Dynamical systems with delay

David A.W. Barton
Lloyds Tercentenary Foundation Fellow
Bristol Centre for Applied Nonlinear Mathematics

6th September 2007

Introduction

Delay differential equations (DDEs)

\[ x'(t) = f(x(t), x(t - \tau_1), \ldots, x(t - \tau_i)) \]

- Infinite dimensional dynamical systems (“simple PDEs?”)
  \( \Rightarrow \) even 1st order systems can have complicated dynamics!
- Forward evolution is uniquely determined by a history segment
  \( x_t := x(t+s) \) for \( -\tau_1 \leq s \leq 0 \)
- Lots of applications!
  - Often a modelling decision to simplify model...

Outline

- Motivation: Why look at DDEs?
- Theory: Formal definition, Numerical methods, Stability & bifurcations, Abstract ODEs & centre manifolds, Other types of DDEs
- Practice: DDE-BIFTOOL example, DDE-BIFTOOL lab (afternoon)

Motivation

Motivation — where do we find delay?

Engineering
- Control systems
- Lasers
- Traffic models
- Metal cutting
- Transmission lines

Nature
- Epidemiology
- Cell cycle
- Protein production
- Population dynamics
- Neuroscience

Motivation — how does delay arise?

Delays are everywhere
- Due to finite response times delays can be put into almost any system
  - particularly high-level models
    - e.g., delayed logistic equation
  - Often interesting but only for unrealistically high delays
  - ODEs are normally sufficient
  - ... but there are times when they are not!

Ways can DDEs arise
- Delay arises directly in the model
- Delay is put in as a simplification

Metal cutting

- Delay time is period of revolution
- Need stability boundaries for stable cutting
- Faster cutting is better

Two degree of freedom model of turning

\[ x''(t) + 2\zeta x'(t) + x(t) = \frac{1}{K_0}K_1p^{\beta-1} \left( \frac{\tau(x)}{\tau_0} + y(t - \tau(x)) - y(t) \right)^q \]

\[ y''(t) + 2\zeta y'(t) + y(t) = K_1p^{\beta-1} \left( \frac{\tau(x)}{\tau_0} + y(t - \tau(x)) - y(t) \right)^q \]

\[ \tau(x)/\tau_0 = 1 + p(x(t) - x(t - \tau(x))) \]
Chaotic transmission lines

Studied experimentally by Blakely and Corron (2004, 2005)

Many applications... High frequency voltage oscillations (25–175 MHz)

⇒ even small delays not negligible

Modelled by

\[
C'x'(t) = -\left(\frac{1}{RC} - \frac{1}{R'}\right)x(t) + \left(\frac{1}{RC} - \frac{1}{R'}\right)g(x(t-\tau)) - Cg'(x(t-\tau))x'(t-\tau)
\]

Hybrid testing (real-time dynamic substructuring)

Combine numerical and physical — delays due to coupling

Save time and money by only building what you have to

Experiments in lab here (Bristol)

Epilepsy seizure modelling

Theory — fixed delays

DDEs with a single fixed delay

\[
x'(t) = f(x(t), x(t-\tau))
\]

Infinite dimensional state-space: \(x_t \in C([0, \tau], \mathbb{R}^n)\)

\(x_t(s) := x(t+s) \in \mathbb{R}^n\) for \(-\tau \leq s \leq 0\)

Chaos (and other complicated dynamics) in 1st order systems

\(x_t(0)\) is the head point

How to simulate?

- Discretise the state (equispaced points)
- Fixed stepsize Runge-Kutta where the time delay is an integer multiple of stepsize
- Store all the computed points for the previous \(\tau\) time
  - (Well suited to multi-step methods)

Fixed delays — numerics

Is it really that easy?

- What if we want an adaptive time-stepper?
- What happens with multiple delays?
- Or time-varying/state-dependent delays?

Need an interpolation scheme

(there are some extended Runge-Kutta methods for this)

In practice use DDE_SOLVER, RADAR5 or Matlab’s DDE23

One problem — discontinuity propagation

- Left and right derivatives at \(t = 0\) don’t always match!
- Left derivative is arbitrary; right derivative from DDE
- Discontinuity is propagated every \(\tau\) time
- ... but it occurs in a higher derivative each time
Fixed delays — breaking points

Fixed delays — stability

It does get harder — stability calculations
- Infinitely many eigenvalues (or Floquet multipliers)
- ... but only finitely many with positive real part
- Can linearise in similar way to ODEs:
  \[ x'(t) = f(x(t), x(t - \tau)) \]
  \[ y'(t) = A_0 y(t) + A_1 y(t - \tau) \]
  where \( A_0 := D f \) and \( A_1 := D^2 f \)
- (for periodic orbits \( A_0 \) and \( A_1 \) are time varying)
- ... and look for exponential solutions: \( y(t) = C \exp(\lambda t) \)

Fixed delays — spectrum

How to compute eigenvalues
- Naive approach: lots of random guesses plus Newton iteration
- Better approach #1: (DDE-BIFTOOL) provides dominant eigenvalues
  - Integrate linearised equations for time \( \tau \) (or so)
  - Extract dominant exponential growth factor
  - Factor this eigendirection out and repeat
  - Use end results as starting points for Newton iteration
- Better approach #2: (Breda et al.) provides eigenvalues closest to zero
  - Use collocation/pseudo-spectral approximation of infinitesimal generator
  - Calculate eigenvalues of resulting matrix

Fixed delays — bifurcations

Bifurcations of equilibria
- (Almost?) any ODE bifurcation can be seen in an appropriate 1st order delay system
- Centre manifold arguments show that any/most DDE bifurcations can be reduced to ODE equivalents
- NOT TRUE FOR OTHER TYPES OF DELAY EQUATION!

Bifurcations of other invariant sets
- Periodic orbits almost certainly same as ODE cases
- Complications arise with connecting orbits (hetero/homoclinic) due to infinite dimensional stable manifold...
- Other invariant sets...

Fixed delays — abstract ODEs

Abstract ODEs
DDEs can be written as an abstract ODE (alt: ODE on a Banach space)
\[ z'(t) = A z(t) + N(z(t)) \text{ for } z(t) \in B \]
(typically \( B \) is space of continuous functions on range \([-\tau, 0)\))
- \( A \) is a linear operator (infinitesimal generator)
- Combines evolution operator (from original DDE) with a shift operator
- \( N \) is a nonlinear operator containing all the nastiness
- Eigenvalues of \( A \) same as eigenvalues of linearised DDE
- Can use so-called Hale’s bilinear form to compute projections onto eigenvectors
  \[ \Rightarrow \text{calculate proj. boundary conditions for connecting orbits} \]

Fixed delays — centre manifolds

Centre manifolds
- Define centre manifold in same way as for ODEs
  - Split eigenspace into stable, unstable and centre manifolds
  - At a bifurcation point interested in dynamics on the centre manifold
    \( \text{e.g., determine criticality of Hopf bifurcation} \)
- Stable manifold \( \Rightarrow \) finite-dim, unstable manifold \( \Rightarrow \) finite-dim, centre manifold \( \Rightarrow \) finite-dim
- Use Hale’s bilinear form to project onto centre manifold
  \( \Rightarrow \) produces system of ODEs
- Can then do normal form calculations as usual
- All possible to do numerically!
  \( \text{e.g., 2-parameter continuation of change in Hopf criticality} \)

Other types of delay equations

Distributed delay differential equations
\[ x'(t) = f \left( x(t), \int_{-\tau}^0 K(s - t) x(s) \, ds \right) \]
- Integrate over all delay history with some kernel \( K \)
- Typically used as a “fading memory” (e.g., \( K(s) = \exp(s) \))
- If \( K \) has compact support then bifurcation theory (almost) identical to fixed delay case
Other types of delay equations

State-dependent delay differential equations (SD-DDEs)

\[ x'(t) = f(x(t), x(t - \tau(x_t))) \]

- The time delay varies
- Definition of \( \tau \) can be explicit or implicit
- Numerically not too much more difficult than fixed delays
- Theoretically much harder: lots of technical difficulties
- Hopf bifurcation theorem proved last year (2006)
- Centre manifold theorem?!?
- Hale’s bilinear form is not applicable…
- …but a very useful class of equations

Neutral delay differential equations

\[ x'(t) = f(x(t), x(t - \tau), x'(t - \tau)) \]

- Spectrum no longer has “nice” properties
- Possible for infinitely many eigenvalues to cross imaginary axis
- \( \Rightarrow \) Truely infinite dimensional bifurcations(?)
- Infinitesimal changes to \( \tau \) can cause arbitrarily large changes in the spectrum
- Discontinuity propagation is a problem as no smoothing properties
- Numerical problems due to eigenvalues with infinitely large imaginary part close to imaginary axis

Other types of delay equations

DDEs with piecewise-constant argument

\[ x'(t) = f(x(t), x(\tau)) \quad \tau = [t] \]

DDEs with piecewise-smooth RHS

\[ x'(t) = \begin{cases} 
f_1(x(t), x(\tau)) & \text{if } h(x(t), x(t - \tau)) \geq 0 \\
 f_2(x(t), x(\tau)) & \text{if } h(x(t), x(t - \tau)) < 0 
\end{cases} \]

Partial DDEs

- e.g.
  \[ u_{t0}(x, t) + \alpha^2 u_{xxxx}(x, t) = 0 \]
  \[ M_{tilde}(0, t) + K_{tilde}(0, t) + Cu(0, t) = \sin(\omega t) + u_{xxx}(0, t - \tau) \]
  ...

DDE-BIFTOOL

Continuation for constant and state-dependent DDEs

- Author: Koen Engelborghs (KU Leuven, Belgium) + others
- (Extension for NDDEs by David Barton, Bristol)
- Written in Matlab
- Programmatic interface to the continuation routines
  - Access to all internal data
  - Can pick individual points, view and modify on the fly
- Uses similar numerical algorithms to AUTO
- Very different interface compared with AUTO
- (Another package PDDE-CONT is more like AUTO)