Università di Bologna

# Computational methods for linear matrix equations

V. Simoncini

Dipartimento di Matematica, Università di Bologna

valeria.simoncini@unibo.it

# Abstract

The numerical solution of possibly large dimensional algebraic linear systems permeates scientific modelling. Often systems with multiple right-hand sides arise, whose efficient numerical solution usually requires ad-hoc procedures. In the past decades a new class of linear equations has shown to be the natural algebraic framework in the discretization of mathematical models in a variety of scientific applications. These problems are given by multiterm linear matrix equations of the form $A_1XB_1 + A_2XB_2 + \ldots + A_kXB_k = C$, where all appearing terms are matrices of conforming dimensions, and $X$ is the (unknown) matrix solution. The case $k = 2$ is called the Sylvester equation, and computational methods for its solution are well established for small dimensions. Efficient methods for large scale Sylvester equations have recently been developed, under certain hypotheses on the data.
The general multiterm case is the current challenge, and it turns out to be a key ingredient in problems such as time-space, stochastic and parametric partial differential equations. We survey various methodologies for addressing the efficient and reliable solution of linear matrix equations. We focus on the algorithmic aspects as well as on the mathematical properties underlying the developed approaches. Typical application problems will be pinpointed.

# Outline

- Linear systems with multiple right-hand sides
- Shifted linear systems
- Two-term linear matrix equations (Sylvester, Lyapunov, special cases)
- General multi-term linear matrix equations: applications and algorithms
- Systems of linear matrix equations
- If time allows: Algebraic Riccati equation

## Multiple right-hand side algebraic linear systems

$$AX = C, \qquad C = [c_1, \ldots, c_s] \in \mathbb{R}^{n\times s}$$

$A \in \mathbb{C}^{n\times n}$, $C$ full column rank, $s \ll n$

- $A$ large and sparse
- $A$ large and structured: blocks, banded, ...
- $A$ functional: $A = D_1 S^{-1} D_2$, preconditioned, integral, ...
- ....

## The **Block** Arnoldi iteration for $AX = C$

```
[V(1:n,1:s),~]=qr(C,0);
for j=1:m

    jms=(j-1)*s+1; j1s=(j+1)*s; js=j*s;js1=js+1;
    Vp = A*V(:,jms:js);

    %new bases block (modified gram)
    for kk=1:j
      k1=(kk-1)*s+1; k2=kk*s;
      H(k1:k2,jms:js) = V(1:n,k1:k2)'*Vp;         % (j,m) orth coeffs
      Vp = Vp - V(:,k1:k2)*H(k1:k2,jms:js);       % deflation
    end

    [V(1:n,js1:j1s),H(js1:j1s,jms:js)]=qr(Vp,0);  %orth within block
end
```

# The **Block** Conjugate Gradient method

```
$R_0 = B - AX_0,\ P_0 = R_0 \in \mathbb{C}^{n\times s}$
for $i = 0, 1, \dots$
    $\boldsymbol{\alpha}_i = (P_i^* A P_i)^{-1}(R_i^* R_i) \in \mathbb{C}^{s\times s}$
    $X_{i+1} = X_i + P_i \boldsymbol{\alpha}_i$
    $R_{i+1} = R_i - AP_i \boldsymbol{\alpha}_i$
    $\boldsymbol{\beta}_{i+1} = (P_i^* A P_i)^{-1}(R_{i+1}^* A P_i) \in \mathbb{C}^{s\times s}$
    $P_{i+1} = R_i + P_i \boldsymbol{\beta}_{i+1}$
end
```

