DDEM: A Modeling Package for Delay Differential Equations

Hossein ZivariPiran
hzp@cs.toronto.edu

Department of Computer Science
University of Toronto

SciCADE09
(Beijing, May 2009)
The Modeling Process

Modeling with Delay Differential Equations

Why Modeling Packages?

Difficulties of Designing a Modeling Package

DDEM

♦ Software Design
♦ User Interface
♦ Numerical Experiments
The Modeling Process

- Physical/Biological Phenomenon
- Physical Laws / Empirical Rules
- Mathematical Model
- Sensitivity Analysis
- Observed Data
- Refined Mathematical Model
- Parameter Estimation
- Practical Mathematical Model
The Modeling Process - Parameterized Models

<table>
<thead>
<tr>
<th></th>
<th>IVP ODE/DAE/DDE</th>
<th>BVP ODE</th>
<th>BVP PDE</th>
<th>IVP PDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y(t) )</td>
<td>( y(t; p) )</td>
<td>( y(x) )</td>
<td>( y(x; p) )</td>
<td>( u(x, t) )</td>
</tr>
<tr>
<td>( u(x) )</td>
<td>( u(x; p) )</td>
<td>( u(x, t) )</td>
<td>( u(x, t; p) )</td>
<td></td>
</tr>
</tbody>
</table>

Parameters are used to:

- Make the model applicable in similar situations.
- Analyze the effect of uncertainties.
- Represent unknown quantities.
Computing a numerical approximation for fixed values of parameters and some initial/boundary conditions.

- Traditionally considered as the core of numerical studies.
- Many researchers are working on developing faster/more reliable methods.
- **Generalized Methods** - vs - **Specialized Methods**
Forward Sensitivity Analysis

- The (first order) solution sensitivity with respect to the model parameter $p_i$ is defined as the vector

$$s_i(t; \mathbf{p}) = \left\{ \frac{\partial}{\partial p_i} \right\} y(t; \mathbf{p}), \quad (i = 1, \ldots, \mathcal{L})$$

- The second order solution sensitivity with respect to the model parameters $p_i$ and $p_j$ is defined as the vector

$$r_{ij}(t; \mathbf{p}) = \left\{ \frac{\partial}{\partial p_j} \right\} s_i(t; \mathbf{p}) = \left\{ \frac{\partial^2}{\partial p_j \partial p_i} \right\} y(t; \mathbf{p}), \quad (i, j = 1, \ldots, \mathcal{L})$$
Parameter Estimation Problem

- A Parameterized Model
  \[ y(t; p) \]

- A Set of Data (Observations/Measurements)
  \[ \{ Y(\gamma_i) \approx y(\gamma_i; p^*) \} \]

- Estimate \( p^* \) by minimizing an objective function.
  e.g.
  \[ W(p) = \sum_i [Y(\gamma_i) - y(\gamma_i; p)]^2. \]
Parameter Estimation/Fitting is not easy, like almost any other optimization problem,

- The parameters may not be identifiable ⇒ Numerical Difficulties
- The efficiency may depend strongly on the place/number of data points ⇒ Strategies for Collecting the Best Data
An Initial Value Problem (IVP) for Ordinary Differential Equations (ODEs)

\[ y'(t) = f(t, y(t)), \]
\[ y(t_0) = y_0, \]

Retarded Delay Differential Equations (RDDEs)

\[ y'(t) = f(t, y(t), y(t - \sigma_1), \ldots, y(t - \sigma_\nu)), \quad \text{for} \ t_0 \leq t \leq t_F \]
\[ y(t) = \phi(t), \quad \text{for} \ t \leq t_0 \]

\[ \sigma_i = \sigma_i(t, y(t)) \geq 0 \text{ delay (constant / time dependent / state dependent)} \]
\[ \phi(t) \text{ history function (constant / time dependent)} \]

Neutral Delay Differential Equations (NDDEs)

\[ y'(t) = f(t, y(t), y(t - \sigma_1), \ldots, y(t - \sigma_\nu)), \]
\[ y'(t - \sigma_{\nu+1}), \ldots, y'(t - \sigma_{\nu+\omega})), \quad \text{for} \ t_0 \leq t \leq t_F \]
\[ y(t) = \phi(t), \quad y'(t) = \phi'(t), \quad \text{for} \ t \leq t_0. \]
A Parameterized DDE

\[ y'(t; \mathbf{p}) = f(t, y(t; \mathbf{p}), y(t - \sigma(t; \mathbf{p})); \mathbf{p}) \quad \text{for} \quad t_0(\mathbf{p}) \leq t \]
\[ y(t; \mathbf{p}) = \phi(t; \mathbf{p}), \quad \text{for} \quad t \leq t_0(\mathbf{p}) \]

