Lecture on reaction diffusion systems

Dario Madeo

Department of Information Engineering and Mathematics
Università degli studi di Siena
Widely used in many fields, such as chemistry, physics, biology, ecology, finance, etc.

Reaction-diffusion dynamics develops itself over time, but also over a spatial domain.

Generally, state variable $u(t, x)$ represents the concentration of a specific quantity at time instant $t$ and at the spatial point $x$.

The variation of the concentration $u$ is regulated by 2 phenomena:
- Concentration is transformed by local reactions.
- Concentration diffuses toward less occupied spatial region.
Mathematical model for reaction-diffusion systems

- Ordinary differential equations (ODE) are not enough to describe them, since there is also the spatial dimensions!
- Partial differential equations (PDE) are used!

\[
\frac{\partial u}{\partial t} - \sigma \Delta(u) = f(u)
\]

- \( \sigma > 0 \) is the diffusion coefficient.
- \( \Delta(u) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \) regulates the diffusion.
- \( f(u) \) is the reaction term.
Examples

**Figure:** Heat equation - Time evolution of a temperature profile

**Figure:** Subcritical Turing bifurcation: formation of a pattern from noisy initial conditions in the reaction-diffusion system of FitzHugh-Nagumo type.
Strong formulation

Find $u(t, x, y) \in C^2$ such that:

\[
\begin{cases}
\dot{u} - \sigma \Delta(u) = f(u) & \forall t, \forall (x, y) \in \Omega \\
\mathbf{n}' \nabla(u) = 0 & \forall (x, y) \in \partial\Omega \quad \text{Boundary condition} \\
u(0, x, y) = v(x, y) & \forall (x, y) \in \Omega \quad \text{Initial condition}
\end{cases}
\]

- Closed form solutions are rare!
- The space $C^2$ is too restrictive.
- We need to “relax” the problem.
Weak formulation

- Choose a test function \( w \in \mathcal{H}^1 \) (Sobolev space).
- Multiply both LHS and RHS of the PDE by \( w \).
- Integrate over all the spatial domain \( \Omega \).
- Use the divergence theorem with the boundary condition

**Find** \( u \in \mathcal{H}^1 \) **such that:**

\[
\begin{cases}
(\dot{u}, w) + \sigma a(u, w) = (f(u), w) \\
(u(0, x, y), w) = (v(x, y), w)
\end{cases}
\forall w \in \mathcal{H}^1
\]

where

- \((\alpha, \beta) = \int_{\Omega} \alpha \beta d\Omega\).
- \(a(\alpha, \beta) = \int_{\Omega} \nabla(\alpha)'\nabla(\beta) d\Omega\).
- \(\mathcal{H}^1 \{ \alpha : \Omega \to \mathbb{R} : \int_{\Omega} (\alpha^2 + \|\nabla(\alpha)\|^2) < +\infty \}\).
Galerkin approximation

- Weak formulation is still hard to solve.
- Reduce the size of the solution/test functions space.
- Choose $N$ linearly independent elements of $\mathcal{H}^1$.

$$\text{span}\{\phi_i\}_{i=1}^N = \mathcal{U} \subset \mathcal{H}^1$$

Find $u \in \mathcal{U}$ such that:

$$\begin{cases} 
(\dot{u}, \phi_i) + \sigma a(u, \phi_i) = (f(u), \phi_i) \\
(u(0, x, y), \phi_i) = (v(x, y), \phi_i)
\end{cases} \quad \forall \phi_i$$
Finite element method (FEM) is a numerical method based on the Galerkin approximation.

The spatial domain $\Omega$ is split into primitive shapes, called elements.
FEM: Basis functions

- $\phi_i(x, y)$ are piecewise polynomial
- $\phi_i(x_j, y_j) = \delta_{i,j}$, where $(x_j, y_j)$ is the $j$-th vertex.
- $\phi_i(x, y)$ is defined on all elements that share the $i$-th vertex.
- On the other elements, $\phi_i(x, y)$ is identically null.
- Solution is approximated as follows:

$$
\begin{align*}
    u(t, x, y) &= \sum_{i=1}^{N} \phi_i(x, y) U_i(t), \\
    \dot{u}(t, x, y) &= \sum_{i=1}^{N} \phi_i(x, y) \dot{U}_i(t)
\end{align*}
$$
FEM: Matricial formulations (1/2)

