Access to the Solver of Functional Differential Equations Through the Web Interface

SVYATOSLAV I. SOLODUSHKIN
Ural State University
Department of Computational Mathematics
Turgeneva Street 4, 620000 Ekaterinburg
RUSSIA
solodushkin_s@mail.ru

Abstract: This paper contains presentation of Web server that provides an interface to a MATLAB solver of functional differential equations in ordinary and partial derivatives. ASP.NET4/C# was used to elaborate system and COM technology to bind these programs with MATLAB.

Key–Words: Functional Differential Equations, MATLAB, Numerical methods

1 Introduction

Functional differential equations (FDE) in partial and ordinary derivatives are widely used for describing and mathematical modeling various processes and systems with delay [4, 6]. At present theoretical aspects of FDE are elaborated with almost the same completeness as the corresponding parts of ordinary differential equations (ODE) and differential equations in partial derivatives (PDE).

However unlike ODE and PDE even for linear FDE there no general methods of finding solutions in explicit forms. So elaboration of numerical methods and their programm realization for FDE is a very important problem.

Numerical methods are elaborated quite well both for ODE and PDE for; moreover these methods are presented in the form of standard packets and toolboxes for use with MATLAB.

Over the past ten years numerical algorithms were constructed for general forms FDE in ordinary derivatives [2, 4, 5]. Per contra numerical algorithms for FDE in partial derivatives are elaborated much more poor. It’s possible to mention just the works [1, 3, 7, 8]. The numerical methods which were described in [4, 7, 8] are expressed mostly as M-files for MATLAB.

The goal of this work is to present information and computing server that allow one to solve (FDE) in partial and ordinary derivatives. This server is a result of efforts of chair of Computational Mathematics of the Ural State University, headmaster prof. Pimenov.

2 The class of solvable equations and corresponding algorithms

Systems with delays

\[
\dot{x}(t) = f(t, x(t), x(t+\tau)), \quad -\tau \leq s < 0, \quad (1)
\]

are infinite dimensional systems because of the presence of the functional component \( \{x(t+s), \quad -\tau \leq s < 0\} \), which characterizes delays. One of the problem of simulations of such systems consist in describing of \( f(t, x(t), x(t+s)) \) by finite number of parameters, because for computer simulations usually only finite algorithms with finite number of input parameters can be used.

Analyzing the structure of systems with delays one can see that in concrete cases right-hand sides of such systems are combinations of finite dimensional functions and integrals. Numerical algorithms for simulating nonlinear time-delay systems of the following forms were realized

\[
\dot{x}(t) = f(t, x(t), x(t-\tau_1), \int_{-\tau_2}^{0} \gamma(t, s)x(t, s)ds),
\]

(2)

\[
\dot{x}(t) = f(t, x(t), x(t-\tau_1), \int_{-\tau_2}^{0} \phi(x(t, s))ds),
\]

(3)

where \( f(\cdot, \cdot, \cdot, \cdot) : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \); \( \gamma(\cdot, \cdot) \) is \( m \times n \) matrix with continuous elements on \( \mathbb{R} \times [-\tau_2; 0] \); \( \phi(\cdot) : \mathbb{R} \to \mathbb{R}^m \); \( \tau_1, \tau_2 \) are positive constants. The right-hand sides of systems (2), (3) can be described just by finite numbers of functions.

It’s possible to consider initial conditions of systems with delays as the pair \( \{x^0, x^0(t+s)\}, \quad -\tau \leq s < 0 \)
0, where \( x^0 \in \mathbb{R} \) and a function \( x^0(\cdot) \in C[-\tau; 0) \) and \( \lim_{s \to 0} x^0(s) = x^0 \).

Runge-Kutta-like 4(5) order numerical methods with automatic step size are realized.

To simulate the system user must input the following data:
1) right-hands side of the system,
2) initial conditions
   2.1 initial time moment \( t_0 \)
   2.2 initial function \( x^0(t + s), -\tau \leq s < 0 \)
   2.3 initial value vector-column \( x^0 \),
3) final time moment \( t_f \).

