lecce

Long term spatial patterns





Labyrinths, spots, stripes, etc.

Numerical modelling issues

$$\begin{cases} u_t = \ell_1(u) + f_1(u,v), \\ v_t = \ell_2(v) + f_2(u,v), \quad \text{with} \quad (x,y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0,T] \end{cases}$$

- Problem is stiff
  - Use appropriate time discretizations
  - Time stepping constraints
- Pattern visible only after long time period (transient unstable phase)
- Pattern visible only if domain is well represented

Space discretization of the reaction-diffusion PDE

ℓ_i: elliptic operator $\Rightarrow \ell_i(u) \approx A_i \mathbf{u}$, so that

$$\begin{cases} \dot{\mathbf{u}} = A_1\mathbf{u} + f_1(\mathbf{u},\mathbf{v}), & \mathbf{u}(0) = \mathbf{u}_0, \\ \dot{\mathbf{v}} = A_2\mathbf{v} + f_2(\mathbf{u},\mathbf{v}), & \mathbf{v}(0) = \mathbf{v}_0 \end{cases}$$

Key fact: Ω simple domain, e.g., $\Omega = [0,\ell_x] \times [0,\ell_y]$. Therefore

$$A_i = I_y \otimes T_{1i} + T_{2i}^\top \otimes I_x \in \mathbb{R}^{N_x N_y \times N_x N_y}, \; i = 1,2$$

$\Rightarrow A\mathbf{u} = \mathrm{vec}(T_1 U + U T_2)$

Matrix-oriented formulation of reaction-diffusion PDEs

$$\begin{cases} \dot{U} = T_{11}U + UT_{12} + F_1(U,V), & U(0) = U_0, \\ \dot{V} = T_{21}V + VT_{22} + F_2(U,V), & V(0) = V_0 \end{cases}$$

$F_i(U,V)$ nonlinear vector function $f(\mathbf{u},\mathbf{v})$ evaluated componentwise

$\mathrm{vec}(U_0) = \mathbf{u}_0$, $\mathrm{vec}(V_0) = \mathbf{v}_0$, initial conditions

Remark: Computational strategies for time stepping can exploit this setting

---

For simplicity of exposition, we consider $\dot{\mathbf{u}} = A\mathbf{u} + f(\mathbf{u})$, that is

$$\dot{U} = T_1U + UT_2 + F(U), \quad (x,y) \in \Omega, \ t \in]0,T]$$

Time stepping Matrix-oriented methods

IMEX methods

1. *First order Euler:* $\mathbf{u}_{n+1} - \mathbf{u}_n = h_t(A\mathbf{u}_{n+1} + f(\mathbf{u}_n))$ so that

$$(I - h_t A)\mathbf{u}_{n+1} = \mathbf{u}_n + h_t f(\mathbf{u}_n), \quad n = 0, \ldots, N_t - 1$$

Matrix-oriented form: $U_{n+1} - U_n = h_t(T_1 U_{n+1} + U_{n+1} T_2) + h_t F(U_n)$,

so that

$$(I - h_t T_1)\mathbf{U}_{n+1} + \mathbf{U}_{n+1}(-h_t T_2) = U_n + h_t F(U_n), \quad n = 0, \ldots, N_t - 1.$$

Time stepping Matrix-oriented methods

IMEX methods

1. *First order Euler:* $\mathbf{u}_{n+1} - \mathbf{u}_n = h_t(A\mathbf{u}_{n+1} + f(\mathbf{u}_n))$ so that

$$(I - h_t A)\mathbf{u}_{n+1} = \mathbf{u}_n + h_t f(\mathbf{u}_n), \quad n = 0, \ldots, N_t - 1$$

Matrix-oriented form: $U_{n+1} - U_n = h_t(T_1 U_{n+1} + U_{n+1} T_2) + h_t F(U_n)$,

so that

$$(I - h_t T_1)\mathbf{U}_{n+1} + \mathbf{U}_{n+1}(-h_t T_2) = U_n + h_t F(U_n), \quad n = 0, \ldots, N_t - 1.$$