For Example, in the neutral delay logistic Gause-type predator-prey system [Kuang 1991]

\[ y'_1(t) = y_1(t)(1 - y_1(t - \tau) - \rho y'_1(t - \tau)) - \frac{y_2(t)y_1(t)^2}{y_1(t)^2 + 1}, \]
\[ y'_2(t) = y_2(t) \left( \frac{y_1(t)^2}{y_1(t)^2 + 1} - \alpha \right), \]

where \(\alpha = 1/10\), \(\rho = 29/10\) and \(\tau = 21/50\), for \(t\) in \([0, 30]\). The history functions are

\[ \phi_1(t) = \frac{33}{100} - \frac{1}{10}t, \]
\[ \phi_2(t) = \frac{111}{50} + \frac{1}{10}t, \]

for \(t \leq 0\).

Parameters are

\[ \mathbf{p} = [\tau, \rho, \alpha, a, b, c, d]. \]
Classical Theory of Step by Step Integration for IVPs.
- Runge-Kutta (RK).
- Linear Multistep (LM).

Continuous Solution using Polynomial Approximation.
- Continuous Runge-Kutta (CRK).
- Linear Multistep methods have natural approximating polynomials.

DDEs: Combining an “interpolation” method (for evaluating delayed solution values) with an ODE integration method (for solving the resulting “ODE”).
Derivative Discontinuities

In general

$$\phi'(t_0) \neq f(t_0, \phi(t_0), \phi(t_0 - \sigma_1), \cdots, \phi(t_0 - \sigma_N))$$

Due to the existence of delays, discontinuities propagate along the integration interval.

Solution is smoothed for RDDEs but in general not for NDDEs.

The RK and LM methods fail in presence of discontinuities.

Treatment: Tracking Discontinuities and forcing them to be mesh points.
The predator-prey model
DDEs - Sensitivity Analysis

- **Finite Difference Approach**

\[
\{ \frac{\partial}{\partial p_i} \} y(t; \mathbf{p}) \approx \frac{y(t; \mathbf{p} + e_i \Delta p_i) - y(t; \mathbf{p})}{\Delta p_i}.
\]

Due to the rounding errors, the approximation is only \( O(\sqrt{\text{Tol}}) \) with the best choice for \( \Delta p_i \).

- **Internal Differentiation Approach for IVPs**

\[
y'(t; \mathbf{p}) = f(t, y(t; \mathbf{p}); \mathbf{p}), \quad y(t_0) = y_0(\mathbf{p})
\]

\[
\downarrow
\]

Differentiation + Chain Rule + Clairaut’s Theorem

\[
\downarrow
\]

\[
s'_i = \frac{\partial f}{\partial y} s_i + \frac{\partial f}{\partial p_i}, \quad s_i(t_0) = \frac{\partial y_0(\mathbf{p})}{\partial p_i}, \quad (i = 1, \ldots, \mathcal{L})
\]
Adapted Internal Differentiation Approach for DDEs

\[ y'(t; p) = f(t, y(t; p), y(\alpha(t, y; p); p), y'(\alpha(t, y; p); p); p) \]
\[ y(t; p) = \phi(t; p), \text{ for } t \leq t_0(p) \]

\[ \downarrow \]

Differentiation + Chain Rule + Clairaut’s Theorem

\[ \downarrow \]

\[ s'_i(t) = \frac{\partial f}{\partial y} s_i(t) + \frac{\partial f}{\partial y(\alpha_k)} \left( y'(\alpha_k) \left( \frac{\partial \alpha}{\partial y} s_i(t) + \frac{\partial \alpha}{\partial p} \right) + s_i(\alpha) \right) \]
\[ + \frac{\partial f}{\partial y'(\alpha)} \left( y''(\alpha) \left( \frac{\partial \alpha}{\partial y} s_i(t) + \frac{\partial \alpha}{\partial p} \right) + s'_i(\alpha) \right) \]
\[ + \frac{\partial f}{\partial p} \]
The predator-prey model
Using the Theory of Hybrid ODE systems [Tolsma & Barton]