## Applications

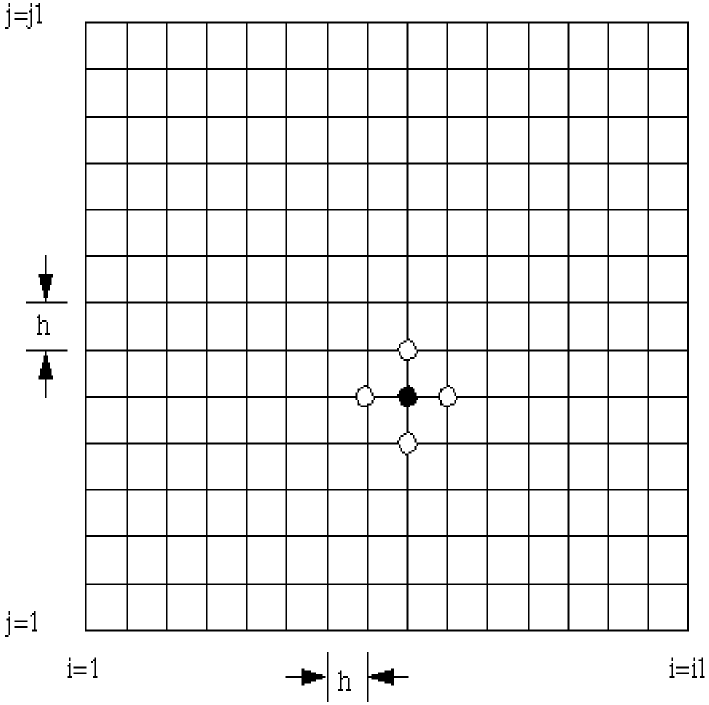
- Eigenvalue problems and tracking
- Control of dynamical systems
- Assignment problems
- Within Riccati equation solvers
- ...

Examples as motivation for using linear matrix equations

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

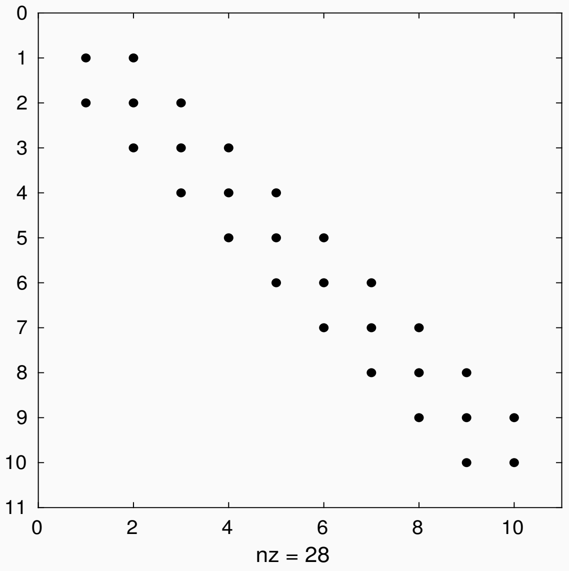
$$A\mathbf{u} = f \qquad A = I \otimes T_1 + T_1 \otimes I, \;\; f = \text{vec}(F),$$

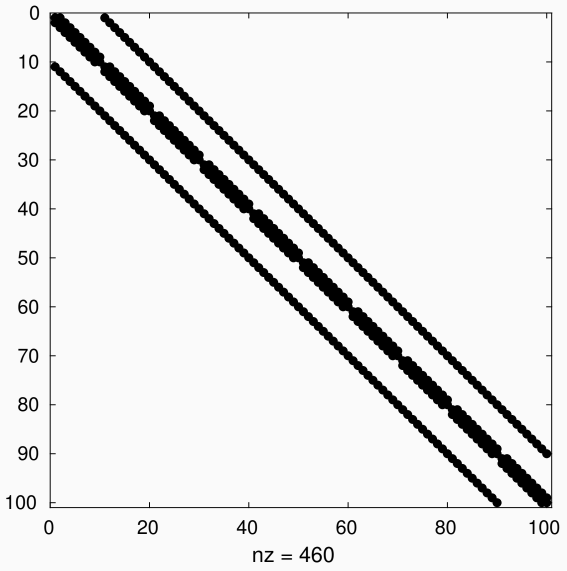
$((M \otimes N)$ Kronecker product, $(M \otimes N) = (M_{i,j} N))$

