Numerical bifurcation analysis of delay equations: a user-friendly MatCont interface

Davide Liessi
with Enrico Santi and Rossana Vermiglio

supported by the Italian Ministry of University and Research (MUR) through the PRIN 2020 project (No. 2020JLWP23) "Integrated Mathematical Approaches to Socio-Epidemiological Dynamics" (CUP: E15F21005420006)

CD Lab – Computational Dynamics Laboratory
Department of Mathematics, Computer Science and Physics
University of Udine, Italy

17th IFAC Workshop on Time Delay Systems
29 September 2022

MatCont: continuation for ODEs in MATLAB [1]

- time integration
- computation and continuation of equilibria, limit cycles, homoclinic and heteroclinic orbits
- detection and continuation of bifurcations
  (fold, Hopf, Neimark–Sacker, period doubling, Bogdanov–Takens, ...)
- detection of custom events
- GUI

$$x' = rx(1-x) \quad \rightsquigarrow \quad \texttt{x'=r*x*(1-x)}$$

Original version available at
`https://sourceforge.net/projects/matcont/`

[1] Dhooge, Govaerts, Kuznetsov, Meijer, and Sautois, *New features of the software MatCont for bifurcation analysis of dynamical systems*, Math. Comput. Model. Dyn. Sys., 14:2 (2008), pp. 147–175, DOI: 10.1080/13873950701742754.

Essentials on F. Scarabel's approach (DDE, $d = 1$) [2]

$$x'(t) = F(x_t), \qquad X := C([-\tau, 0]; \mathbb{R})$$

$$\left.\begin{array}{r} 0 = \theta_0 > \theta_1 > \cdots > \theta_M \geq -\tau \text{ nodes} \\ \ell_0, \ell_1, \ldots, \ell_M \text{ Lagrange polynomials} \end{array}\right\} \rightsquigarrow X_M := \mathbb{R} \times \mathbb{R}^M$$

$$P_M \colon X_M \ni (\Psi_0, \Psi) \mapsto \Psi_0 \ell_0(\cdot) + \sum_{j=1}^{M} \Psi_j \, \ell_j(\cdot) \in X$$

$$(d_M \mid D_M) := \left(\begin{array}{c|ccc} \ell_0'(\theta_1) & \ell_1'(\theta_1) & \ldots & \ell_M'(\theta_1) \\ \vdots & \vdots & \ddots & \vdots \\ \ell_0'(\theta_M) & \ell_1'(\theta_M) & \ldots & \ell_M'(\theta_M) \end{array}\right) \quad \text{differentiation matrix without first row}$$

$$\rightsquigarrow \begin{cases} x_M' = F(P_M(x_M, V_M)) \\ V_M' = d_M x_M + D_M V_M \end{cases}, \qquad x(t) \approx x_M(t)$$

[2] Breda, Diekmann, Gyllenberg, Scarabel, and Vermiglio, *Pseudospectral discretization of nonlinear delay equations: New prospects for numerical bifurcation analysis*, SIAM J. Appl. Dyn. Syst., 15:1 (2016), pp. 1–23, DOI: 10.1137/15M1040931.

Essentials on F. Scarabel's approach (RE, $d = 1$) [3]

$$x(t) = F(x_t), \qquad X := L^1([-\tau, 0]; \mathbb{R})$$

$$\rightsquigarrow \; X_M := \mathbb{R}^M \qquad \text{(the value in 0 is 0)}$$

$$P_M \colon X_M \ni \Psi \mapsto \sum_{j=1}^{M} \Psi_j \, \ell_j(\cdot) \in X$$

$$\rightsquigarrow \; V_M' = D_M V_M - F(P_M D_M V_M) \cdot \mathbf{1}$$

$$x(t) \approx F(P_M D_M V_M(t))$$

(V_M represents the integrated state)

[3] Scarabel, Diekmann, and Vermiglio, *Numerical bifurcation analysis of renewal equations via pseudospectral approximation*, J. Comput. Appl. Math., 397, 113611 (2021), DOI: 10.1016/j.cam.2021.113611.

Goals

enable MatCont's GUI to treat DDEs and REs directly:

- extend input to allow DDEs and REs
- discretize into ODEs internally
- hide auxiliary components of the solution from the user

$\rightsquigarrow$ user-friendly interface (no need to write/edit code)

Example: the Mackey–Glass equation (1)

$$y'(t) = \beta \frac{y(t-\tau)}{1 + y(t-\tau)^n} - \gamma y(t)$$

movie!

`https://youtu.be/4mJbbvIfDuI`

Example: the Mackey–Glass equation (2)

previously the user had to write the definition of the right-hand side (requiring knowledge of the method) in one file...

```
FM = @(x) beta*VM(end)./(1+VM(end).^n)-gamma*xM;
```

...and use MatCont's command line interface, requiring a rather deep knowledge of the software and being accustomed to writing code:

```
...
Weq=feval(handles{1},M,xeq,yeq);
[x0,v0]=init_EP_EP(@PS_MackeyGlass,Weq,par0,ap1);
[xe,ve,se,he,fe]=cont(@equilibrium,x0,v0,opt);
...
for ii=1:size(se)
    if strcmp(se(ii).label,'H ')==1
        H_index=se(ii).index;
        break;
    end
end
par(ap1)=xe(end,H_index);
H=xe(1:MM,H_index);
...
ntst=20; ncol=4;
[x0,v0]=init_H_LC(@PS_MackeyGlass,H,parH,ap1,1e-6,ntst,ncol);
[xlc,vlc,slc,hlc,flc]= cont(@limitcycle,x0,v0,opt);
...
```

using the GUI is much simpler!

Overview of changes

- new buttons and boxes in input GUI
- regular expressions for input interpretation
- generation of system definition files
- hide auxiliary components of the solution in GUI and plotting functions
- correctly compute solutions of REs with $x(t) \approx F(P_M D_M V_M(t))$

- 26 changed files (23 GUI, 3 auxiliary), 4 added files
- almost 2000 lines of new code

New syntax for delay equations

- time dependency of coordinates for current time terms and discrete delays

$$y(t-1) \quad \rightsquigarrow \quad \texttt{y[t-1]}$$
$$y(t),\, y'(t) \quad \rightsquigarrow \quad \texttt{y[t], y'[t]} \quad \text{or} \quad \texttt{y, y'}$$

- integrals for distributed delays

$$\int_{-3}^{-1} x(t+s)(1-x(t+s))\,\mathrm{d}s$$
$$\rightsquigarrow \quad \texttt{\textbackslash int\_\{-3\}\^{}\{-1\}\{x[t+s]*(1-x[t+s])\}\{s\}}$$

- DDEs `y'[t] = ...`
- REs `x[t] = ...`

Numerical choices

- Chebyshev extremal nodes for interpolation
- Clenshaw–Curtis quadrature

Future work

- fix the real-time plotting of solutions of REs
  (requires changes to MatCont's core functions)
- cache the computed solutions of REs
- make the numerical parameters (degree of interpolation and quadrature) configurable during the analysis
- optimize
- submit the changes to MatCont's authors for integration in the main distribution (hopefully!)
  (or maybe fork the project…)
- further extend MatCont to support state dependent delays, infinite delays, …

Download

CD Lab

http://cdlab.uniud.it/software

Download

CD Lab

http://cdlab.uniud.it/software

Thank you!