Matrix equations. Application to PDEs

V. Simoncini

Dipartimento di Matematica

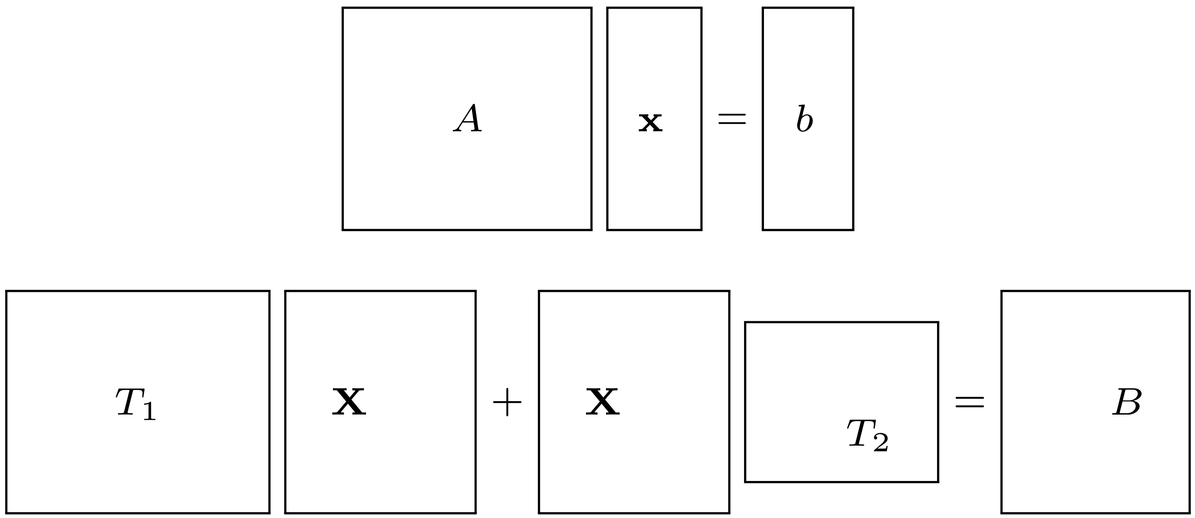
Alma Mater Studiorum - Università di Bologna

valeria.simoncini@unibo.it

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

Problem: solve the linear problem

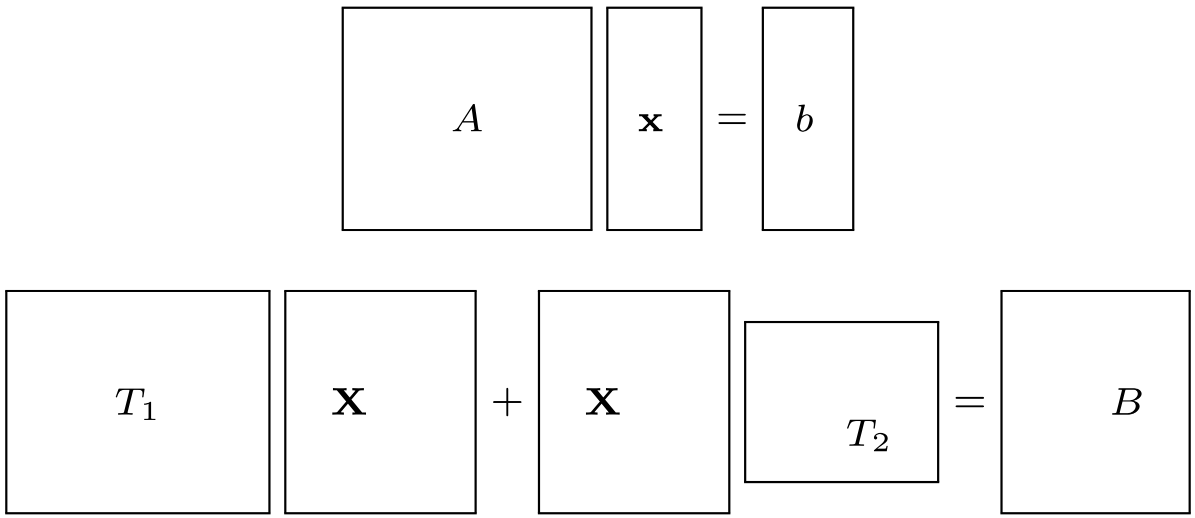
$$A\mathbf{x} = b \qquad \text{or} \qquad T_1\mathbf{X} + \mathbf{X}T_2 = B$$