## Numerical considerations

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

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





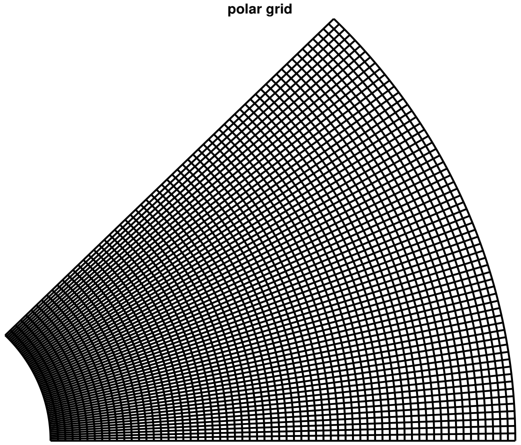
$T_1$ $\qquad$ $A$

## Discretization of more complex domains

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

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

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



♣ 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:

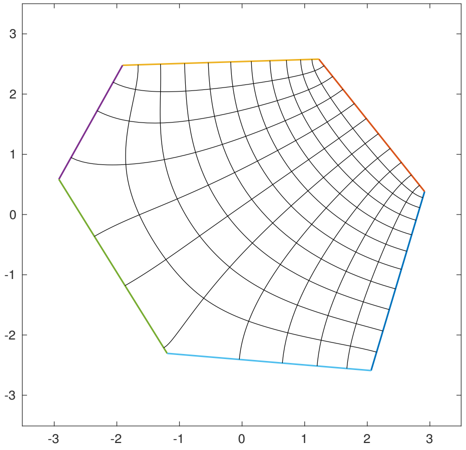
$$\boxed{T\widehat{\boldsymbol{U}} + +\widehat{\boldsymbol{U}}T = \widehat{F}}$$

# Poisson equation in a polygon with more than 4 edges

♣ 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:

$\boxed{T_1U + UT_2 = F}$, $\qquad \widetilde{F}$+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

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

$(x, y) \in \Omega \subset \mathbb{R}^2$, $\phi_i, \psi_i, \gamma_i$, $i = 1, 2$ 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

... A classical approach, Bickley & McNamee, 1960, Wachspress, 1963

(Early literature on difference equations)

# Broader applicability

- Isogeometric Analysis (IGA)
- Certain spectral methods
- Space-Time discretizations
- Parameter dependent problems
- PDEs with stochastic inputs
- PDE-constrained optimization
- etc

# System 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 $\Omega$

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

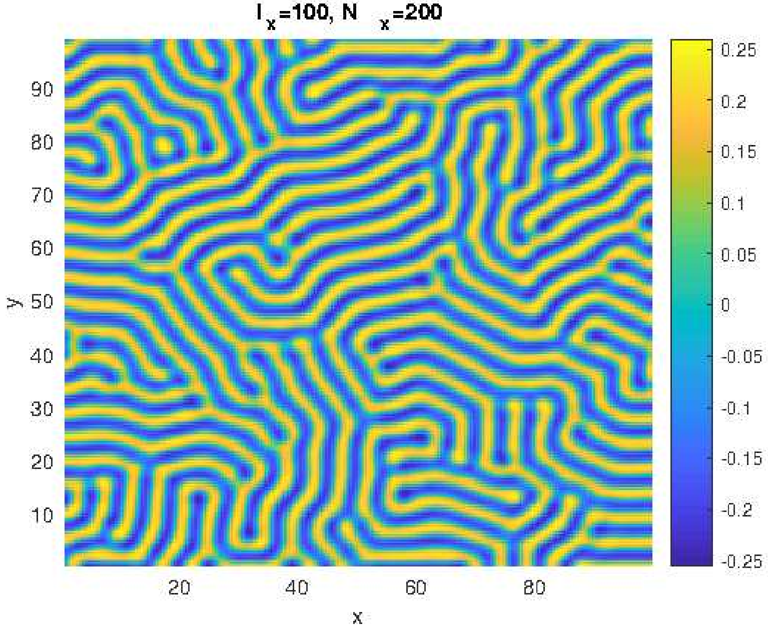
## Applications:

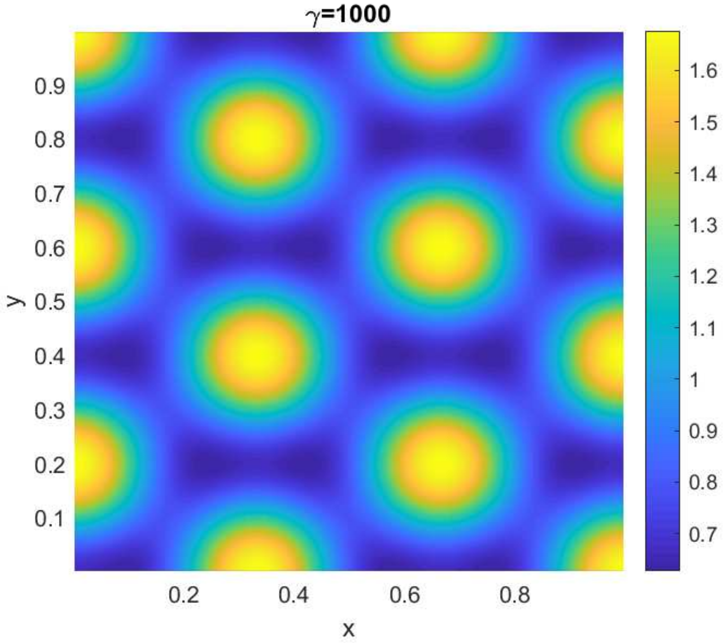
chemistry, biology, ecology, and more recently in metal growth by electrodeposition, tumor growth, biomedicine and cell motility

⇒ 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

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

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

$\Rightarrow A\mathbf{u} = \mathrm{vec}(T_1U + UT_2)$

# 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

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

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

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

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

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

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

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

# 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 $e^{h_t A}\mathbf{u} = \left(e^{h_t T_2^T} \otimes e^{h_t T_1}\right)\mathbf{u} = \text{vec}(e^{h_t T_1} U e^{h_t T_2})$

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

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

## 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)

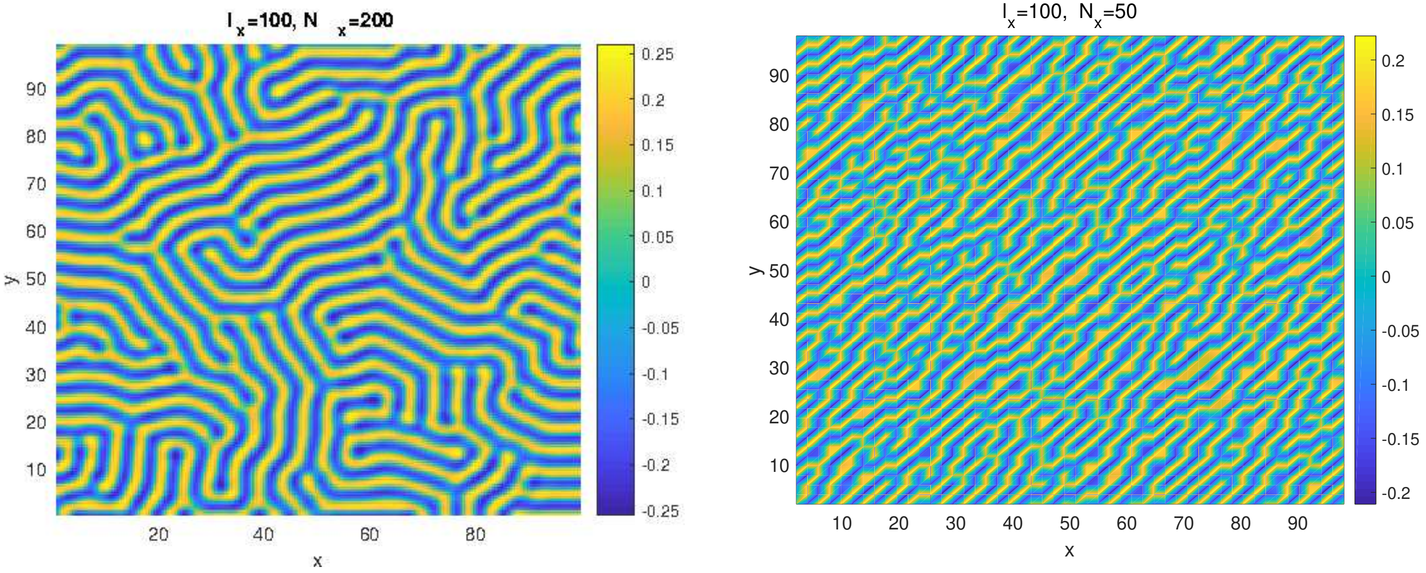
# A numerical example of system of RD-PDEs