- Sensitivity Update Equation,

\[ \frac{\partial y}{\partial p_l}(\lambda_{r+1}^+) = \frac{\partial y}{\partial p_l}(\lambda_{r+1}^-) + \left(y'(\lambda_{r+1}^-) - y'(\lambda_{r+1}^+)\right) \frac{d\lambda_{r+1}(p)}{dp_l} \]

- where

\[
\begin{align*}
\frac{d\lambda_{r+1}(p)}{dp_l} &= \frac{\partial t_0(p)}{\partial p_l} - \frac{\partial}{\partial y} \frac{\partial y}{\partial p_l} + \frac{\partial}{\partial p_l} - \frac{\partial}{\partial t} \frac{d\lambda_r(p)}{dp_l} \\
\frac{d\lambda_0(p)}{dp_l} &= \frac{\partial t_0(p)}{\partial p_l}
\end{align*}
\]
DDEs - Parameter Estimation

- **Algorithms for Nonlinear Least-Squares**
  
  - Unconstrained
    
    $$
    \min_{p} W(p) = \sum_{i} [Y(\gamma_i) - y(\gamma_i; p)]^2.
    $$
    
    ↓
    
    Levenberg-Marquardt
    
    Variations of Sequential Quadratic Programming (SQP)

  - Constrained
    
    $$
    \min_{p} W(p) = \sum_{i} [Y(\gamma_i) - y(\gamma_i; p)]^2,
    $$
    
    $$
    c_j(p) = 0, \quad j \in \mathcal{E},
    $$
    
    $$
    c_j(p) \geq 0, \quad j \in \mathcal{I}.
    $$
    
    ↓
    
    Sequential Quadratic Programming (SQP)

- **The smoothness** of functions involved in the problem, the objective function and constraints, is a *necessary* assumption.
Computing the Gradient/Hessian

\[
\left( \frac{\partial W(p)}{\partial p_l} \right)_\pm = -2 \sum_i [Y(\gamma_i) - y(\gamma_i; p)] \left( \frac{\partial y(\gamma_i; p)}{\partial p_l} \right)_\pm
\]

\[
\left( \frac{\partial^2 W(p)}{\partial p_l \partial p_m} \right)_\pm = 2 \sum_i \left[ \left( \frac{\partial y(\gamma_i; p)}{\partial p_l} \right)_\pm \left( \frac{\partial y(\gamma_i; p)}{\partial p_m} \right)_\pm - [Y(\gamma_i) - y(\gamma_i; p)] \left( \frac{\partial^2 y(\gamma_i; p)}{\partial p_l \partial p_m} \right)_\pm \right]
\]

Recall the jump equation for sensitivities

\[
\frac{\partial y}{\partial p_l} (\lambda_{r+1}^+) = \frac{\partial y}{\partial p_l} (\lambda_{r+1}^-) + \left( y'(\lambda_{r+1}^-) - y'(\lambda_{r+1}^+) \right) \frac{d\lambda_{r+1}(p)}{dp_l}
\]

\[\downarrow\]

The General Rule: A jump occurs in \( W(p) \) when

a discontinuity point \( \lambda_{r+1}(p) \) passes a data point \( \gamma_i \).
Avoiding The Non-smoothness

Force the ordering by adding \( \lambda_r(p) \leq \gamma_i < \lambda_{r+1}(p) \) to the set of constraints.

The partial derivatives (gradient) of the new constraints, \( \frac{\partial \lambda_{r+1}(p)}{\partial p} \), can be computed recursively.
Why Modeling Packages?

- Sensitivity analyzer and parameter estimator have become as essential as simulator for studying a phenomenon using a mathematical model.

- Increase of the processing power ⇒ Use of more complex/detailed models ⇒ sensitivity analysis and parameter estimation become more complicated.

- Sensitivity analysis and parameter estimation are highly dependent on the simulator ⇒ An integrated design could lead to efficient communications and prevent possible redundancies.
Difficulties of Designing a Modeling Package

- We need to develop an efficient simulator, a sensitivity analyzer and a parameter estimator ⇒ Time Consuming/Expensive

- How to deal with the "Generality - vs - Efficiency" problem?

- Want to have a manageable design ⇒ with little effort be able to change some algorithms or add new algorithms.

- Want to have a user-friendly interface ⇒ Easy enough for nonspecialists and controllable enough for specialist.

- Ideally Parallelizable ⇒ At least thread safe ⇒ No Global Variables.