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

Problem: solve the linear problem

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

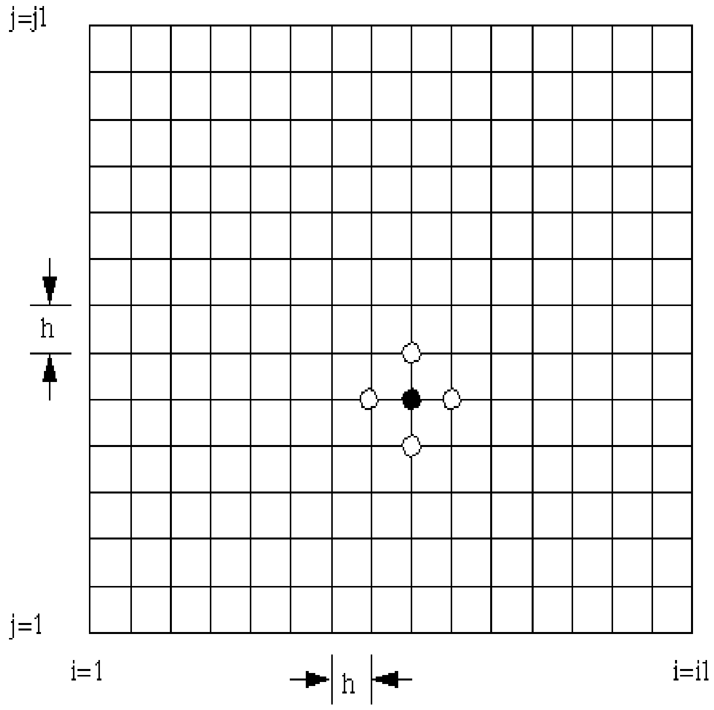


Remark: In discretizing PDEs with tensor bases, the two problems may be mathematically equivalent !

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

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

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

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

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

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

$$T_1 \mathbf{U} + \mathbf{U} T_1^\top = F, \qquad F_{ij} = f(x_i, y_j)$$

,

The Poisson equation

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

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

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

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

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

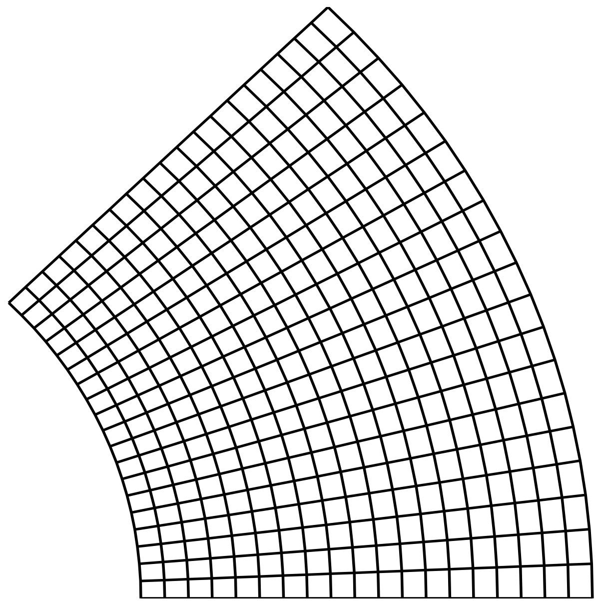
$$T_1 \mathbf{U} + \mathbf{U} T_1^\top = F, \quad F_{ij} = f(x_i, y_j)$$