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



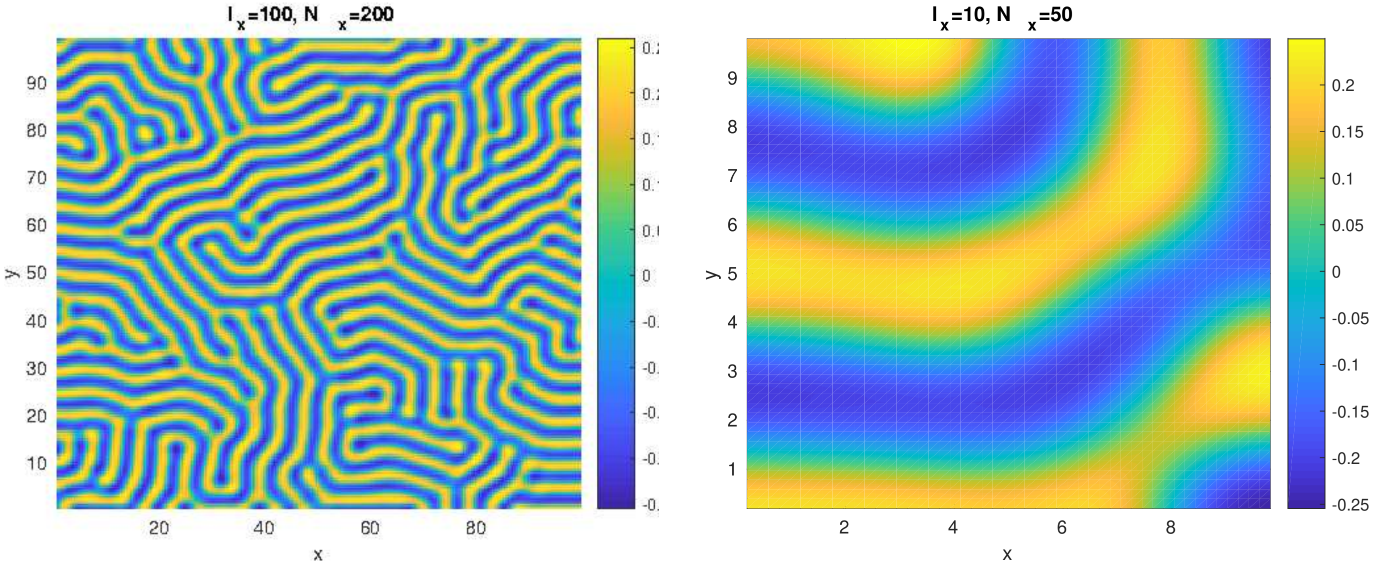
# A numerical example of system of RD-PDEs

Model describing an electrodeposition process for metal growth

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

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

## Turing pattern



REFERENCES

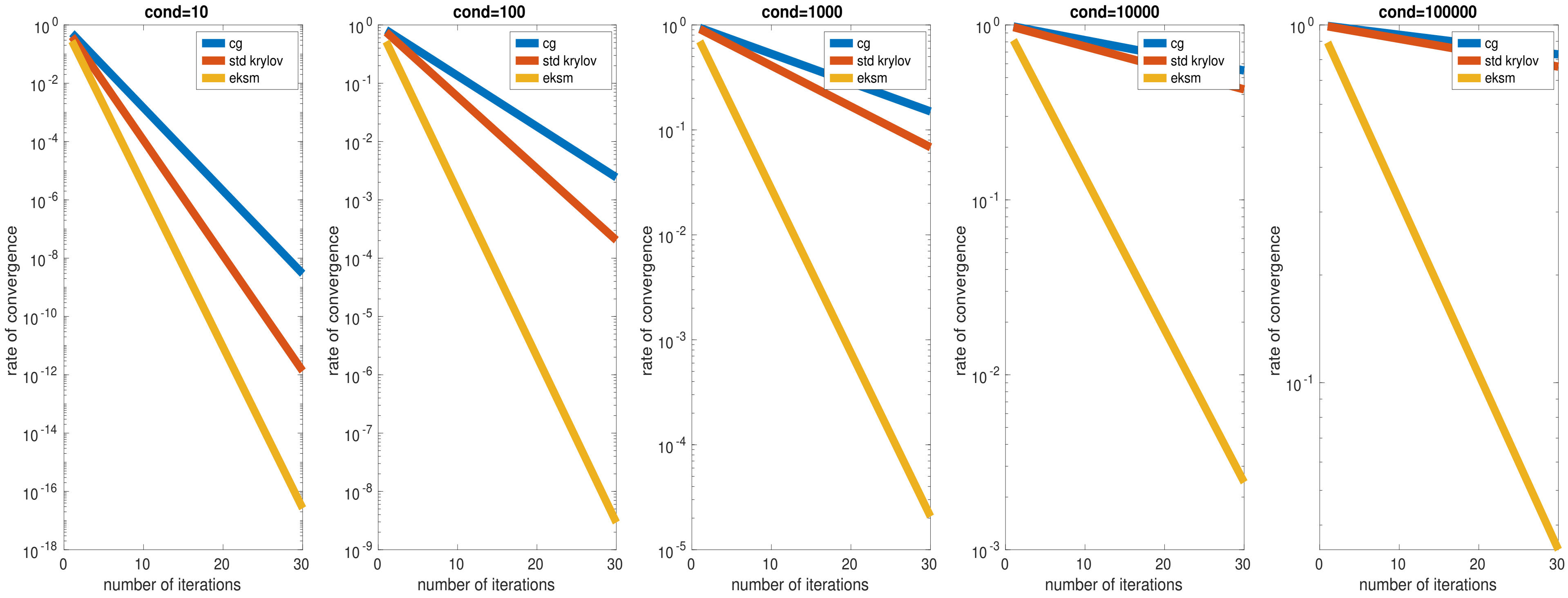
* V.S., *Computational methods for linear matrix equations* SIAM Rev., 58(2016)

* 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.