- Easily incorporable into other packages/programms ⇒ Easy to do experiments, for instance, comparing different algorithms ⇒ No Assumption for Global Modules.
Software Design

User Calls

- User
  - A
  - B
  - C

- Simulator
  - B
  - C

- Sensitivity Analyzer
  - C

- Parameter Estimator
  - C

DDEM – p.18/21
Sensitivity Analyzer

Variational Integrator

Jump Handler

Simulator::
Master Integrator
Parameter Estimator

- Master Control
  - Optimizer (SQP)
    - e04unc/nag_opt_nlin_lsq from NAG
  - Subfunctions Calculator
  - Constraints Calculator
  - Solution/Gradient Provider
    - Simulator:: Master Integrator
    - Sensitivity Analyzer:: Variational Integrator
Simulation

```c
problem1->create(nVariables, nParameters, nEventFuncs, nHistorySegments);
problem1->setF(f);
problem1->setHistorySegments(history);
problem1->setDelayArguments(nu, omega, delays, stateDependent);
problem1->setY0(initial Value);

myIVP2DDEImprovedCRK1 = new IVP2DDEImprovedCRK(0/*interp_Flag*/);
mySimulator1 = new ddemSimulator(myIVP2DDEImprovedCRK1);

simulator1->simulate(problem1,
    end time, parameters,
    relTolerance, absTolerance,
    simulationSolution1, communication pointer);
```
Sensitivity Analysis

```c
needSensitivity[0] = TRUE;
needSensitivity[1] = FALSE;
.
.
sensitivityAnalyzer1->computeSensitivities(
simulator1,
problem1,
end time, parameters,
relTolerance, absTolerance,
needSensitivity,
sensitivitySolution1,
communication pointer);
```
Parameter Estimation

Constraints:
- Linear Constraints (Coefficients Matrix)
- Nonlinear Constraints (Function)
- Simple Bounds

```cpp
data1->Load("example01.d");

constraints1->setSimpleLowerBound(0, .05);

parameterEstimator1->EstimateParameters(simulator1, problem1, data1, constraints1, optimumParameters, relTolerance, absTolerance, statistics, communication pointer);
```
Sensitivity of the solution with respect to starting jump for the artificial problem,

\[ y' = y(y(t)), \]

for \( t \) in \([2, 5.5]\). The history function is

\[ y = 0.5, \quad \text{for } t < 2 \]

and

\[ y(2) = 1. \]

The \( C^0 \) discontinuity of the solution at \( \xi_0 = 2 \) introduces break points at \( \xi_1 = 4 \left( C^1 \right) \) and \( \xi_2 = 4 + 2 \ln 2 \approx 5.386 \left( C^2 \right) \).

The exact solution to this problem is

\[
y(t) = \begin{cases} 
t/2, & \text{for } \xi_0 \leq t \leq \xi_1 \\
2 \exp(t/2 - 2), & \text{for } \xi_1 \leq t \leq \xi_2 \\
4 - 2 \ln(1 + \xi_2 - t) & \text{for } \xi_2 \leq t \leq 5.5
\end{cases}
\]

Parameters are

\[ p = [y(2)]. \]
**Artificial Problem**

- **Graph 1:**
  - **X-axis:** Time (t)
  - **Y-axis:** y
  - Data points: (2, 1), (2.5, 2), (3, 3), (3.5, 4), (4, 5), (4.5, 6), (5, 7), (5.5, 8)

- **Graph 2:**
  - **X-axis:** Time (t)
  - **Y-axis:** y'
  - Data points: (2, 0.4), (2.5, 0.8), (3, 1.2), (3.5, 1.6), (4, 2), (4.5, 2.4), (5, 2.8), (5.5, 3.2)

- **Graph 3:**
  - **X-axis:** Time (t)
  - **Y-axis:** y(p + Δp)
  - Data points: (2, 1.5), (2.5, 2), (3, 2.5), (3.5, 3), (4, 3.5), (4.5, 4), (5, 4.5), (5.5, 5)

- **Graph 4:**
  - **X-axis:** Time (t)
  - **Y-axis:** \( \frac{\partial y}{\partial y(3)} \)
  - Data points: (2, 1), (2.5, 1.5), (3, 2), (3.5, 2.5), (4, 3), (4.5, 3.5), (5, 4), (5.5, 4.5)
Artificial Problem - Finite Differences