Lexicographic ordering: $(M \otimes N) = (M_{i,j} N)_{k,\ell}$

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

A more general domain, with an explicit mapping

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



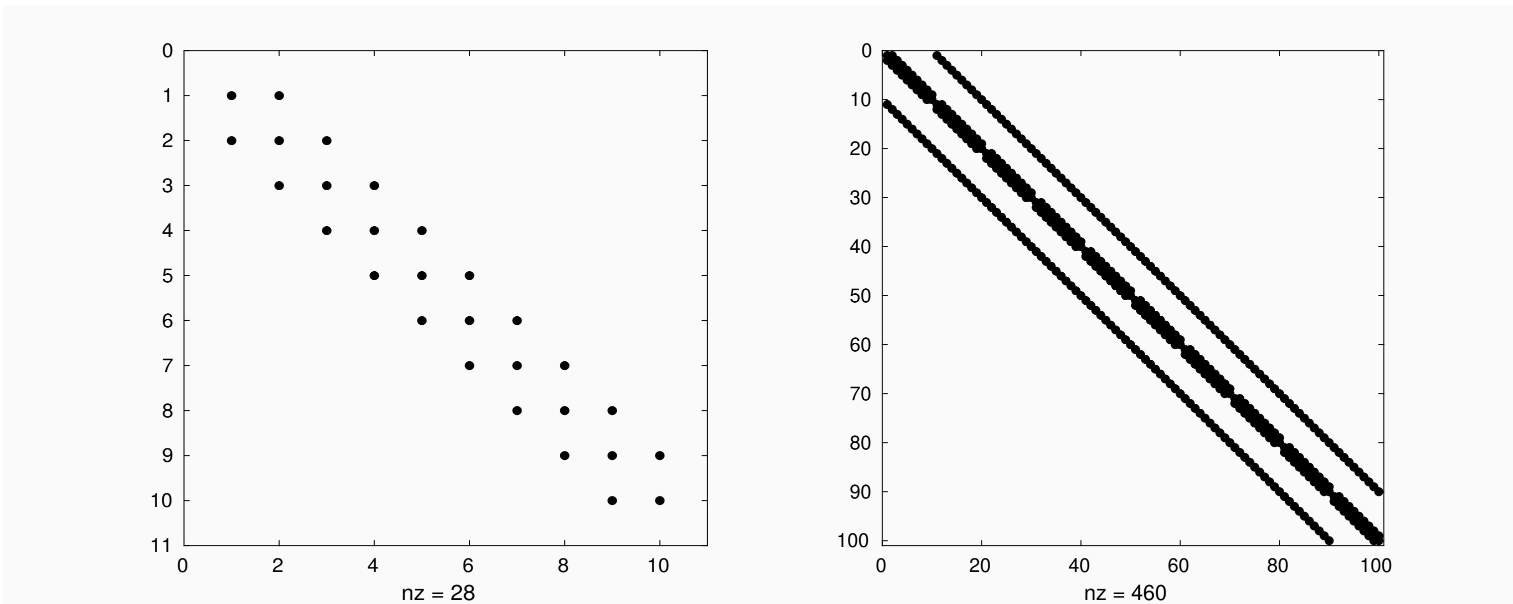
In polar coordinates (r, θ): $\qquad -u_{rr} - \frac{1}{r}u_r - u_{\theta\theta} = \widetilde{f}$

$\Rightarrow \qquad A_1\mathbf{X} + \mathbf{X}A_2 = F$

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 $\qquad$ A

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 f_i: nonlinear reaction terms

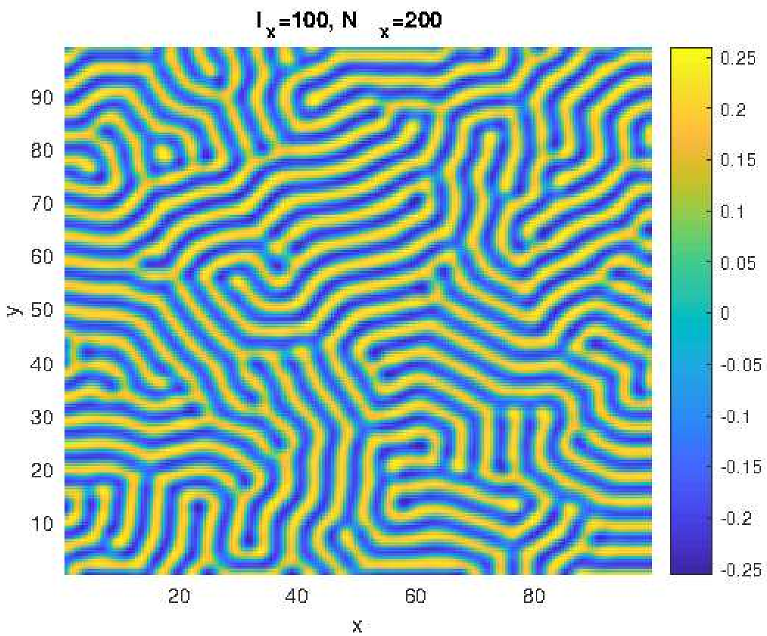
Applications:

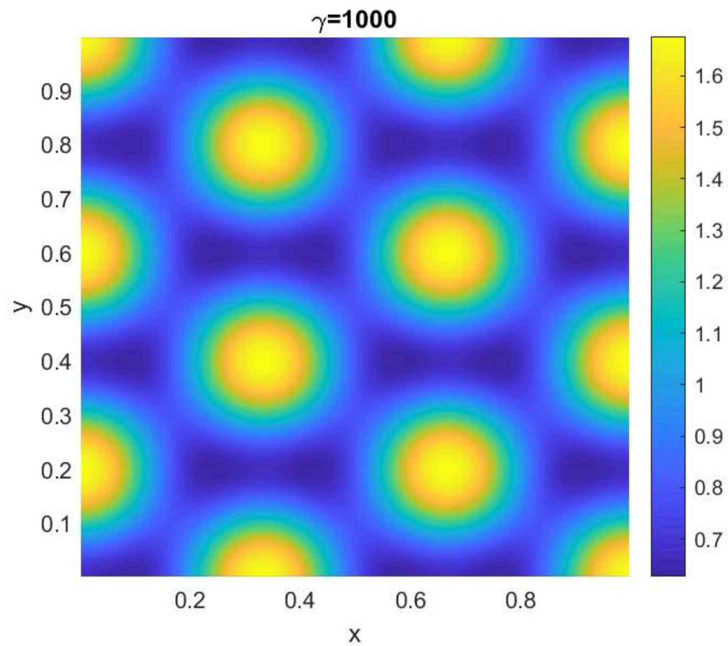
chemistry, biology, ecology, and more recently in metal growth by electrodeposition, tumor growth, biomedicine and cell motility

$\Rightarrow$ 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

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

$F_i(U,V)$ 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

$$\dot{U} = T_1 U + UT_2 + F(U), \quad (x,y) \in \Omega, \ t \in]0,T]$$

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:

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

$e^{h_t A}$: 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

$$\text{Solve} \qquad T_1 \mathbf{V}_n + \mathbf{V}_n T_2 = E_1 F(U_n) E_2^T - F(U_n) \qquad (2)$$

$$\text{Compute} \qquad U_{n+1} = E_1 U_n E_2^T + h_t V_n$$

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)

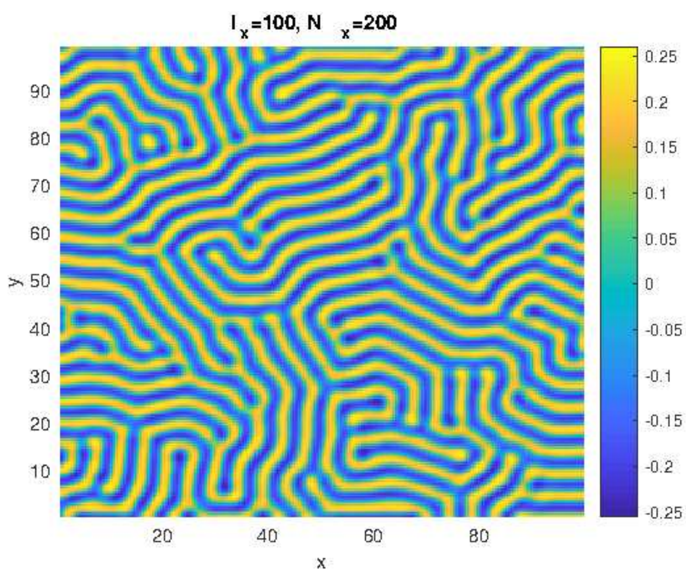
A numerical example of system of RD-PDEs