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

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

# Typical convergence rates



# Typical experimental results

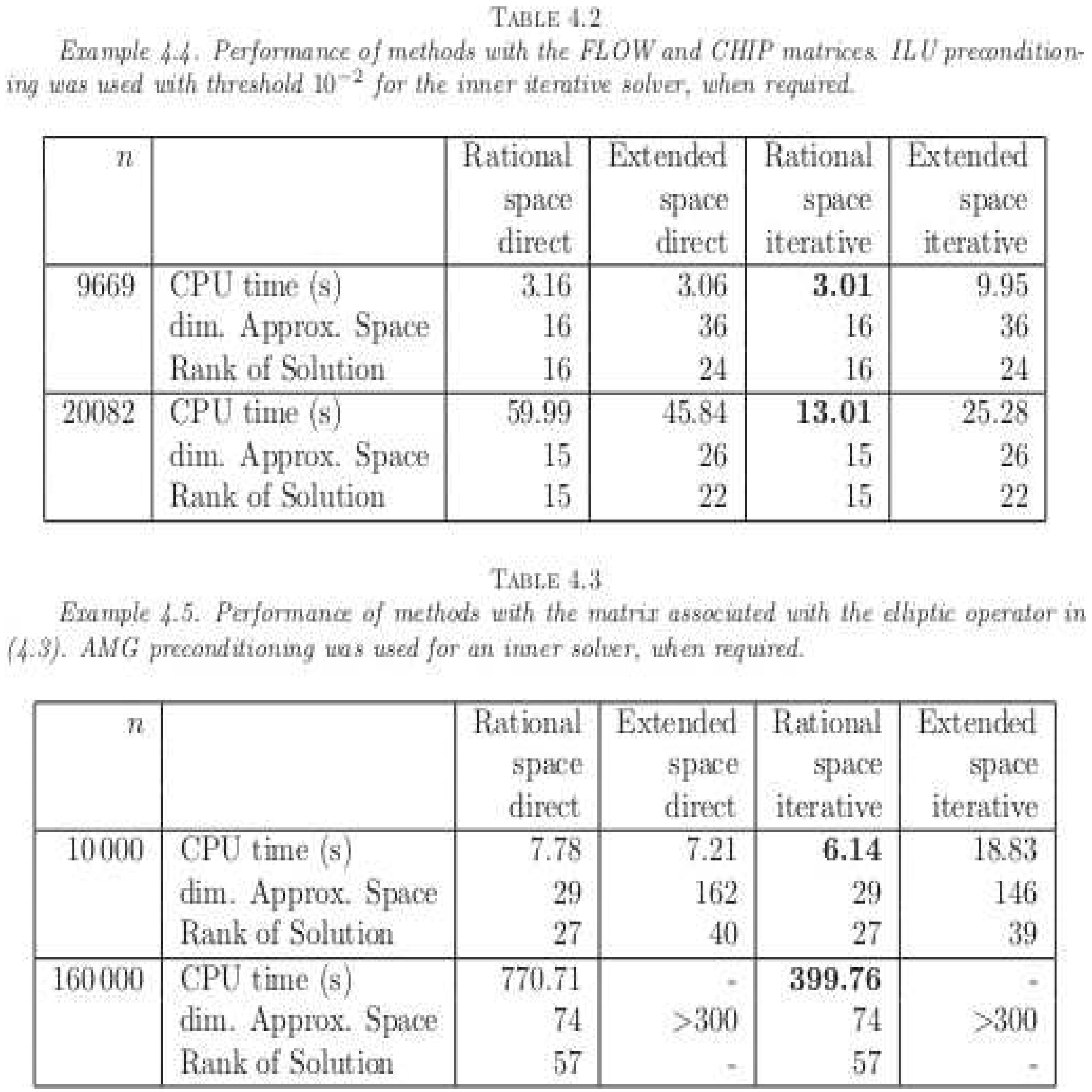


TABLE 4.2

*Example 4.4. Performance of methods with the FLOW and CHIP matrices. ILU preconditioning was used with threshold $10^{-2}$ for the inner iterative solver, when required.*

| $n$ | | Rational space direct | Extended space direct | Rational space iterative | Extended space iterative |
|---|---|---|---|---|---|
| 9669 | CPU time (s) | 3.16 | 3.06 | **3.01** | 9.95 |
| | dim. Approx. Space | 16 | 36 | 16 | 36 |
| | Rank of Solution | 16 | 24 | 16 | 24 |
| 20082 | CPU time (s) | 59.99 | 45.84 | **13.01** | 25.28 |
| | dim. Approx. Space | 15 | 26 | 15 | 26 |
| | Rank of Solution | 15 | 22 | 15 | 22 |

TABLE 4.3

*Example 4.5. Performance of methods with the matrix associated with the elliptic operator in (4.3). AMG preconditioning was used for an inner solver, when required.*

| $n$ | | Rational space direct | Extended space direct | Rational space iterative | Extended space iterative |
|---|---|---|---|---|---|
| 10 000 | CPU time (s) | 7.78 | 7.21 | **6.14** | 18.83 |
| | dim. Approx. Space | 29 | 162 | 29 | 146 |
| | Rank of Solution | 27 | 40 | 27 | 39 |
| 160 000 | CPU time (s) | 770.71 | - | **399.76** | - |
| | dim. Approx. Space | 74 | >300 | 74 | >300 |
| | Rank of Solution | 57 | - | 57 | - |