Closed form terms

- \((\dot{u}, \phi_i) = \sum_{j=1}^{N} (\phi_i, \phi_j) \dot{U}_i(t) = \sum_{j=1}^{N} m_{i,j} \dot{U}_i(t)\)

- \(a(u, \phi_i) = \sum_{j=1}^{N} a(\phi_i, \phi_j) U_i(t) = \sum_{j=1}^{N} k_{i,j} U_i(t)\)

- \((u(0, x, y), \phi_i) = \sum_{j=1}^{N} (\phi_i, \phi_j) U_i(0) = \sum_{j=1}^{N} m_{i,j} U_i(0)\)

Other terms

- \((f(u), \phi_i) = F_i(U(t))\)

- \((v(x, y), \phi_i) = V_i\)

These terms are usually evaluated by using numerical quadrature methods.
FEM: Matricial formulations (2/2)

Find $U(t) \in \mathbb{R}^N$ such that:

\[
\begin{cases}
M \dot{U}(t) + \sigma K U(t) = F(U(t)) \\
M U(0) = V
\end{cases} \\
\forall t > 0
\]

where

- $M$ is the mass matrix
- $K$ is the stiffness matrix

FEM approximates a PDE problem with a set of ODE
$M$ is also known as Gram matrix; if all $\phi_i$ are linearly independent, then $M$ is non singular.

- Suppose that there exist $c_j \in \mathbb{R}$, such that
  $$\exists k \neq i : c_k \neq 0 \land \phi_i = \sum_{j \neq i} c_j \phi_j \ \forall (x, y) \in \Omega.$$  
- Then, $\phi_i(x_i, y_i) = \sum_{j \neq i} c_j \phi_j(x_i, y_i) = 0$.
- But $\phi_i(x_i, y_i) = 1$ by definition!.
- $\Rightarrow \{\phi_i\}_{i=1}^N$ is a linearly independent set.

- $M$ is also a positive definite matrix.
- $K$ is a semidefinite positive matrix.
- Both $M$ and $K$ are sparse matrix.
Consider $MU(0) = V \Rightarrow U(0) = M^{-1}V$

The previous result is also the solution of the following minimization problem:

$$\arg \min_{U(0)} \int_\Omega (u(0, x, y) - v(x, y))^2 \, d\Omega$$

with $u(0, x, y) \in U$.

$$\frac{\partial}{\partial U_j} \int_\Omega \left( \sum_{i=1}^{N} \phi_i U_i(0) - v \right)^2 \, d\Omega = 0$$

$$2 \int_\Omega \phi_j \left( \sum_{i=1}^{N} \phi_i U_i(0) - v \right) \, d\Omega = 0$$

$$\sum_{i=1}^{N} (\phi_i, \phi_j) U_i(0) = (\phi_j, v)$$
Example:

- Approximation of the function
  \[ f(x) = \frac{1}{50}(x - 0.8)(x - 2.4)(x - 5)^2 \]
- \( \Omega = [0, 3] \subset \mathbb{R}, N = 4 \)

**Weierstrass theorem guarantee...**
Polynomial reaction terms are used in many models:

- Chemical kinetics
- Malthusian growth \( f(u) = ru \)
- Verhulst model \( f(u) = ru \left(1 - \frac{u}{k}\right)\)
- ...

In general:

\[
f(u) = \sum_{p=1}^{N_p} \theta_p u^p
\]
Parameter estimation

Suppose to have a reaction-diffusion model with polynomial reaction term \( f(u) = \sum_{p=1}^{N_p} \theta_p u^p \).

At certain time instants \( t_i \), we measure the whole state of the system in all the mesh points \( (Y(t_i) = U(t_i)) \).

We want to estimate the system parameters \( (\sigma, \theta_1, \ldots, \theta_{N_p}) \) using the FEM approximation of the infinite dimensional system.

\[
\begin{align*}
M \dot{U}(t) + \sigma K U(t) &= F(U(t), \Theta) \quad \forall t > 0 \\
MU(0) &= V
\end{align*}
\]
Standard approach

- We simulate the system on fine grain mesh using Comsol(R). In this way we obtain “real world” data $Y(t_i)$.
- We use the Levenberg - Marquardt algorithm implemented in the Optimization Toolbox for Matlab(R) to solve the following nonlinear least square problem:

$$\min_{\sigma, \Theta} \sum_{i} \| Y(t_i) - U(t_i, \sigma, \Theta) \|^2$$

- $U(t_i)$ are evaluated by solving the approximated system (FEM) on a coarser mesh.
Remarks on the standard approach

- $U(t)$ must be evaluated on-line when the algorithm changes its guess about parameters.
- The Jacobian $J = \frac{\partial U(t_i, \sigma, \Theta)}{\partial (\sigma, \Theta)}$ used by the minimization algorithm is numerically approximated on-line through finite differences.
- We could provide to the algorithm the exact way to calculate $J$, but it is not known.
- Numerical quadrature methods are heavily used to evaluate $U(t)$ and $J$.
- Definitely, we are not using any information on the underlying system to perform the estimation (black-box approach).
A different approach for parameter estimation

In the next slides, a different approach for parameter estimation is shown. This method is based on the work:

Our approach (1/3)

- The term \((u^p, \phi_i)\) of the reaction term can be evaluated without numerical quadrature.
  - \(p = 1 \Rightarrow F(U(t)) = D_1(U(t)) = MU(t)\)
  - \(p = 2 \Rightarrow F(U(t)) = [G_1 U(t) \cdots G_N U(t)]U(t) = D_2(U(t))U(t)\), where \(G_i = \{(\phi_i, \phi_j \phi_k)\}_{j, k=1}^N\)
  - ...
  - \(p = P \Rightarrow F(U(t)) = D_P(U(t))U(t)\)
- The approximated problem can be rewritten as follows:

\[
\begin{cases} 
M\dot{U}(t) + \sigma KU(t) = \sum_{p=1}^{N_P} \theta_p D_p(U(t))U(t) \\
MU(0) = V
\end{cases}
\quad \forall t > 0
\]
Our approach (2/3)

We can approximate time derivatives with Euler formula:

\[
\begin{align*}
Y(t_{i+1}) &= \frac{U(t_{i+1}) - U(t_i)}{\tau_i} \\
&= M^{-1} \left( -\sigma K + \sum_{p=1}^{N_p} \theta_p D_p(U(t_i)) \right) U(t_i) \\
U(0) &= M^{-1} V
\end{align*}
\]

where \( \tau_i = t_{i+1} - t_i \), and \( t_0 = 0 \). Then, following linear relation between measurements and parameters holds:

\[ Y = W\Lambda \]

where \( \Lambda = (\sigma, \Theta) \), and \( W \) depends on the system equation.
Our approach (3/3)

\[ Y = W \Lambda \text{ is suitable for a least square approach!} \]

\[ \hat{\Lambda} = W^\# Y \]

where \( W^\# = (W'W)^{-1}W' \) is the pseudo-inverse of \( W \).

- We performed both standard and new approaches on meshes of different sizes.
- In the next slides, we will show the comparison of the performances of both methods.
The following reaction terms have been used:

- **Logistic equation:**
  \[
  f(u) = ru \left(1 - \frac{u}{k}\right) \ (\sigma = 0.01, \ r = 0.1, \ k = 10).
  \]

- **Aleee equation:**
  \[
  f(u) = \alpha u + \beta u^2 - \gamma u^3 \ (\sigma = 0.001, \ \alpha = 0.01, \ \beta = 0.01, \ \gamma = 0.005).
  \]
## Results (2/7)

<table>
<thead>
<tr>
<th>Par.</th>
<th>Nominal Value</th>
<th>Est. Value $N = 81$</th>
<th>Est. Value $N = 121$</th>
<th>Est. Value $N = 256$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>1e-2</td>
<td>1.055e-2</td>
<td>0.998e-2</td>
<td>0.967e-2</td>
</tr>
<tr>
<td>$r$</td>
<td>1e-1</td>
<td>0.983e-1</td>
<td>0.989e-1</td>
<td>0.995e-1</td>
</tr>
<tr>
<td>$k$</td>
<td>10</td>
<td>9.999</td>
<td>9.999</td>
<td>9.998</td>
</tr>
<tr>
<td>$MSE$</td>
<td>-</td>
<td>2.65e-5</td>
<td>9.97e-6</td>
<td>6.4e-7</td>
</tr>
</tbody>
</table>

**Table:** Estimated parameter values for the logistic equation with different number of nodal points $N$. Linear least squares identification.
## Results (3/7)

<table>
<thead>
<tr>
<th>Par.</th>
<th>Nominal Value</th>
<th>Est. Value $N = 81$</th>
<th>Est. Value $N = 121$</th>
<th>Est. Value $N = 256$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>1e-2</td>
<td>1.56e-2</td>
<td>1.41e-2</td>
<td>1.22e-2</td>
</tr>
<tr>
<td>$r$</td>
<td>1e-1</td>
<td>1.037e-1</td>
<td>1.019e-1</td>
<td>1.005e-1</td>
</tr>
<tr>
<td>$k$</td>
<td>10</td>
<td>9.788</td>
<td>9.899</td>
<td>9.962</td>
</tr>
<tr>
<td>MSE</td>
<td>-</td>
<td>8.67e-3</td>
<td>5.08e-3</td>
<td>2.19e-3</td>
</tr>
</tbody>
</table>

**Table:** Estimated parameter values for the logistic equation with different number of nodal points $N$. Nonlinear least squares identification.
Results (4/7)

Figure: Time evolution of the spatial mean square error between the simulation of the logistic equation with nominal parameters and estimated ones in the finer (top) and coarser (bottom) meshes. Solid line: LS algorithm; dashed line: NLS algorithm.
## Results (5/7)

<table>
<thead>
<tr>
<th>Par.</th>
<th>Nominal Value</th>
<th>Est. Value $N = 81$</th>
<th>Est. Value $N = 121$</th>
<th>Est. Value $N = 256$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>1e-3</td>
<td>0.869e-3</td>
<td>0.936e-3</td>
<td>0.952e-3</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1e-2</td>
<td>1.629e-2</td>
<td>1.623e-2</td>
<td>1.609e-2</td>
</tr>
<tr>
<td>$\beta$</td>
<td>1e-2</td>
<td>0.644e-2</td>
<td>0.647e-2</td>
<td>0.654e-2</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>5e-3</td>
<td>4.511e-3</td>
<td>4.514e-3</td>
<td>4.521e-3</td>
</tr>
<tr>
<td><strong>MSE</strong></td>
<td>-</td>
<td>2.14e-4</td>
<td>1.12e-4</td>
<td>0.91e-4</td>
</tr>
</tbody>
</table>

**Table:** Estimated parameter values for the Allee equation with different number of nodal points $N$. Linear least squares identification.
## Results (6/7)

<table>
<thead>
<tr>
<th>Par.</th>
<th>Nominal Value</th>
<th>Est. Value ( N = 81 )</th>
<th>Est. Value ( N = 121 )</th>
<th>Est. Value ( N = 256 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma )</td>
<td>1e-3</td>
<td>3.312e-3</td>
<td>2.256e-3</td>
<td>1.513e-3</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>1e-2</td>
<td>0.905e-2</td>
<td>2.313e-2</td>
<td>2.051e-2</td>
</tr>
<tr>
<td>( \beta )</td>
<td>1e-2</td>
<td>1.129e-2</td>
<td>3.818e-2</td>
<td>4.741e-2</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>5e-3</td>
<td>5.248e-3</td>
<td>4.352e-3</td>
<td>4.416e-3</td>
</tr>
<tr>
<td>( MSE )</td>
<td>-</td>
<td>1.09e-1</td>
<td>0.48e-1</td>
<td>0.99e-2</td>
</tr>
</tbody>
</table>

**Table:** Estimated parameter values for the Allee equation with different number of nodal points \( N \). Nonlinear least squares identification.
Results (7/7)

Figure: Time evolution of the spatial mean square error between the simulation of the Allee model with nominal parameters and estimated ones in the finer (top) and coarser (bottom) meshes. Solid line: linear LS algorithm; dashed line: nonlinear LS algorithm.