- $\Delta p = 10^{-3}$
- $\Delta p = 10^{-5}$
- $\Delta p = 10^{-7}$
- $\Delta p = 10^{-9}$
The predator-prey model

\[ y_1'(t) = y_1(t)(1 - y_1(t - \tau) - \rho y_1'(t - \tau)) - \frac{y_2(t)y_1(t)^2}{y_1(t)^2 + 1} \]

\[ y_2'(t) = y_2(t) \left( \frac{y_1(t)^2}{y_1(t)^2 + 1} - \alpha \right) \]

where \( \alpha = 1/10 \), \( \rho = 29/10 \) and \( \tau = 21/50 \), for \( t \) in \([0, 30] \). The history functions are

\[ \phi_1(t) = \frac{33}{100} - \frac{1}{10} t \]

\[ \phi_2(t) = \frac{111}{50} + \frac{1}{10} t \]

for \( t \leq 0 \).

Parameters are

\[ p = [\tau, \rho, \alpha, a, b, c, d] \]

where

\[ \phi_1(t) = a + b t \]

\[ \phi_2(t) = c + d t \]
The predator-prey model (structure-related parameters, $y_1$)
The predator-prey model (structure-related parameters, $y_2$)

\begin{align*}
\frac{\partial y_2}{\partial t} & \quad \text{(top left)} \\
\frac{\partial y_2}{\partial \tau} & \quad \text{(top right)} \\
\frac{\partial y_2}{\partial \rho} & \quad \text{(bottom left)} \\
\frac{\partial y_2}{\partial \alpha} & \quad \text{(bottom right)}
\end{align*}
The predator-prey model (history-related parameters, $y_1$)
The predator-prey model (history-related parameters, $y_2$)
The predator-prey model ($\frac{\partial y_1}{\partial \tau}$) - Finite Differences
Numerical Experiments - Parameter Estimation

- Estimating $y(2)$ for the artificial problem,

$$y' = y(y(t)),$$

for $t$ in $[2, 5.5]$. The history function is

$$y = 0.5, \quad \text{for } t < 2$$

and

$$y(2) = 1.$$

- Start with up to 10% random perturbation in original $y(2)$.
  For $\gamma$'s we choose 10 random points, one of which is a discontinuity point.

- Run the parameter estimator 10 times.

- Results

<table>
<thead>
<tr>
<th>Estimator Choice</th>
<th>FCN</th>
<th>Iterations</th>
<th>Time</th>
<th>OBJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Simple</td>
<td>3325</td>
<td>40.5</td>
<td>0.775</td>
<td>$1.58101e - 322$</td>
</tr>
<tr>
<td>SensJac</td>
<td>142</td>
<td>1.8</td>
<td>0.060</td>
<td>$6.24537e - 16$</td>
</tr>
<tr>
<td>SensJac AddedCons</td>
<td>79</td>
<td>1</td>
<td>0.033</td>
<td>$6.24537e - 16$</td>
</tr>
</tbody>
</table>
Numerical Experiments - Parameter Estimation

- Estimating structure-related parameters $\tau$, $\rho$, $\alpha$ for the predator-prey model

\[
\frac{\text{d}y_1}{\text{d}t}(t) = y_1(t)(1 - y_1(t - \tau) - \rho y_1'(t - \tau)) - \frac{y_2(t)y_1(t)^2}{y_1(t)^2 + 1}
\]

\[
\frac{\text{d}y_2}{\text{d}t}(t) = y_2(t)\left(\frac{y_1(t)^2}{y_1(t)^2 + 1} - \alpha\right)
\]

- Start with up to 10% random perturbation in original $\tau$, $\rho$ and $\alpha$.
  For $\gamma$'s we choose 10 random points, one of which is a discontinuity point.

- Run the parameter estimator 10 times.

- Results

<table>
<thead>
<tr>
<th>Estimator Choice</th>
<th>FCN</th>
<th>Iterations</th>
<th>Time</th>
<th>OBJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Simple</td>
<td>1003787</td>
<td>696.3</td>
<td>93.52</td>
<td>$8.62827e - 06$</td>
</tr>
<tr>
<td>SensJac</td>
<td>60604</td>
<td>16.8</td>
<td>5.399</td>
<td>$8.62779e - 06$</td>
</tr>
<tr>
<td>SensJac AddedCons</td>
<td>11708</td>
<td>3.2</td>
<td>1.056</td>
<td>$8.62779e - 06$</td>
</tr>
</tbody>
</table>