# CG algorithm for multiterm linear matrix equations

$$\mathcal{L}(X) = C \quad \Leftrightarrow \quad \sum_{i=1}^{\ell} A_i X B_i = C$$

```
Require: Matrix function $\mathcal{L}$: $\mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, right-hand side $C \in \mathbb{R}^{n\times n}$
Ensure: Matrix $X \in \mathbb{R}^{n\times n}$ fulfilling $||\mathcal{L}(X) - C||_F/||C||_F \leq tol$.
$X^{(0)} = 0$, $R^{(0)} = C$, $P^{(0)} = R^{(0)}$, $Q^{(0)} = \mathcal{L}(P^{(0)})$
$\xi_0 =< P^{(0)}, Q^{(0)} >$, $k = 0$
while $||R_k||_F > tol$ do do
    $\omega_k =< R^{(k)}, P^{(k)} > /\xi_k$
    $X^{(k+1)} = X^{(k)} + \omega_k P^{(k)}$
    $R^{(k+1)} = C - \mathcal{L}(X^{(k+1)})$
    $\beta_k = - < R^{(k+1)}, Q^{(k)} > /\xi_k$
    $P^{(k+1)} = R^{(k+1)} + \beta_k P^{(k)}$
    $Q^{(k+1)} = \mathcal{L}(P^{(k+1)})$
    $\xi_{k+1} =< P^{(k+1)}, Q^{(k+1)} >$
    $k = k + 1$
end while
$X = X^{(k)}$
```

# TCG algorithm for multiterm linear matrix equations

$$\mathcal{L}(X) = C \quad \Leftrightarrow \quad \sum_{i=1}^{\ell} A_i X B_i = C$$

```
Require: Matrix function $\mathcal{L}$: $\mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$, right-hand side $C \in \mathbb{R}^{n \times n}$ in low-rank format.
    Truncation operator $\mathcal{T}$.
Ensure: Matrix $X \in \mathbb{R}^{n \times n}$ fulfilling $||\mathcal{L}(X) - C||_F / ||C||_F \leq tol$.
$X^{(0)} = 0$, $R^{(0)} = C$, $P^{(0)} = R^{(0)}$, $Q^{(0)} = \mathcal{L}(P^{(0)})$
$\xi_0 =< P^{(0)}, Q^{(0)} >$, $k = 0$
while $||R_k||_F > tol$ do do
    $\omega_k =< R^{(k)}, P^{(k)} > /\xi_k$
    $X^{(k+1)} = X^{(k)} + \omega_k P^{(k)}$                 $X^{(k+1)} \leftarrow \mathcal{T}(X^{(k+1)})$
    $R^{(k+1)} = C - \mathcal{L}(X^{(k+1)})$               Optionally: $R^{(k+1)} \leftarrow \mathcal{T}(R^{(k+1)})$
    $\beta_k = - < R^{(k+1)}, Q^{(k)} > /\xi_k$
    $P^{(k+1)} = R^{(k+1)} + \beta_k P^{(k)}$                $P^{(k+1)} \leftarrow \mathcal{T}(P^{(k+1)})$
    $Q^{(k+1)} = \mathcal{L}(P^{(k+1)})$                   Optionally: $Q^{(k+1)} \leftarrow \mathcal{T}(Q^{(k+1)})$
    $\xi_{k+1} =< P^{(k+1)}, Q^{(k+1)} >$
    $k = k + 1$
end while
$X = X^{(k)}$
```

$\Rightarrow X^{(k)} = X_{k,1} X_{k,2}^{\top}$ low rank format! Similarly for all involved matrices

Kressner-Tobler, 2011-2012.