2. *Second order SBDF*, known as IMEX 2-SBDF method

$$3\mathbf{u}_{n+2} - 4\mathbf{u}_{n+1} + \mathbf{u}_n = 2h_t A\mathbf{u}_{n+2} + 2h_t(2f(\mathbf{u}_{n+1}) - f(\mathbf{u}_n)), \quad n = 0, 1, \ldots, N_t$$

Matrix-oriented form: for $n = 0, \ldots, N_t - 2$,

$$(3I - 2h_t T_1)\,\mathbf{U}_{n+2} + \mathbf{U}_{n+2}\,(-2h_t T_2) = 4U_{n+1} - U_n + 2h_t(2F(U_{n+1}) - F(U_n))$$

Time stepping Matrix-oriented methods

Exponential integrator

Exponential first order Euler method:

$$\boxed{\mathbf{u}_{n+1} = e^{h_t A}\mathbf{u}_n + h_t \varphi_1(h_t A) f(\mathbf{u}_n)}$$

$e^{h_t A}$: matrix exponential, $\varphi_1(z) = (e^z - 1)/z$ first "phi" function

That is,

$$\mathbf{u}_{n+1} = e^{h_t A}\mathbf{u}_n + h_t \mathbf{v}_n, \quad \text{where } A\mathbf{v}_n = e^{h_t A} f(\mathbf{u}_n) - f(\mathbf{u}_n) \qquad n = 0, \ldots, N_t - 1. \tag{1}$$

Time stepping Matrix-oriented methods

Exponential integrator

Exponential first order Euler method:

$$\boxed{\mathbf{u}_{n+1} = e^{h_t A}\mathbf{u}_n + h_t\varphi_1(h_t A) f(\mathbf{u}_n)}$$

$e^{h_t A}$: matrix exponential, $\varphi_1(z) = (e^z - 1)/z$ first "phi" function

That is,

$$\mathbf{u}_{n+1} = e^{h_t A}\mathbf{u}_n + h_t\mathbf{v}_n, \quad \text{where } A\mathbf{v}_n = e^{h_t A} f(\mathbf{u}_n) - f(\mathbf{u}_n) \qquad n = 0, \ldots, N_t - 1.$$

Matrix-oriented form: since $e^{h_t A}\mathbf{u} = \left(e^{h_t T_2^T} \otimes e^{h_t T_1}\right)\mathbf{u} = \mathrm{vec}(e^{h_t T_1} U e^{h_t T_2})$

1. Compute $E_1 = e^{h_t T_1}$, $E_2 = e^{h_t T_2^T}$
2. For each n

$$\begin{aligned} \text{Solve} \qquad & T_1\mathbf{V}_n + \mathbf{V}_n T_2 = E_1 F(U_n) E_2^T - F(U_n) \\ \text{Compute} \qquad & U_{n+1} = E_1 U_n E_2^T + h_t V_n \end{aligned} \tag{2}$$

Time stepping Matrix-oriented methods

Computational issues:

- Dimensions of T_1, T_2 very modest
- T_1, T_2 quasi-symmetric (non-symmetry due to b.c.)
- T_1, T_2 do not depend on time step

♣ Matrix-oriented form all in spectral space (after eigenvector transformation)

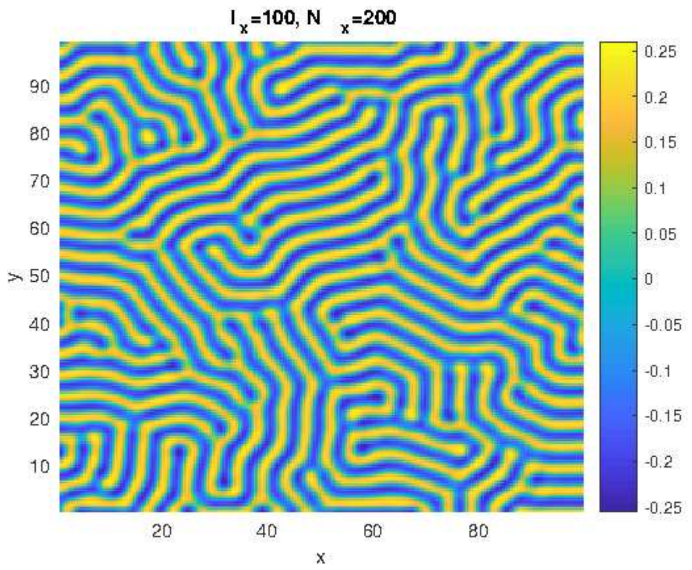
A numerical example of system of RD-PDEs