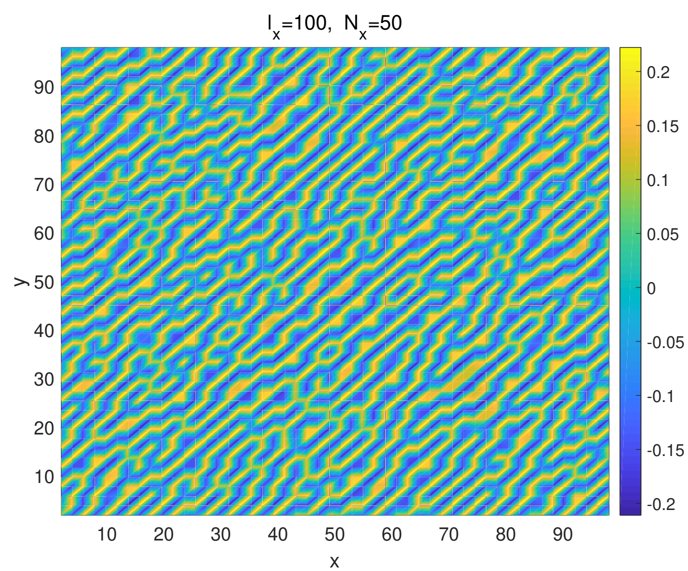
Model describing an electrodeposition process for metal growth

$$f_1(u,v) = \rho\left(A_1(1-v)u - A_2\,u^3 - B(v-\alpha)\right)$$

$$f_2(u,v) = \rho\left(C(1+k_2u)(1-v)[1-\gamma(1-v)] - Dv(1+k_3u)(1+\gamma v)\right)$$

Turing pattern





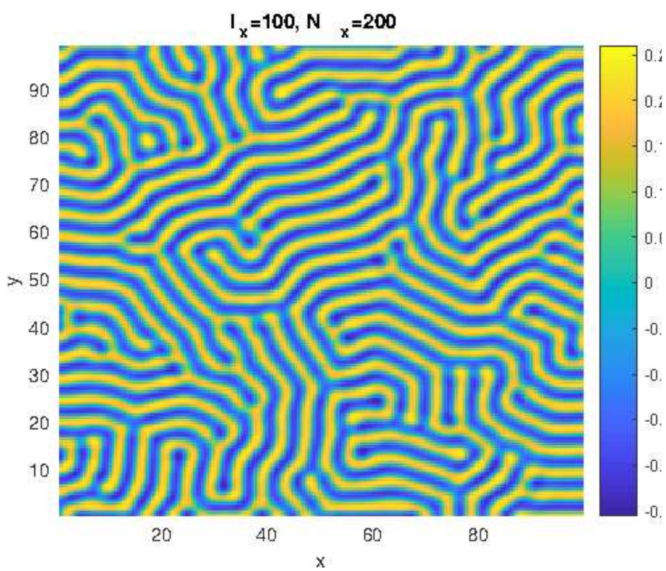
A numerical example of system of RD-PDEs

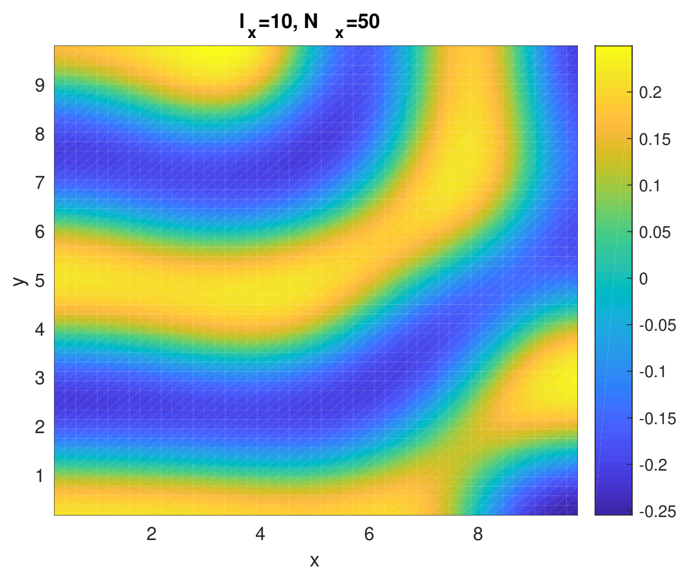
Model describing an electrodeposition process for metal growth

$$f_1(u,v) = \rho\left(A_1(1-v)u - A_2\,u^3 - B(v-\alpha)\right)$$

$$f_2(u,v) = \rho\left(C(1+k_2u)(1-v)[1-\gamma(1-v)] - Dv(1+k_3u)(1+\gamma v)\right))$$

