On time-dependent matrix-oriented differential problems

Valeria Simoncini

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

Joint works with
M.C. D'Autilia & I. Sgura, Univ. Lecce
J. Henning, K. Urban, Ulm Uni.(D), D. Palitta, Univ. Bologna

The differential problem

We are interested in solving

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^2,\ t \in \mathcal{T}$$

with given initial conditions $u(x,y,0) = u_0(x,y)$ and proper b.c.

- $\mathcal{L}$ linear in u (typically 2ˆorder diff operator in space, w/separable coeffs)
- f nonlinear function in u

Discretization: use tensor bases
(finite differences, conformal mappings, IGA, spectral methods, etc.)

The differential problem

We are interested in solving

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^2,\ t \in \mathcal{T}$$

with given initial conditions $u(x,y,0) = u_0(x,y)$ and proper b.c.

- $\mathcal{L}$ linear in u (typically 2ˆorder diff operator in space, w/separable coeffs)
- f nonlinear function in u

Discretization: use tensor bases
(finite differences, conformal mappings, IGA, spectral methods, etc.)

Approaches to time discretization

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^d,\ d = 2,3,\ t \in \mathcal{T}$$

- **Time marching schemes:** classical strategies stemming from ODEs
  Lead to

Sequence of (matrix) space problems at subsequent time steps

- **All-at-once schemes:** time discretization similar to space discretization (tensor basis methods)

Lead to

(Non)linear matrix equations

This presentation: exploit matrix-matrix computations throughout the time evolution

Approaches to time discretization

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^d,\ d = 2,3,\ t \in \mathcal{T}$$

- **Time marching schemes:** classical strategies stemming from ODEs
  Lead to

Sequence of (matrix) space problems at subsequent time steps

- **All-at-once schemes:** time discretization similar to space discretization (tensor basis methods)

Lead to

(Non)linear matrix equations

This presentation: exploit matrix-matrix computations throughout the time evolution

Approaches to time discretization

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^d,\ d = 2,3,\ t \in \mathcal{T}$$

- **Time marching schemes:** classical strategies stemming from ODEs
  Lead to

Sequence of (matrix) space problems at subsequent time steps

- **All-at-once schemes:** time discretization similar to space discretization (tensor basis methods)

Lead to

(Non)linear matrix equations

This presentation: exploit matrix-matrix computations throughout the time evolution

Time marching scheme. Matrix-oriented discretization.

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^2,\ t \in \mathcal{T}$$

Linear operator:

$$\boxed{\mathcal{L}(u) = \Delta u}$$

Standard (vector) discretization in space, $n_x \times n_y$ grid:

- $\Delta u \Rightarrow \mathcal{A}\boldsymbol{u} \qquad \mathcal{A} \in \mathbb{R}^{n_x n_y \times n_x n_y}$
- $f(u,t) \Rightarrow \boldsymbol{f}(\boldsymbol{u},t) \qquad$ ($n_x n_y$ components, evaluated component-wise)

with lexicographic ordering of the rectangle nodes

Matrix-oriented discretization in space:

- $\Delta u \Rightarrow A\boldsymbol{U} + \boldsymbol{U}B, \quad A \in \mathbb{R}^{n_x \times n_x},\ B \in \mathbb{R}^{n_y \times n_y}, (\boldsymbol{U})_{ij} \approx u(x_i, y_j)$
  with $\mathcal{A} = I \otimes A + B^\top \otimes I$
- $f(u,t) \Rightarrow \mathcal{F}(\boldsymbol{U},t)$ ($n_x \times n_y$, evaluated component-wise)

Time marching scheme. Matrix-oriented discretization.

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^2,\ t \in \mathcal{T}$$

Linear operator:

$$\boxed{\mathcal{L}(u) = \Delta u}$$

Standard (vector) discretization in space, $n_x \times n_y$ grid:

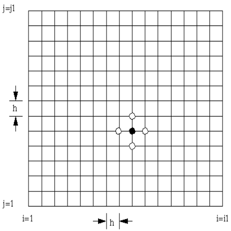
- $\Delta u \Rightarrow \mathcal{A}\boldsymbol{u} \qquad \mathcal{A} \in \mathbb{R}^{n_x n_y \times n_x n_y}$
- $f(u,t) \Rightarrow \boldsymbol{f}(\boldsymbol{u},t) \qquad$ ($n_x n_y$ components, evaluated component-wise)

with lexicographic ordering of the rectangle nodes

Matrix-oriented discretization in space:

- $\Delta u \Rightarrow A\boldsymbol{U} + \boldsymbol{U}B, \quad A \in \mathbb{R}^{n_x \times n_x},\ B \in \mathbb{R}^{n_y \times n_y}, (\boldsymbol{U})_{ij} \approx u(x_i, y_j)$
  with $\mathcal{A} = I \otimes A + B^\top \otimes I$
- $f(u,t) \Rightarrow \mathcal{F}(\boldsymbol{U},t)$ ($n_x \times n_y$, evaluated component-wise)

Reminder: matrix formulation of tensor discretization



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

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

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

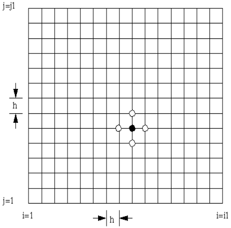
Let $T = \frac{1}{h^2}\mathrm{tridiag}(-1, \underline{2}, -1)$. Collecting all nodes together,

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

Therefore, directly from the grid,

$$-u_{xx} - u_{yy} \quad \Rightarrow \quad TU + UT + b.c.$$

Reminder: matrix formulation of tensor discretization



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

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

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

Let $T = \frac{1}{h^2}\mathrm{tridiag}(-1, \underline{2}, -1)$. Collecting all nodes together,

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

Therefore, directly from the grid,

$$-u_{xx} - u_{yy} \qquad \Rightarrow \qquad TU + UT + b.c.$$

The matrix differential equation

$$\dot{\boldsymbol{U}}(t) = A\boldsymbol{U}(t) + \boldsymbol{U}(t)B + \mathcal{F}(\boldsymbol{U}, t), \quad \boldsymbol{U}(0) = \boldsymbol{U}_0$$

Computational strategies. Time stepping methods:

- **Small scale:** matrix-oriented IMEX methods, exponential integrators
- **Large scale:** In sequence:
  1. Order reduction procedure ($\Rightarrow$ POD-type)
  2. Feasible handling of nonlinear term $\mathcal{F}(\boldsymbol{U}, t)$ ($\Rightarrow$ matrix DEIM)
  3. Time stepping of reduced problem

The matrix differential equation

$$\dot{\boldsymbol{U}}(t) = A\boldsymbol{U}(t) + \boldsymbol{U}(t)B + \mathcal{F}(\boldsymbol{U}, t), \quad \boldsymbol{U}(0) = \boldsymbol{U}_0$$

Computational strategies. Time stepping methods:

- **Small scale:** matrix-oriented IMEX methods, exponential integrators
- **Large scale:** In sequence:
  1. Order reduction procedure ($\Rightarrow$ POD-type)
  2. Feasible handling of nonlinear term $\mathcal{F}(\boldsymbol{U}, t)$ ($\Rightarrow$ matrix DEIM)
  3. Time stepping of reduced problem

The matrix differential equation

$$\dot{\boldsymbol{U}}(t) = A\boldsymbol{U}(t) + \boldsymbol{U}(t)B + \mathcal{F}(\boldsymbol{U}, t), \quad \boldsymbol{U}(0) = \boldsymbol{U}_0$$

Computational strategies. Time stepping methods:

- **Small scale:** matrix-oriented IMEX methods, exponential integrators
- **Large scale:** In sequence:
  1. Order reduction procedure ($\Rightarrow$ POD-type)
  2. Feasible handling of nonlinear term $\mathcal{F}(\boldsymbol{U}, t)$ ($\Rightarrow$ matrix DEIM)
  3. Time stepping of reduced problem

Small scale time stepping

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^2,\ t \in \mathcal{T}$$

- Problem is stiff
  - Use appropriate time discretizations
  - Time stepping constraints
- Possibly long time period (e.g., for pattern detection), with occurrence of transient unstable phase
- Phenomenon sets in only if domain is well represented

$$\dot{\boldsymbol{U}}(t) = A\boldsymbol{U}(t) + \boldsymbol{U}(t)B + \mathcal{F}(\boldsymbol{U},t), \quad \boldsymbol{U}(0) = \boldsymbol{U}_0$$

Time stepping Matrix-oriented methods

IMEX methods

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

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

Matrix-oriented form: $U_{n+1} - U_n = h_t(AU_{n+1} + U_{n+1}B) + h_t F(U_n)$, so that

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

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

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

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

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

Time stepping Matrix-oriented methods

IMEX methods

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

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

Matrix-oriented form: $U_{n+1} - U_n = h_t(AU_{n+1} + U_{n+1}B) + h_t F(U_n)$, so that