To describe right-hand side standard MATLAB syntax is used, the limped delay terms \( x(t - \tau_i(s)) \) and the distributed delay terms \( \int_{-\tau_f(t)}^{0} \gamma(t, s)x(t, s) \, ds \)
\( \int_{-\tau_f(t)}^{0} \phi(x(t, s)) \, ds \) are computed by means of the functions \( \text{xdelay}, \text{int1}, \text{int2} \) respectively. The input parameters for \( \text{int1}(l, u, f, num) \) are
\( l \) Low limit of integration — a function \( -\tau(t) \),
\( u \) Up limit of integration — usually 0,
\( f \) String variable with matrix \( \gamma \) depending on parameter \( s \) and \( t \),
\( num \) Optional parameter — number of coordinate \( x \) returned by the function \( \text{int1} \) in case of multidimensional system.

The input parameters for \( \text{int2}(l, u, f, num) \) are
\( l \) Low limit of integration — a function \( -\tau(t) \),
\( u \) Up limit of integration — usually 0,
\( f \) String variable with matrix \( \gamma \) depending on parameter \( s \),
\( num \) Optional parameter — number of coordinate \( x \) returned by the function \( \text{int2} \) in case of multidimensional system.

The input parameters for \( \text{xdelay}(d, num) \) are
\( d \) Optional parameter — value of delay \( \tau \)
\( num \) Optional parameter — number of coordinate \( x \) returned by the function \( \text{xdelay} \) in case of multidimensional system.

For example to solve nonlinear system with distributed delay of value \( \pi \)

\[
\dot{x}_1(t) = -\frac{1}{2} \int_{-\pi}^{0} x_1(t + s) \, ds + \frac{2x_1(t) - \pi/2x_2(t)}{\sqrt{x_1^2(t) + x_2^2(t)}}
\]
\[
\dot{x}_2(t) = -\frac{1}{2} \int_{-\pi}^{0} x_2(t + s) \, ds + \frac{2x_2(t) + \pi/2x_1(t)}{\sqrt{x_1^2(t) + x_2^2(t)}}
\]
on the segment \([t_0; t_f] = [1; 10]\) with the initial state

\[
\dot{x}(t) = \begin{pmatrix} t \cos(t) \\ t \sin(t) \end{pmatrix}, \text{for} \ t \geq 1 - \pi
\]

As a result user obtains 2-dimensional array, where (1) the first column corresponding to the time grid \( t \) which is builded automatically (the first moment is \( t_0 \) and the last is \( t_f \)); (2) all other columns (as many as the system dimension) contains the values of approximate solution respectively evaluated at the time points of \( t \).
Heat conduction equation of a general form with time delay is considered too
\[
\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} + f(x, t, u(x, t), u_t(x, \cdot)).
\] (4)

3 Developed application architecture

Remote user interact with systems by means of his Web browser over HTTP. The developed application we describe here is a ligament of Web server and MATLAB. To organize this interaction between Web server and MATLAB, COM technology is used. Dynamic pages are created on ASP.NET4/C#, MATLAB programs implement numerical algorithms are expressed as M-files for MATLAB.

After user click the button "Calculate" event handler written in C# create an instance of COM object

```csharp
MLApp.MLAppClass matlab = new MLApp.MLAppClass();
```

In this architecture MATLAB is an Automation Server, an interaction with the Automation Server is performed through an interface that COM object provide: functions Execute, Feval, PutFullMatrix, GetFullMatrix.

4 Conclusion

Elaborated system allows one to easily add new algorithms, which are implemented as M-files for MATLAB. For that it’s necessary to construct an appropriate Web page with Web form, where user can input a description of his system. The system has a modular structure that will allow in further work on the parallel MATLAB, while the Web interface does not change.

Acknowledgements: The research was supported by the RFBR (grant No. 10-01-00337) Mathematical models and numerical methods for functional-differential equations in partial derivatives.

References: