Matrix equations. Application to PDEs

V. Simoncini

Dipartimento di Matematica

Alma Mater Studiorum - Università di Bologna

valeria.simoncini@unibo.it

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

Reaction-diffusion PDEs

$$u_t = \ell(u) + f(u), \quad u = u(x, y, t), \qquad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0, T],$$

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

ℓ: diffusion operator linear in u $\qquad$ f: nonlinear reaction terms

An application: reaction-diffusion PDEs

$$u_t = \ell(u) + f(u), \quad u = u(x,y,t), \qquad (x,y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0,T],$$

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

ℓ: diffusion operator linear in u $\qquad$ f: nonlinear reaction terms

Generalization to systems:

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

An application: reaction-diffusion PDEs

$$u_t = \ell(u) + f(u), \quad u = u(x,y,t), \qquad (x,y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0,T],$$

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

ℓ: diffusion operator linear in u $\qquad$ f: nonlinear reaction terms

Generalization to systems:

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

Applications:

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

An application: reaction-diffusion PDEs

$$u_t = \ell(u) + f(u), \quad u = u(x, y, t), \qquad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0, T],$$

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

ℓ: diffusion operator linear in u $\qquad$ f: nonlinear reaction terms

Generalization to systems:

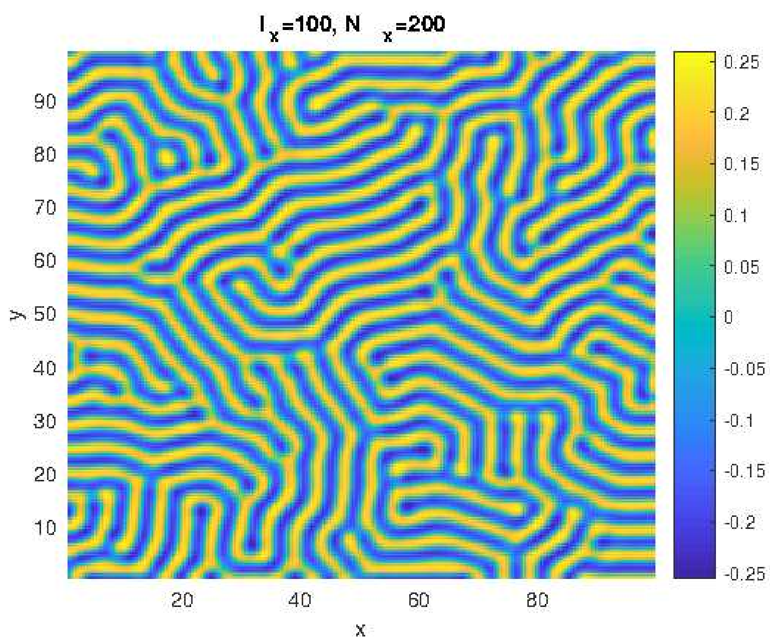
$$\begin{cases} u_t = \ell_1(u) + f_1(u, v), \\ v_t = \ell_2(v) + f_2(u, v), \quad \text{with} \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in]0, T] \end{cases}$$

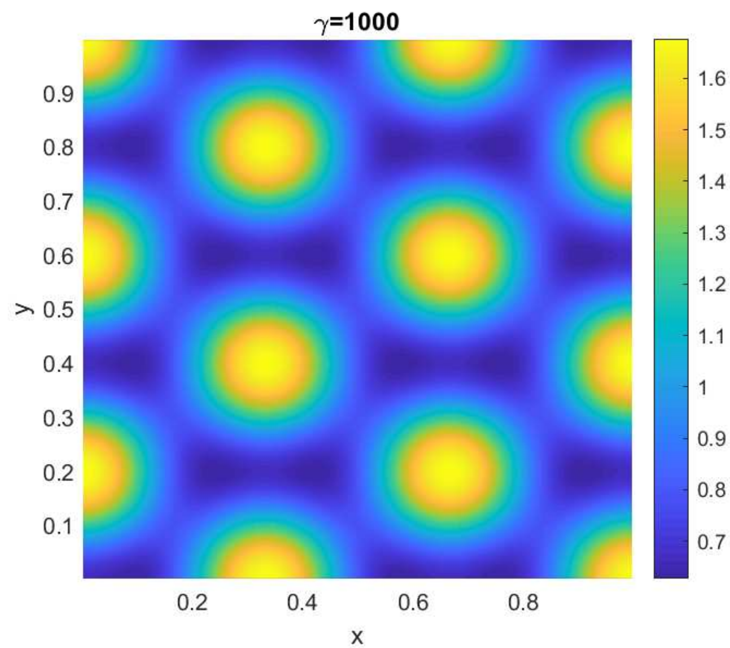
Application: mathematical description of morphogenesis by A.Turing

coupling between diffusion and nonlinear kinetics can lead to the so-called diffusion-driven or *Turing* instability

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

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: elliptic operator $\Rightarrow \ell(u) \approx A\mathbf{u}$, so that

$$\dot{\mathbf{u}} = A\mathbf{u} + f(\mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0$$

Analogously:

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

Space discretization of the reaction-diffusion PDE

ℓ: multiple of the Laplace operator $\Rightarrow \ell(u) \approx A\mathbf{u}$, so that

$$\dot{\mathbf{u}} = A\mathbf{u} + f(\mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0$$

Analogously:

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

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

$$A_{(i)} = I_y \otimes T_1 + T_2^T \otimes I_x \in \mathbb{R}^{N_x N_y \times N_x N_y}, \; i = 1, 2$$

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

Matrix-oriented formulation of reaction-diffusion PDEs

$$\dot{U} = T_1 U + U T_2 + F(U), \quad U(0) = U_0$$

$F(U)$ nonlinear vector function $f(\mathbf{u})$ evaluated componentwise

$\mathrm{vec}(U_0) = \mathbf{u}_0$ initial condition

Analogously,

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

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

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

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)

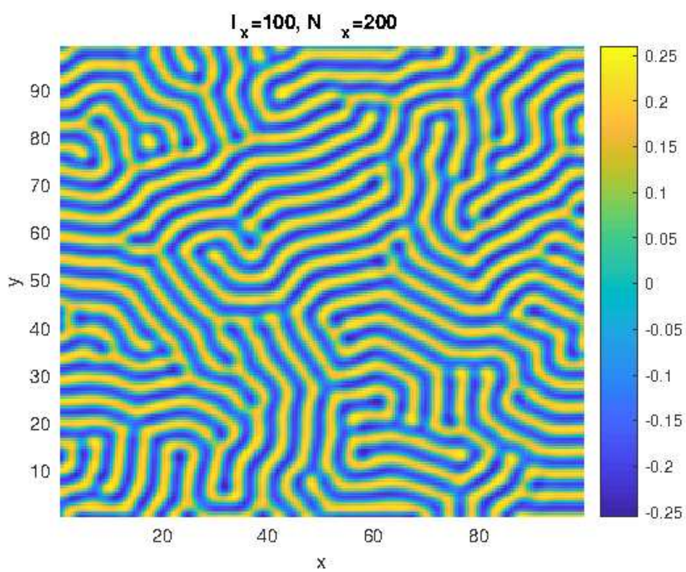
A numerical example of system of RD-PDEs

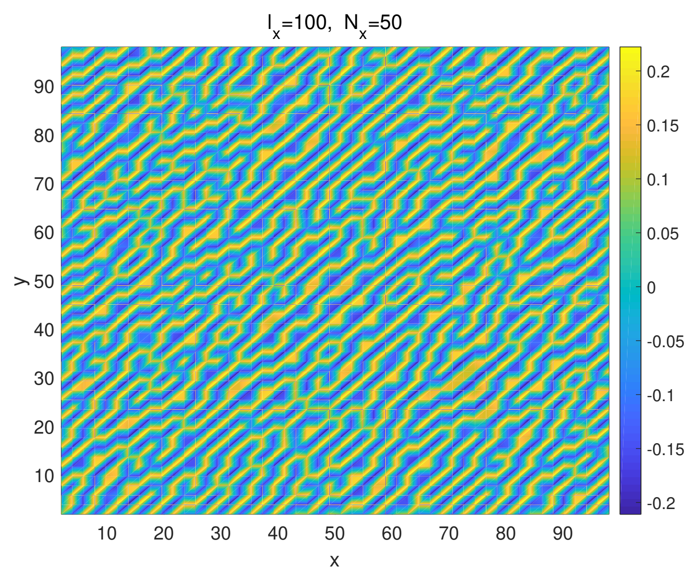
Model describing an electrodeposition process for metal growth

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

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

Turing pattern





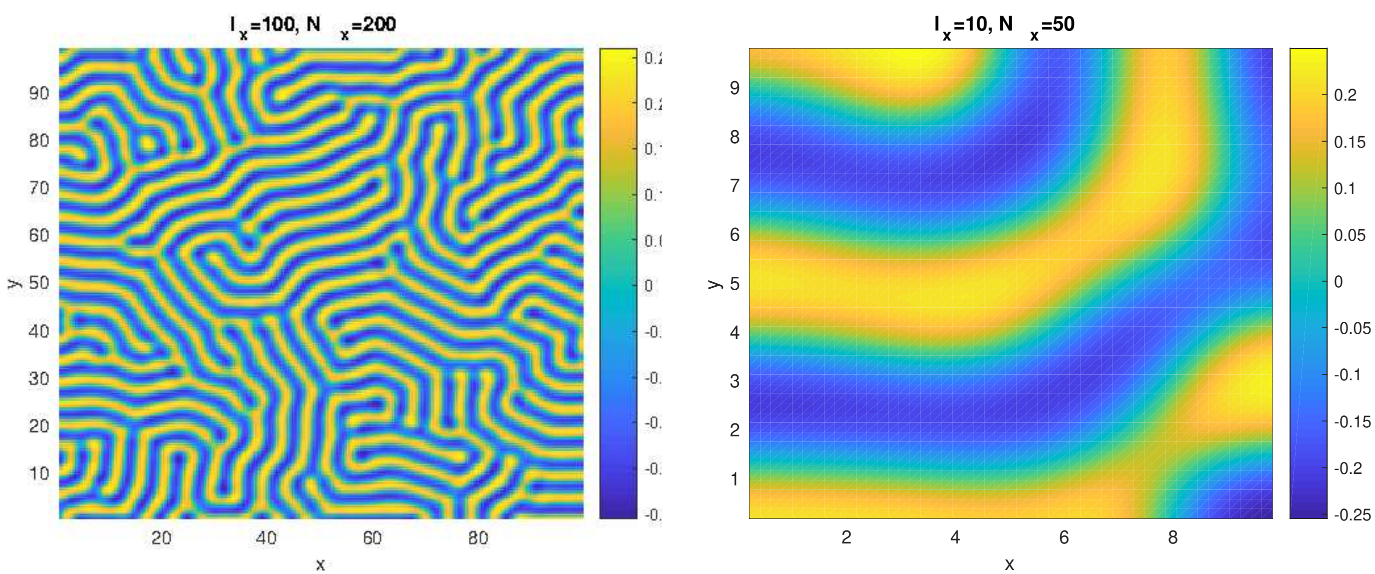
A numerical example of system of RD-PDEs

Model describing an electrodeposition process for metal growth

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

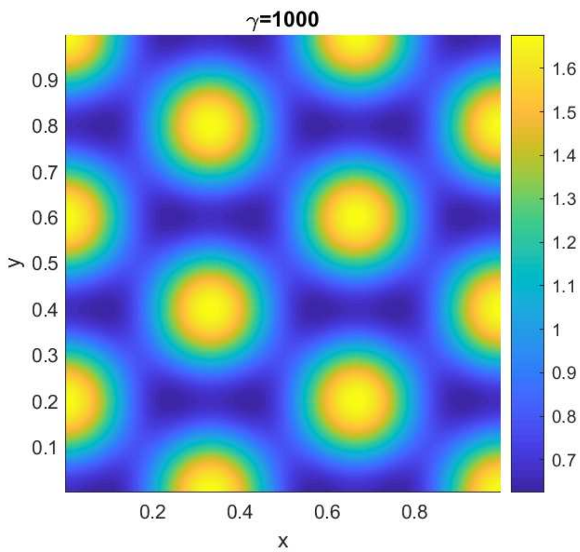
$$f_2(u,v) = \rho\left(C(1+k_2u)(1-v)[1-\gamma(1-v)] - Dv(1+k_3u)(1+\gamma v)\right)$$

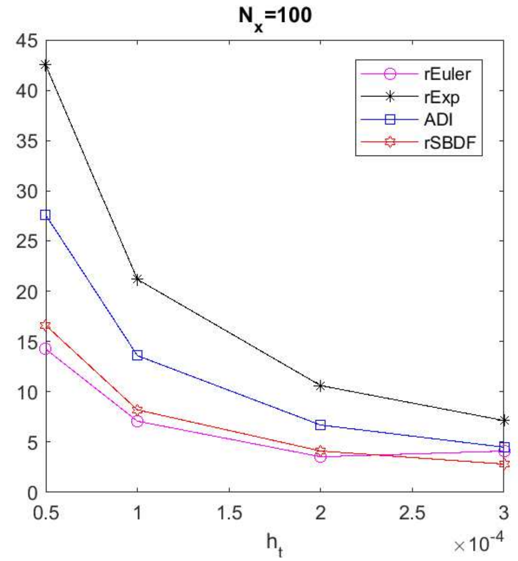
Turing pattern

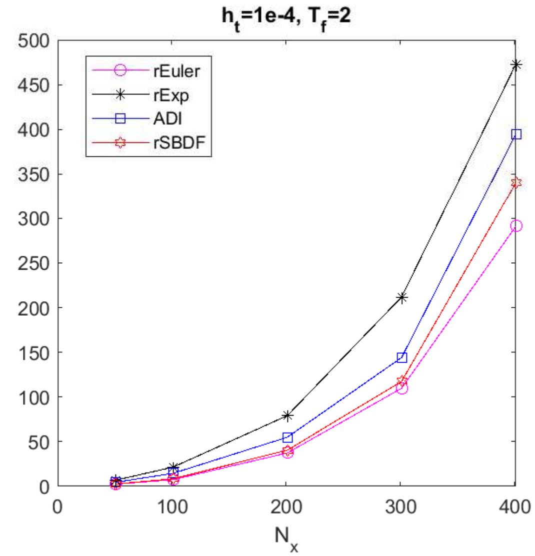


Schnackenberg model

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







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 $N_x = 50, 100, 200, 300, 400$

The three-dimensional case

$$\dot{\mathbf{u}} = A\mathbf{u} + f(\mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0 \text{ in } \qquad \Omega = [0, \ell_x] \times [0, \ell_y] \times [0, \ell_z]$$

High computational costs

Typically:

$$A = I_z \otimes I_y \otimes T_1 + I_z \otimes T_2^T \otimes I_x + T_3^T \otimes I_y \otimes I_x \in \mathbb{R}^{N_x N_y N_z \times N_x N_y N_z}$$

♣ Tensor versions of

- IMEX methods
- Exponential integrators

♣ DEIM-type projection

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 and maturity of linear system solvers

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 and maturity of linear system solvers

Outlook:

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

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 and maturity of linear system solvers

Outlook:

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

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

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