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

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

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

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

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

Time stepping Matrix-oriented methods

IMEX methods

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

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

Matrix-oriented form: $U_{n+1} - U_n = h_t(AU_{n+1} + U_{n+1}B) + h_t F(U_n)$,
so that

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

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

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

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

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

Time stepping Matrix-oriented methods

IMEX methods

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

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

Matrix-oriented form: $U_{n+1} - U_n = h_t(AU_{n+1} + U_{n+1}B) + h_t F(U_n)$,
so that

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

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

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

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

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

Time stepping Matrix-oriented methods

Exponential integrator

Exponential first order Euler method:

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

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

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

Matrix-oriented form: since $e^{h_t\mathcal{A}}\boldsymbol{u} = \left(e^{h_tB^\top} \otimes e^{h_tA}\right)\boldsymbol{u} = \mathrm{vec}(e^{h_tA}\boldsymbol{U}e^{h_tB})$

1. Compute $E_1 = e^{h_tA}$, $E_2 = e^{h_tB^\top}$
2. For each n

$$\begin{array}{ll} \text{Solve} & A\mathbf{V}_n + \mathbf{V}_nB = E_1F(\boldsymbol{U}_n)E_2^\top - F(\boldsymbol{U}_n) \\ \text{Compute} & \boldsymbol{U}_{n+1} = E_1\boldsymbol{U}_nE_2^\top + h_tV_n \end{array}$$

Time stepping Matrix-oriented methods

Exponential integrator

Exponential first order Euler method:

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

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

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

Matrix-oriented form: since $e^{h_t \mathcal{A}} \boldsymbol{u} = \left(e^{h_t B^\top} \otimes e^{h_t A} \right) \boldsymbol{u} = \mathrm{vec}(e^{h_t A} \boldsymbol{U} e^{h_t B})$

1. Compute $E_1 = e^{h_t A}$, $E_2 = e^{h_t B^\top}$
2. For each n

$$\begin{aligned} \text{Solve} \qquad & A\mathbf{V}_n + \mathbf{V}_n B = E_1 F(\boldsymbol{U}_n) E_2^\top - F(\boldsymbol{U}_n) \\ \text{Compute} \qquad & \boldsymbol{U}_{n+1} = E_1 \boldsymbol{U}_n E_2^\top + h_t V_n \end{aligned}$$

Time stepping Matrix-oriented methods

Computational issues:

- Dimensions of A, B very modest
- A, B quasi-symmetric (non-symmetry due to bc's)
- A, B do not depend on time step

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

Numerical properties:

Structural properties are preserved

Time stepping Matrix-oriented methods

Computational issues:

- Dimensions of A, B very modest
- A, B quasi-symmetric (non-symmetry due to bc's)
- A, B do not depend on time step

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

Numerical properties:

Structural properties are preserved

A numerical example of system of RD-PDEs

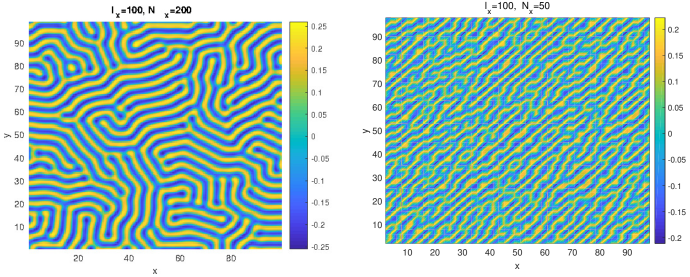
$$\begin{cases} u_t = \mathcal{L}_1(u) + f_1(u,v), \\ v_t = \mathcal{L}_2(v) + f_2(u,v), \quad \text{with} \quad (x,y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0,T] \end{cases}$$

Model describing an electrodeposition process for metal growth

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

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

Turing pattern



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

A numerical example of system of RD-PDEs

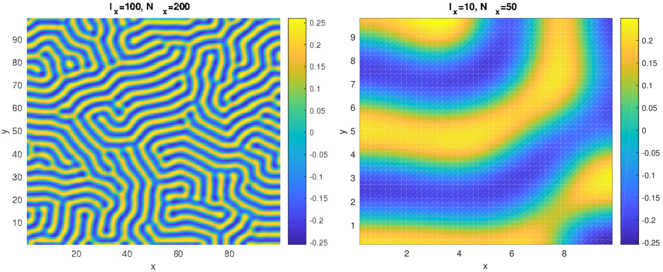
$$\begin{cases} u_t = \mathcal{L}_1(u) + f_1(u,v), \\ v_t = \mathcal{L}_2(v) + f_2(u,v), \quad \text{with} \quad (x,y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0, T] \end{cases}$$

Model describing an electrodeposition process for metal growth

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

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

Turing pattern



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

Approaches to time discretization

$$u_t = \mathcal{L}(u) + f(u,t), \quad u = u(x,y,t) \quad \text{with } (x,y) \in \Omega \subset \mathbb{R}^d,\ d = 2,3,\ t \in \mathcal{T}$$

- **Time marching schemes:** classical strategies stemming from ODEs
  Lead to

Sequence of (matrix) space problems at subsequent time steps

- **All-at-once schemes:** time discretization similar to space discretization (tensor basis methods)

Lead to

(Non)linear matrix equations

Space-Time discretizations via tensorized high order methods

- The heat equation:

$$u_t + \mathcal{L}(u) = f, \qquad u(0) = 0, \qquad f \in L_2(I, X')$$

$\mathcal{L} : X \to X'$ elliptic op. with coercive bilinear form $a : X \times X \to \mathbb{R}$, $X \equiv H_0^1(\Omega)$

Variational formulation:

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

where

trial: $U := H^1_{(0)}(I; X') \cap L_2(I; X)$ **test**: $V := L_2(I; X)$

$b(u,v) := \int_0^T \int_\Omega u_t(t,x)\, v(t,x)\, dx\, dt + \int_0^T a(u(t), v(t))\, dt$ $\langle f, v \rangle := \int_0^T \int_\Omega f(t,x)\, v(t,x)\, dx\, dt$

- The wave equation:

$$u_{tt} + \mathcal{L}(u) = f, \qquad u(0) = 0, u_t(0) = 0$$

Variational formulation:

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

trial: $U := L_2(I; H)$ **test**: $V := \{v \in L_2(I; H) : v_{tt} + \mathcal{L}(v) \in L_2(I; H), v(T) = v_t(T) = 0\}$

$b(u,v) := (u, v_{tt} + \mathcal{L}(v))_{L_2(I;H)}$

J. Henning, D. Palitta, V. S., and K. Urban, 2020, 2022.

Space-Time discretizations via tensorized high order methods

- The heat equation:

$$u_t + \mathcal{L}(u) = f, \qquad u(0) = 0, \qquad f \in L_2(I, X')$$

$\mathcal{L} : X \to X'$ elliptic op. with coercive bilinear form $a : X \times X \to \mathbb{R}$, $X \equiv H_0^1(\Omega)$

Variational formulation:

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

where

trial: $U := H^1_{(0)}(I; X') \cap L_2(I; X)$ **test**: $V := L_2(I; X)$

$b(u,v) := \int_0^T \int_\Omega u_t(t,x)\, v(t,x)\, dx\, dt + \int_0^T a(u(t), v(t))\, dt \quad \langle f, v\rangle := \int_0^T \int_\Omega f(t,x)\, v(t,x)\, dx\, dt$

- The wave equation:

$$u_{tt} + \mathcal{L}(u) = f, \qquad u(0) = 0, u_t(0) = 0$$

Variational formulation:

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

trial: $U := L_2(I; H)$ **test**: $V := \{v \in L_2(I; H) : v_{tt} + \mathcal{L}(v) \in L_2(I; H),\ v(T) = v_t(T) = 0\}$

$b(u,v) := (u, v_{tt} + \mathcal{L}(v))_{L_2(I;H)}$

J. Henning, D. Palitta, V. S., and K. Urban, 2020, 2022.

Wave equation. Petrov-Galerkin discretization

For trial space $U_\delta \subset U$, and test space $V_\delta \subset V$,

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

- Finite elements in time with, e.g., piecewise quadratic splines
- Conformal finite element space, e.g., piecewise quadratic B-splines

Test space: as the tensor product space[1]

$$V_\delta := R_{\Delta t} \otimes Z_h = span\{\varphi_\nu := \varrho^k \otimes \phi_i : \; k = 1, ..., N_t, \; i = 1, ..., N_h, \nu = (k, i)\}$$

Trial space: apply the adjoint operator B^* to each test basis function, i.e., for $\mu = (\ell, j)$ and $\mathcal{L} = -\Delta$

$$\psi_\mu := B^*(\varphi_\mu) = B^*(\varrho^\ell \otimes \phi_j) = (\partial_{tt} + \mathcal{L})(\varrho^\ell \otimes \phi_j) = \ddot{\varrho}^\ell \otimes \phi_j + \varrho^\ell \otimes \mathcal{L}(\phi_j)$$

i.e., (inf-sup-optimal space)

$$U_\delta := B^*(V_\delta) = span\{\psi_\mu : \; \mu = 1, ..., \mathcal{N}_\delta\}$$

[1] $R_{\Delta t} := span\{\varrho^1, ..., \varrho^{N_t}\} \subset H^2_{\{T\}}(I), Z_h := span\{\phi_1, ..., \phi_{N_h}\} \subset H^1_0(\Omega) \cap H^2(\Omega)$

Wave equation. Petrov-Galerkin discretization

For trial space $U_\delta \subset U$, and test space $V_\delta \subset V$,

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

- Finite elements in time with, e.g., piecewise quadratic splines
- Conformal finite element space, e.g., piecewise quadratic B-splines

Test space: as the tensor product space[1]

$$V_\delta := R_{\Delta t} \otimes Z_h = span\{\varphi_\nu := \varrho^k \otimes \phi_i : \ k = 1, ..., N_t, \ i = 1, ..., N_h, \nu = (k, i)\}$$

Trial space: apply the adjoint operator B^* to each test basis function, i.e., for $\mu = (\ell, j)$ and $\mathcal{L} = -\Delta$

$$\psi_\mu := B^*(\varphi_\mu) = B^*(\varrho^\ell \otimes \phi_j) = (\partial_{tt} + \mathcal{L})(\varrho^\ell \otimes \phi_j) = \ddot{\varrho}^\ell \otimes \phi_j + \varrho^\ell \otimes \mathcal{L}(\phi_j)$$

i.e., (inf-sup-optimal space)

$$U_\delta := B^*(V_\delta) = span\{\psi_\mu : \ \mu = 1, ..., \mathcal{N}_\delta\}$$

[1] $R_{\Delta t} := span\{\varrho^1, ..., \varrho^{N_t}\} \subset H^2_{\{T\}}(I), Z_h := span\{\phi_1, ..., \phi_{N_h}\} \subset H^1_0(\Omega) \cap H^2(\Omega)$

The linear system. The stiffness matrix

For spaces induced by :
trial: $\{\psi_\mu := \sigma^\ell \otimes \xi_j : \mu = 1, ..., \mathcal{N}_\delta\}$
test: $\{\varphi_\nu = \varrho^k \otimes \phi_i : \nu = 1, ..., \mathcal{N}_\delta\}$
In the inf-sup optimal case $\psi_\mu = B^*(\varphi_\mu)$, we get the representation

$$\begin{aligned}[\mathbb{B}_\delta]_{(\ell,j),(k,i)} &= (\ddot{\varrho}^\ell \otimes \phi_j + \varrho^\ell \otimes \mathcal{L}(\phi_j), \ddot{\varrho}^k \otimes \phi_i + \varrho^k \otimes \mathcal{L}(\phi_i))_\mathcal{H} \\ &= (\ddot{\varrho}^\ell, \ddot{\varrho}^k)_{L_2(I)} (\phi_j, \phi_i)_{L_2(\Omega)} + (\varrho^\ell, \varrho^k)_{L_2(I)} (\mathcal{L}(\phi_j), \mathcal{L}(\phi_i))_{L_2(\Omega)} \\ &\quad + (\ddot{\varrho}^\ell, \varrho^k)_{L_2(I)} (\phi_j, \mathcal{L}(\phi_i))_{L_2(\Omega)} + (\varrho^\ell, \ddot{\varrho}^k)_{L_2(I)} (\mathcal{L}(\phi_j), \phi_i)_{L_2(\Omega)}\end{aligned}$$

so that

$$\mathbb{B}_\delta = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{N}_h^\top + {\boldsymbol{N}_{\Delta t}}^\top \otimes \boldsymbol{N}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h, \text{ for } \psi_\mu = B^*(\varphi_\mu),$$

Note: $\mathbb{B}_\delta$ is symmetric and positive definite for $\mathcal{L} = -\Delta$

Solving $\mathbb{B}_\delta \boldsymbol{u}_\delta = \boldsymbol{g}_\delta$ yields the expansion coefficients of $u_\delta \in \mathbb{U}_\delta$