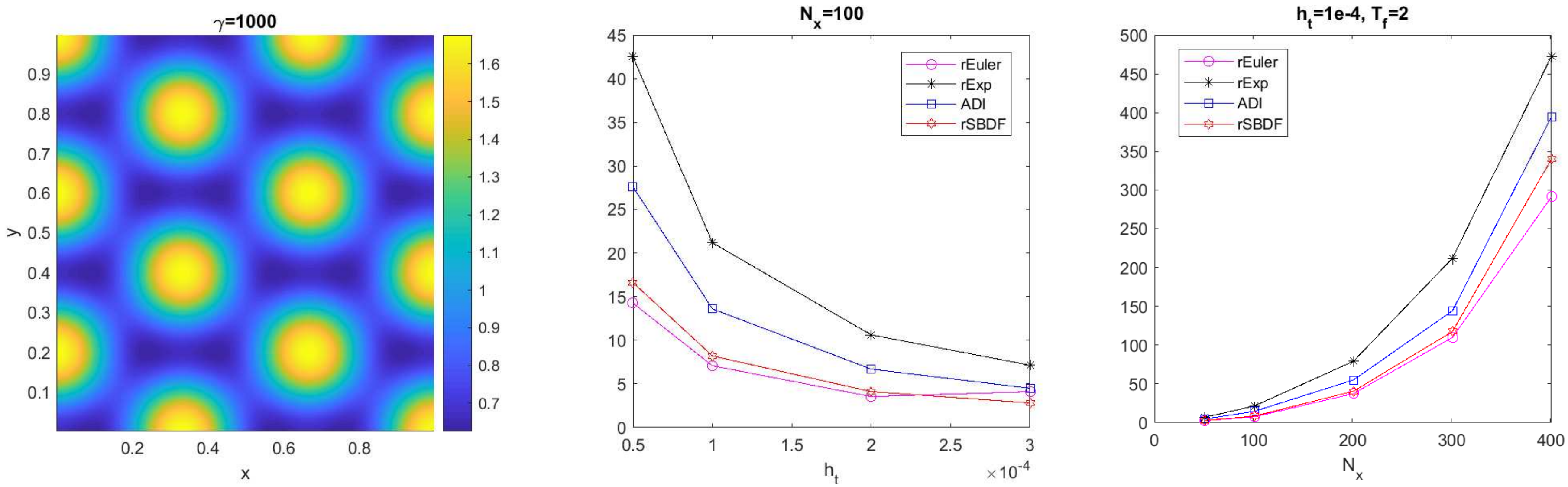
Turing pattern





Schnackenberg model

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



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

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

Variational formulation

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

where

$V := L_2(\mathcal{I}; H^1_0(\Omega))$

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

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

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

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

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

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:

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

F_δ matrix of low rank $\quad\Rightarrow\quad$ $\mathbf{U}_\delta$ 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
N_h	N_t	Its	μ_{mem}	rank($\widetilde{U}_\delta$)	Time (s)	Time (s)
41 300	100	18	19	11	101.32	296.16
	300	18	19	10	100.19	871.38
	500	18	19	10	101.92	1 468.40
347 361	100	20	21	9	4279.83	13 805.09
	300	20	21	9	4289.21	41 701.10
	500	20	21	8	4305.18	70 044.52

Conclusions and Outlook

Large-scale linear matrix equations are a new computational tool

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

Outlook:

- Large scale Nonlinear time-dependent problems with DEIM
- Matrix-oriented 3D time-dependent problems require tensors
- Low-rank tensor equations require new thinking

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

REFERENCES:

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

[2] Julian Henning, Davide Palitta, V. S. and Karsten Urban. *Matrix oriented reduction of space-time Petrov-Galerkin variational problems.* Dec 2019. arXiv:1912.10082