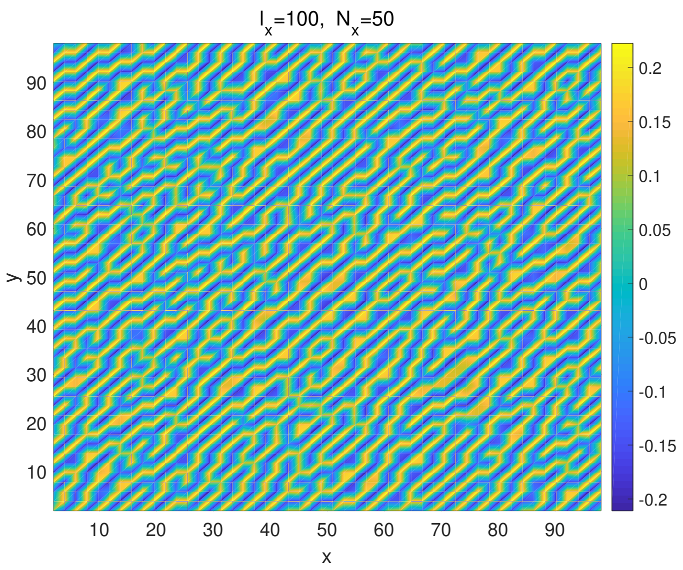
Model describing an electrodeposition process for metal growth

$$f_1(u,v) = \rho\left(\alpha_1(1-v)u - \alpha_2\, u^3 - \beta(v-\alpha)\right)$$

$$f_2(u,v) = \rho\,(\gamma_1(1+k_2u)(1-v)[1-\gamma(1-v)] - \delta_1 v(1+k_3u)(1+\gamma v)))$$

Turing pattern





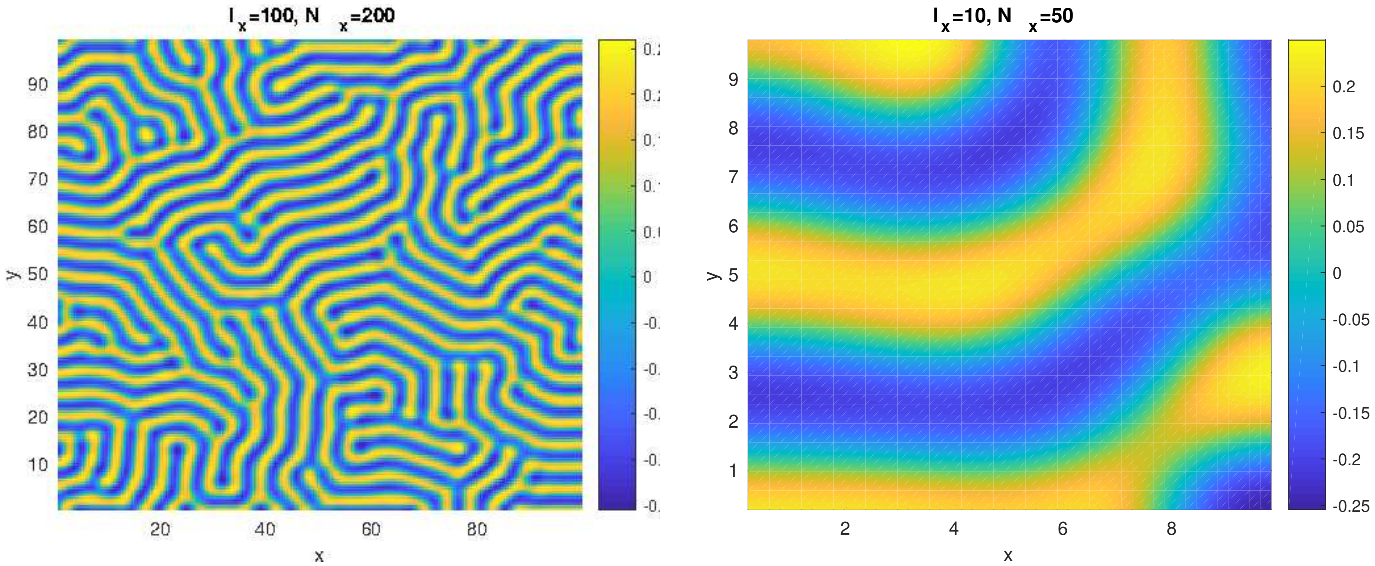
A numerical example of system of RD-PDEs