The stiffness matrix

$$\mathbb{B}_\delta = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{N}_h^\top + \boldsymbol{N}_{\Delta t}^\top \otimes \boldsymbol{N}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h, \text{ with } \psi_\mu = B^*(\varphi_\mu)$$

$\mathbb{B}_\delta$ is sum of tensor products involving some ill-conditioned components

$\kappa(\mathbb{B}_\delta)$ scales like a stiffness matrix of a 4th order problem

$\Rightarrow$ ill-conditioned linear system

$\Rightarrow$ Matrices are generally dense

Iterative solver with structure-aware preconditioning

In particular, $\boldsymbol{Q}_h$ and[2] $\boldsymbol{N}_h \boldsymbol{M}_h^{-1} \boldsymbol{N}_h^\top$ are spectrally equivalent, i.e.,

$$\gamma^2 \, \boldsymbol{z}_h^\top \boldsymbol{Q}_h \boldsymbol{z}_h \leq \boldsymbol{z}_h^\top \boldsymbol{N}_h \boldsymbol{M}_h^{-1} \boldsymbol{N}_h^\top \boldsymbol{z}_h \leq \Gamma^2 \, \boldsymbol{z}_h^\top \boldsymbol{Q}_h \boldsymbol{z}_h \quad \text{for all } \boldsymbol{z}_h \in \mathbb{R}^{N_h}.$$

[2] with some abuse of notation for spaces...

The stiffness matrix

$$\mathbb{B}_\delta = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{N}_h^\top + \boldsymbol{N}_{\Delta t}^\top \otimes \boldsymbol{N}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h, \text{ with } \psi_\mu = B^*(\varphi_\mu)$$

$\mathbb{B}_\delta$ is sum of tensor products involving some ill-conditioned components

$\kappa(\mathbb{B}_\delta)$ scales like a stiffness matrix of a 4th order problem

$\Rightarrow$ ill-conditioned linear system

$\Rightarrow$ Matrices are generally dense

Iterative solver with structure-aware preconditioning

In particular, $\boldsymbol{Q}_h$ and[2] $\boldsymbol{N}_h \boldsymbol{M}_h^{-1} \boldsymbol{N}_h^\top$ are spectrally equivalent, i.e.,

$$\gamma^2 \, \boldsymbol{z}_h^\top \boldsymbol{Q}_h \boldsymbol{z}_h \leq \boldsymbol{z}_h^\top \boldsymbol{N}_h \boldsymbol{M}_h^{-1} \boldsymbol{N}_h^\top \boldsymbol{z}_h \leq \Gamma^2 \, \boldsymbol{z}_h^\top \boldsymbol{Q}_h \boldsymbol{z}_h \qquad \text{for all } \boldsymbol{z}_h \in \mathbb{R}^{N_h}.$$

[2] with some abuse of notation for spaces...

The stiffness matrix

$$\mathbb{B}_\delta = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{N}_h^\top + \boldsymbol{N}_{\Delta t}^\top \otimes \boldsymbol{N}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h, \text{ with } \psi_\mu = B^*(\varphi_\mu)$$

$\mathbb{B}_\delta$ is sum of tensor products involving some ill-conditioned components

$\kappa(\mathbb{B}_\delta)$ scales like a stiffness matrix of a 4th order problem

$\Rightarrow$ ill-conditioned linear system

$\Rightarrow$ Matrices are generally dense

Iterative solver with structure-aware preconditioning

In particular, $\boldsymbol{Q}_h$ and[2] $\boldsymbol{N}_h \boldsymbol{M}_h^{-1} \boldsymbol{N}_h^\top$ are spectrally equivalent, i.e.,

$$\gamma^2 \, \boldsymbol{z}_h^\top \boldsymbol{Q}_h \boldsymbol{z}_h \le \boldsymbol{z}_h^\top \boldsymbol{N}_h \boldsymbol{M}_h^{-1} \boldsymbol{N}_h^\top \boldsymbol{z}_h \le \Gamma^2 \, \boldsymbol{z}_h^\top \boldsymbol{Q}_h \boldsymbol{z}_h \quad \text{for all } \boldsymbol{z}_h \in \mathbb{R}^{N_h}.$$

[2]with some abuse of notation for spaces...

Preconditioned Conjugate Gradients method

$$\mathbb{B}_\delta = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{N}_h^\top + \boldsymbol{N}_{\Delta t}^\top \otimes \boldsymbol{N}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h, \text{ with } \psi_\mu = B^*(\varphi_\mu)$$

$$\mathbb{B}_\delta \boldsymbol{u}_\delta = \boldsymbol{g}_\delta$$

$$\mathbb{P}^{-1}\mathbb{B}\boldsymbol{u}_\delta = \mathbb{P}^{-1}\boldsymbol{g}_\delta$$

Preconditioners

- Sylvester operator preconditioner

$$\mathbb{P} = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h$$

- Spectrally equivalent preconditioner

$$\mathbb{P} = \mathbb{K}_\delta^\top \mathbb{M}_\delta^{-1} \mathbb{K}_\delta \qquad \mathbb{K}_\delta = \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{N}_h, \quad \mathbb{M}_\delta = \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{M}_h$$

Matrix-oriented formulation of PCG:

$$\mathcal{A}(\boldsymbol{U}) = \boldsymbol{G} \quad \text{with} \quad \mathcal{A}(\boldsymbol{U}) = \boldsymbol{M}_h \boldsymbol{U} \boldsymbol{Q}_{\Delta t}^\top + \boldsymbol{N}_h^\top \boldsymbol{U} \boldsymbol{N}_{\Delta t}^\top + \boldsymbol{N}_h \boldsymbol{U} \boldsymbol{N}_{\Delta t} + \boldsymbol{Q}_h \boldsymbol{U} \boldsymbol{M}_{\Delta t}$$

Preconditioned Conjugate Gradients method

$$\mathbb{B}_\delta = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{N}_h^\top + \boldsymbol{N}_{\Delta t}^\top \otimes \boldsymbol{N}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h, \text{ with } \psi_\mu = B^*(\varphi_\mu)$$

$$\mathbb{B}_\delta \boldsymbol{u}_\delta = \boldsymbol{g}_\delta$$

$$\mathbb{P}^{-1}\mathbb{B}\boldsymbol{u}_\delta = \mathbb{P}^{-1}\boldsymbol{g}_\delta$$

Preconditioners

- Sylvester operator preconditioner

$$\mathbb{P} = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h$$

- Spectrally equivalent preconditioner

$$\mathbb{P} = \mathbb{K}_\delta^\top \mathbb{M}_\delta^{-1} \mathbb{K}_\delta \qquad \mathbb{K}_\delta = \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{N}_h, \quad \mathbb{M}_\delta = \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{M}_h$$

Matrix-oriented formulation of PCG:

$$\mathcal{A}(\boldsymbol{U}) = \boldsymbol{G} \quad \text{with} \quad \mathcal{A}(\boldsymbol{U}) = \boldsymbol{M}_h \boldsymbol{U} \boldsymbol{Q}_{\Delta t}^\top + \boldsymbol{N}_h^\top \boldsymbol{U} \boldsymbol{N}_{\Delta t}^\top + \boldsymbol{N}_h \boldsymbol{U} \boldsymbol{N}_{\Delta t} + \boldsymbol{Q}_h \boldsymbol{U} \boldsymbol{M}_{\Delta t}$$

Preconditioned Conjugate Gradients method

$$\mathbb{B}_\delta = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{N}_h^\top + \boldsymbol{N}_{\Delta t}^\top \otimes \boldsymbol{N}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h, \text{ with } \psi_\mu = B^*(\varphi_\mu)$$

$$\mathbb{B}_\delta \boldsymbol{u}_\delta = \boldsymbol{g}_\delta$$

$$\mathbb{P}^{-1}\mathbb{B}\boldsymbol{u}_\delta = \mathbb{P}^{-1}\boldsymbol{g}_\delta$$

Preconditioners

- Sylvester operator preconditioner

$$\mathbb{P} = \boldsymbol{Q}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{Q}_h$$

- Spectrally equivalent preconditioner

$$\mathbb{P} = \mathbb{K}_\delta^\top \mathbb{M}_\delta^{-1} \mathbb{K}_\delta \qquad \mathbb{K}_\delta = \boldsymbol{N}_{\Delta t} \otimes \boldsymbol{M}_h + \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{N}_h, \quad \mathbb{M}_\delta = \boldsymbol{M}_{\Delta t} \otimes \boldsymbol{M}_h$$

Matrix-oriented formulation of PCG:

$$\mathcal{A}(\boldsymbol{U}) = \boldsymbol{G} \quad \text{with} \quad \mathcal{A}(\boldsymbol{U}) = \boldsymbol{M}_h \boldsymbol{U} \boldsymbol{Q}_{\Delta t}^\top + \boldsymbol{N}_h^\top \boldsymbol{U} \boldsymbol{N}_{\Delta t}^\top + \boldsymbol{N}_h \boldsymbol{U} \boldsymbol{N}_{\Delta t} + \boldsymbol{Q}_h \boldsymbol{U} \boldsymbol{M}_{\Delta t}$$

An example

$$u_{tt} = c^2 \Delta u, \qquad u(T) = 0,\ u(1) = u_1 \quad \Omega = (0,1)^3, T = 1,$$

c wave speed, $u_0(r) = \mathbb{1}_{r<\sqrt{2}/5}$ (wo/lg, polar coordinates)

u is not continuous in $\bar{I} \times \bar{\Omega}$

Comparing:

- Space-time approach with PCG (unconditionally stable discr)
- Crank-Nicolson (with standard PCG at each timestep)

An example

$$u_{tt} = c^2 \Delta u, \qquad u(T) = 0,\ u(1) = u_1 \quad \Omega = (0,1)^3,\, T = 1,$$

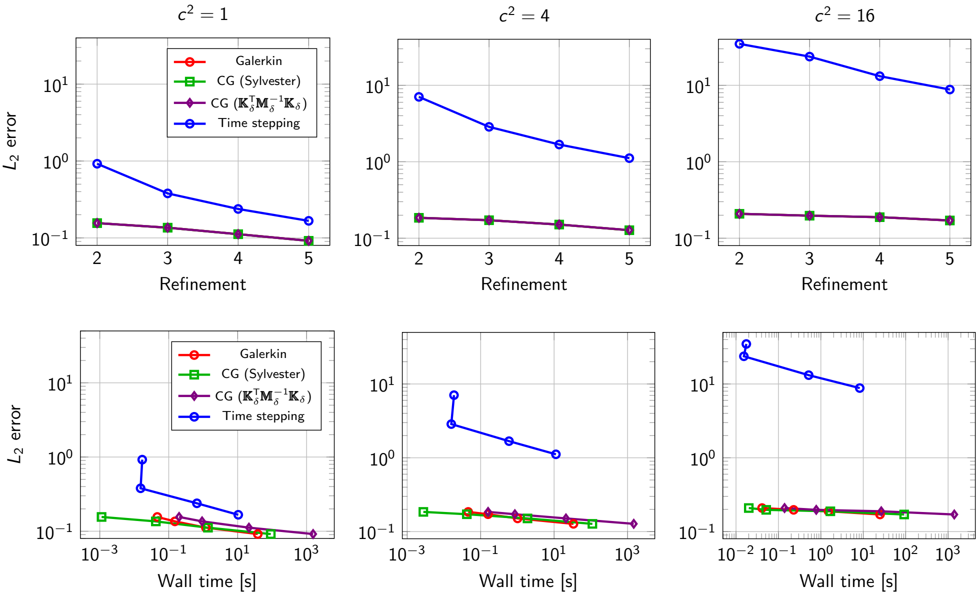
c wave speed, $u_0(r) = \mathbb{1}_{r<\sqrt{2}/5}$ (wo/lg, polar coordinates)

u is not continuous in $\bar{I} \times \bar{\Omega}$

Comparing:

- Space-time approach with PCG (unconditionally stable discr)
- Crank-Nicolson (with standard PCG at each timestep)

Numerical experiments



Conclusions and outlook

- Matrix-oriented formulations
  - make the use of demanding discretizations possible
  - provide new perspectives also for NLA
- Multivariable (tensor) versions under consideration

REFERENCES

- 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, v. 79 (1), 2020, pages 2067-2085.

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

- Julian Henning, Davide Palitta, V. S. and Karsten Urban.
A Very Weak Space-Time Variational Formulation for the Wave Equation: Analysis and Efficient Numerical Solution
ESAIM: M2AN, 56(4) 2022.

- V.S., *Computational methods for linear matrix equations*. SIAM Rev, 2016.

Conclusions and outlook

- Matrix-oriented formulations
  - make the use of demanding discretizations possible
  - provide new perspectives also for NLA
- Multivariable (tensor) versions under consideration

REFERENCES

- 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, v. 79 (1), 2020, pages 2067-2085.

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

- Julian Henning, Davide Palitta, V. S. and Karsten Urban.
A Very Weak Space-Time Variational Formulation for the Wave Equation: Analysis and Efficient Numerical Solution
ESAIM: M2AN, 56(4) 2022.

- V.S., *Computational methods for linear matrix equations*. SIAM Rev, 2016.