Model describing an electrodeposition process for metal growth

$$f_1(u,v) = \rho\left(\alpha_1(1-v)u - \alpha_2\, u^3 - \beta(v-\alpha)\right)$$

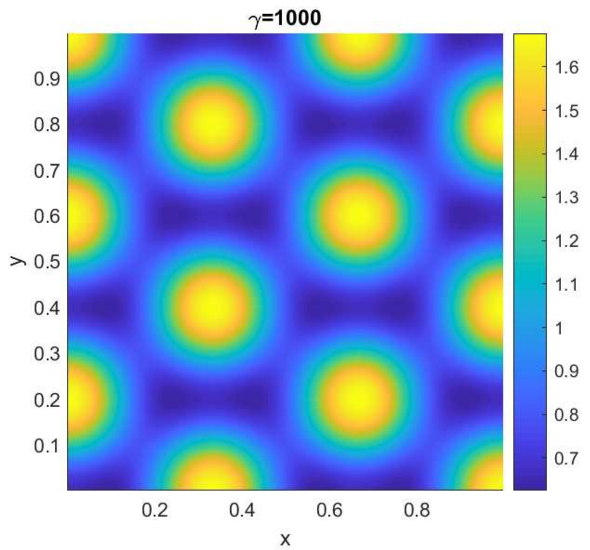
$$f_2(u,v) = \rho\left(\gamma_1(1+k_2 u)(1-v)[1-\gamma(1-v)] - \delta_1 v(1+k_3 u)(1+\gamma v)\right))$$

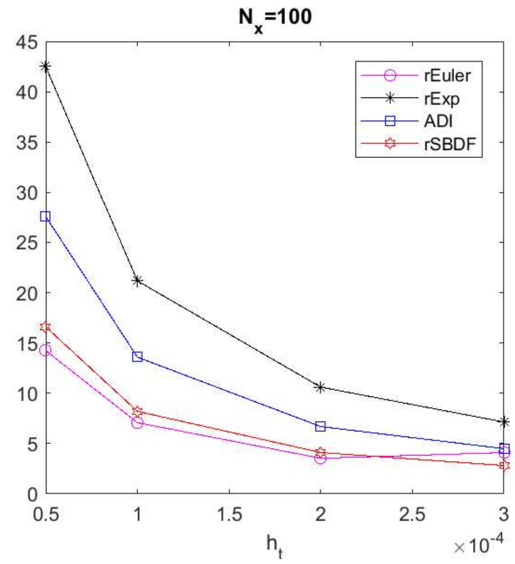
Turing pattern

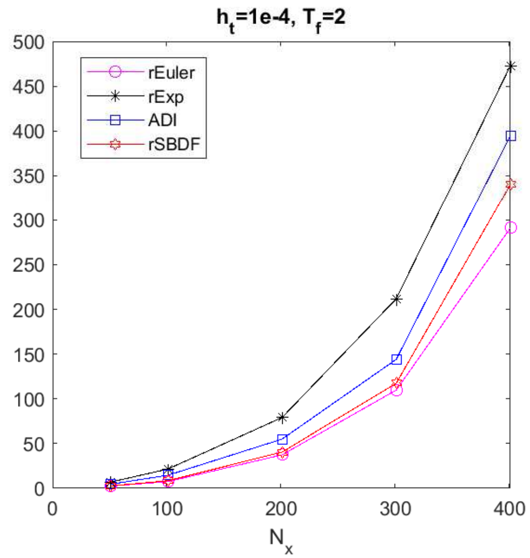


Schnackenberg model

$$f_1(u,v) = \gamma(a - u + u^2 v), \qquad f_2(u,v) = \gamma(b - u^2 v)$$







Left plot: Turing pattern solution for $\gamma = 1000$ ($N_x = 400$)

Center plot: CPU times (sec), $N_x = 100$ variation of h_t

Right plot: CPU times (sec), $h_t = 10^{-4}$, increasing values of $N_x = 50, 100, 200, 300, 400$

All-at-once heat equation

$$u_t + \ell(u) = f \qquad u(0) = 0 \quad \text{(for convenience)}$$

Variational formulation

$$\text{find } u \in U : \quad b(u,v) = \langle f, v\rangle \quad \text{for all } v \in V$$

where

$U := H^1_{(0)}(\mathcal{I}; X') \cap L_2(\mathcal{I}, X),\ X := H^1_0(\Omega),\ V := L_2(\mathcal{I}; X)$

$b(u,v) := \int_0^{\boldsymbol{\tau}} \int_\Omega u_t(t,x)\, v(t,x)\, dx\, dt + \int_0^{\boldsymbol{\tau}} a(u(t), v(t))\, dt$

$\langle f, v\rangle := \int_0^{\boldsymbol{\tau}} \int_\Omega f(t,x)\, v(t,x)\, dx\, dt.$

♣ It can be shown that this formulation is well-posed

♣ Variational approach in space+time allows for adaptivity and order reduction on both types of variables

Joint work with J. Henning, D. Palitta and K. Urban

All-at-once heat equation. Discretized problem

Choose finite-dimensional trial and test spaces, $U_\delta \subset U$, $V_\delta \subset V$.

Then the Petrov-Galerkin method reads

$$\text{find } u_\delta \in U_\delta : \quad b(u_\delta, v_\delta) = \langle f, v_\delta \rangle \quad \text{for all } v_\delta \in V_\delta$$

with $U_\delta := S_{\Delta t} \otimes X_h$, $V_\delta = Q_{\Delta t} \otimes X_h$ where

$S_{\Delta t}$: piecewise linear FE on $\mathcal{I}$

$Q_{\Delta t}$: piecewise constant FE on $\mathcal{I}$

X_h : any conformal space, e.g., piecewise linear FE

♣ Well-posedness (discrete inf-sup cond) depends on the choice of U_δ, V_δ

All-at-once heat equation. Discretized problem

Choose finite-dimensional trial and test spaces, $U_\delta \subset U$, $V_\delta \subset V$.

Then the Petrov-Galerkin method reads

$$\text{find } u_\delta \in U_\delta : \quad b(u_\delta, v_\delta) = \langle f, v_\delta \rangle \quad \text{for all } v_\delta \in V_\delta$$

with $U_\delta := S_{\Delta t} \otimes X_h$, $V_\delta = Q_{\Delta t} \otimes X_h$ where

$S_{\Delta t}$: piecewise linear FE on $\mathcal{I}$

$Q_{\Delta t}$: piecewise constant FE on $\mathcal{I}$

X_h : any conformal space, e.g., piecewise linear FE

♣ Well-posedness (discrete inf-sup cond) depends on the choice of U_δ, V_δ

Remark: This discretization coincides with Crank–Nicolson scheme if trapezoidal approximation of the rhs temporal integration is used

The final linear system

$$B_\delta^\top u_\delta = f_\delta$$

where

$$[B_\delta]_{(k,i),(\ell,j)} = (\dot\sigma^k, \tau^\ell)_{L_2(\mathcal{I})}\,(\phi_i, \phi_j)_{L_2(\Omega)} + (\sigma^k, \tau^\ell)_{L_2(\mathcal{I})}\, a(\phi_i, \phi_j),$$

$$[f_\delta]_{(\ell,j)} = (f, \tau^\ell \otimes \phi_j)_{L_2(\mathcal{I};H)}$$

that is, $B_\delta = D_{\Delta t} \otimes M_h + C_{\Delta t} \otimes K_h$

Remark: We approximate f_δ to achieve full tensor-product structure

The final linear system

$$B_\delta^\top u_\delta = f_\delta$$

where

$$[B_\delta]_{(k,i),(\ell,j)} = (\dot{\sigma}^k, \tau^\ell)_{L_2(\mathcal{I})}\,(\phi_i, \phi_j)_{L_2(\Omega)} + (\sigma^k, \tau^\ell)_{L_2(\mathcal{I})}\,a(\phi_i, \phi_j),$$

$$[f_\delta]_{(\ell,j)} = (f, \tau^\ell \otimes \phi_j)_{L_2(\mathcal{I};H)}$$

that is, $B_\delta = D_{\Delta t} \otimes M_h + C_{\Delta t} \otimes K_h$

Remark: We approximate f_δ to achieve full tensor-product structure

This yields the generalized Sylvester equation:

$$M_h \mathbf{U}_\delta D_{\Delta t} + K_h \mathbf{U}_\delta C_{\Delta t} = F_\delta, \qquad \text{with} \quad F_\delta = [g_1, \ldots, g_P][h_1, \ldots, h_P]^\top$$

$\boxed{F_\delta \text{ matrix of low rank} \quad \Rightarrow \quad \mathbf{U}_\delta \text{ approx by low rank matrix } \widetilde{\mathbf{U}}_\delta}$

A simple example

$\Omega = (-1,1)^3$, with homogeneous Dirichlet boundary conditions

$\mathcal{I} = (0,10)$ and initial conditions $u(0,x,y,z) \equiv 0$

$f(t,x,y,z) := 10\sin(t)t\cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)\cos(\frac{\pi}{2}z)$ (F_δ is thus low rank)

		RKSM				CN Time(s)	
N_h	N_t	Its	μ_{mem}	rank($\widetilde{U}_\delta$)	Time(s)	Direct	Iterative
41 300	300	13	14	9	25.96	123.43	59.10
	500	13	14	9	30.46	143.71	78.01
	700	13	14	9	28.17	153.38	93.03
347 361	300	14	15	9	820.17	14705.10	792.42
	500	14	15	9	828.34	15215.47	1041.47
	700	14	15	7	826.93	15917.52	1212.57

♣ Memory allocations in CN are for full problem

♣ Sylvester-oriented method: overall Space and Time independence

Multiterm linear matrix equation

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

Applications:

- Control
- (Stochastic) PDEs
- Matrix least squares
- ...

Multiterm linear matrix equation

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

Applications:

- Control
- (Stochastic) PDEs
- Matrix least squares
- ...

Main device: Kronecker formulation

$$\left(B_1^\top \otimes A_1 + \ldots + B_\ell^\top \otimes A_\ell\right) x = c$$

Iterative methods: matrix-matrix multiplications and rank truncation

(Benner, Breiten, Bouhamidi, Chehab, Damm, Grasedyck, Jbilou, Kressner, Matthies, Onwunta, Raydan, Stoll, Tobler, Zander, and **many** others...)

Multiterm linear matrix equation

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

Applications:

- Control
- (Stochastic) PDEs
- Matrix least squares
- ...

Alternative approaches:

- Projection onto rich approximation space
- Compression to two-term matrix equation
- Splitting strategy towards two-term matrix equation
- ...

PDEs on polygonal grids and separable coeffs

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

ϕ_i, ψ_i, γ_i, $i=1,2$ sufficiently regular functions + b.c.

Problem discretization by means of a tensor basis

Multiterm linear 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

PDEs with random inputs

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

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

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,

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

$\mu(\mathbf{x})$: expected value of diffusion coef. σ: std dev.

$(\lambda_r, \phi_r(\mathbf{x}))$ eigs of the integral operator $\mathcal{V}$ wrto $V(\mathbf{x},\mathbf{x}') = \frac{1}{\sigma^2}C(\mathbf{x},\mathbf{x}')$

($\lambda_r \searrow$ $\quad C : D \times D \to \mathbb{R}$ covariance fun.)

Discretization by stochastic Galerkin

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

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

$\mathbf{x}$: 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.)

$\mathbf{g}_0$: first column of G_0

$\mathbf{f}_0$: FE rhs of deterministic PDE

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

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

The matrix equation formulation

$$(G_0 \otimes K_0 + G_1 \otimes K_1 + \ldots + G_m \otimes K_m)\,\mathbf{x} = \mathbf{g}_0 \otimes \mathbf{f}_0$$

transforms into

$$K_0\mathbf{X}G_0 + K_1\mathbf{X}G_1 + \ldots + K_m\mathbf{X}G_m = F, \qquad F = \mathbf{f}_0\mathbf{g}_0^\top$$

$(G_0 = I)$

Solution strategy. Conjecture:

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

$$\Downarrow$$

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

(Possibly extending results of Grasedyck, 2004)

Matrix Galerkin approximation of the deterministic part. 1

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

$$V_k^\top R_k = 0, \qquad R_k := K_0 X_k + K_1 X_k G_1 + \ldots + K_m X_k G_m - \mathbf{f}_0 \mathbf{g}_0^\top$$

Computational challenges:

- Generation of $\mathcal{K}_k$ involved $m+1$ 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, SISC 2017)

PDE-Constrained optimization problems

Functional to be minimized:

$$J(y,u) = \frac{1}{2}\int_0^T \int_{\Omega_1} (y-\hat{y})^2 \,\mathrm{d}x\,\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_{\Omega_u} u^2 \,\mathrm{d}x\,\mathrm{d}t. \tag{3}$$

$\star$ y: is the state, $\hat{y}$ is the desired state given on a subset Ω_1 of Ω,

$\star$ u is the control on a subset Ω_u of Ω,
(regularized by the control cost parameter β)

PDE constraining the functional $J(y,u)$ (Dirichlet b.c.):

$$\dot{y} - \Delta y = u \quad \text{in} \quad \Omega_u, \tag{4}$$

$$\dot{y} - \Delta y = 0 \quad \text{in} \quad \Omega/\Omega_u, \tag{5}$$

$$y = 0 \quad \text{on} \quad \partial\Omega. \tag{6}$$

♣ All-at-once strategy (space and time)

(Alexandra Bünger, V.S., and Martin Stoll, tr. 2020)

Conclusions and Outlook

Large-scale linear matrix equations are a new computational tool

General Considerations:

- Matrix-oriented versions lead to computational and numerical advantages
- Matrix equation challenges rely on strength of linear system solvers

Current activities:

- Large Nonlinear time-dependent problems with POD-DEIM (w/ G. Kirsten)
- Matrix-oriented 3D time-dependent problems require tensors

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

References

[0] V.S., *Computational methods for linear matrix equations* SIAM Rev., 58(2016)

[1] Maria Chiara D'Autilia, Ivonne Sgura and V. S. *Matrix-oriented discretization methods for reaction-diffusion PDEs: comparisons and applications.* Computers and Mathematics with Applications, 2020.

[2] Julian Henning, Davide Palitta, V. S. and Karsten Urban. *Matrix oriented reduction of space-time Petrov-Galerkin variational problems.* Proceedings ENUMATH 2019.

[3] Alexandra Bünger, V.S., and Martin Stoll. *A low-rank matrix equation method for solving PDE-constrained optimization problems* Arxiv preprint n. 2005.14499, 2020.

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

[5] Yue Hao and V. S. *Matrix equation solving of PDEs in polygonal domains using conformal mappings*, To appear in J. Numerical Mathematics, 2021.

[6] Catherine E. Powell, David Silvester, V.S. *An efficient reduced basis solver for stochastic Galerkin matrix equations*, SIAM J. Scient. Comput., 39, (2017)