A numerical method for boundary value problems of delay differential algebraic equations with advanced and retarded delays

Alastair Flynn

3 October 2019

Contents

1 Introduction

This document describes a numerical method used to solve boundary value problems of delay differential algebraic equations with advanced and retarded delays.

A collocation method has been developed to solve boundary value problems of delay differential algebraic equations. As with other collocation methods, a functional form is chosen for the variables in the system and a system of non-linear equations is constructed such that a solution of the non-linear system is a solution of the BVP at some chosen collocation points.

2 The class of problems to be tackled

As few assumptions as possible on the structure of the problem are made. The advantage of this is that the code is applicable to a broad class of problems, the disadvantage is that it requires more user input to run the code.

Let $\mathbf{u}$ be a function of $s \in [a, b]$ and $\mathbf{p}$ a vector of parameters. The class of problems to be solved may be written in its most general form as

$$\mathbf{f}(\mathbf{u}, \mathbf{u}', s; \mathbf{p}) = 0 \tag{2.1}$$

with boundary conditions

$$\mathbf{b}(\mathbf{u}, \mathbf{u}', s; \mathbf{p}) = 0 \tag{2.2}$$

The function $\mathbf{f}$ may depend on $\mathbf{u}$ and $\mathbf{u}'$ evaluated at any point, including outside of the interval $[a, b]$, in which case, $\mathbf{u}$ (and $\mathbf{u}'$) must be known outside of $[a, b]$. The boundary conditions may be given at any points in $[a, b]$. The interval $[a, b]$ may be given explicitly or it may be defined by extra boundary conditions; the way in which these boundary conditions are included in the system is described in section 3. There are problems in this class that have infinitely many solutions and there are problems in this class that have no solution. The code has no way of detecting this, it just fails to converge.

This class of problems includes (but is not restricted to):

- Ordinary differential equations
- Purely algebraic equations
- Differential algebraic equations
- Delay differential algebraic equations with advanced and retarded delays

In the method, a distinction is made between *differential variables* whose derivative appears in $\mathbf{f}$ and *algebraic variables* whose derivative does not appear. The dimension of $\mathbf{b}$ is equal to the number of differential variables.

3 Coordinate transformation & mesh selection

A coordinate transformation is applied in order to rescale the problem so that it can be solved efficiently on a uniform mesh [1]. Let $t \in [0, 1]$ and let $s(t)$ be a function of t. We now wish to solve the transformed problem

$$\mathbf{f}\left(\mathbf{u}, \frac{\mathbf{u}'}{s'}, s; \mathbf{p}\right) = 0 \tag{3.1}$$

where $\mathbf{u}$ and s are now functions of t. We add the following ordinary differential equation to the system to define $s(t)$.

$$s'(t) = \frac{L}{\sqrt{1 + \|\mathbf{u}(t)\|}} \tag{3.2}$$

where the norm is the 2-norm taken over differential variables only. The *coordinate transform* s is treated as an extra (differential) variable.

We must now include a boundary condition for s and another boundary condition for the unknown scale L. These are the boundary conditions that define the interval $[a, b]$. For example, if the interval $[a, b]$ is know explicitly then the corresponding boundary conditions would be $s(0) = a$ and $s(1) = b$. The extra boundary conditions must be specified by the user, thus, the number of boundary conditions that must be specified is equal to the number of differential variables plus two.

If $\mathbf{f}(\mathbf{u}, \mathbf{u}', s; \mathbf{p}) = \mathbf{f}_1(\mathbf{u}(\sigma(s)))$ depends on a delayed variable then once the coordinate transform is applied, $\sigma : [0, 1] \to [a, b]$ is a function of t that returns a value in $[a, b]$. Thus, the transformed system is

$$\mathbf{f}_1(\mathbf{u}(s^{-1}(\sigma(s(t))))) = 0 \tag{3.3}$$

The inverse s^{-1} of s is included in the system as an extra (algebraic) variable. So, in order to fully define the coordinate transform, we have added an extra differential variable s and an extra algebraic variable s^{-1}.

The full system now looks as follows

$$0 = \mathbf{f}\left(\mathbf{u}, \frac{\mathbf{u}'}{s'}, s; \mathbf{p}\right) \tag{3.4}$$

$$s'(t) = \frac{L}{\sqrt{1 + \|\mathbf{u}(t)\|}} \tag{3.5}$$

$$t = \frac{s(s^{-1}(t))}{s(1)} \tag{3.6}$$

The user must take care to incorporate the coordinate transform when defining the system to be solved, in particular, the rescaling of derivatives of $\mathbf{u}$ (this is not done automatically, at least for now). From now on we always deal with the rescaled system and write it as $\mathbf{f}(\mathbf{u}, s; \mathbf{p})$ for short.

4 Discrete representation of the variables

The interval $[0, 1]$ is partitioned into K subintervals, upon each of which, each variable is represented by a polynomial. Chebyshev polynomials of the first kind are used as basis polynomials.

The user may select the degree of the polynomials representing the algebraic variables, the differential variables are represented by polynomials of one degree higher. This is because the differential variables are expected to be one degree smoother than the algebraic variables and this should be reflected in the discretisation. This has been suggested, for example in [1], for systems that can be separated into differential and algebraic variables.

The differential variables are required to be continuous by including continuity conditions in the non-linear system of equations that will be constructed in section 5. Only C^0 continuity is required, that is

$$p_{i,k+1}(s_k) = p_{i,k}(s_k) \quad \text{for} \quad i = 1, \ldots, M+1 \quad k = 1, \ldots, K-1 \tag{4.1}$$

where $p_{i,k}$ is the polynomial representing the ith variable on the kth subinterval and s_k are the *breakpoints* between subintervals with the $M + 1$st variable being the coordinate transform.

Let d be the degree of polynomials representing the algebraic variables, M the number of differential variables and N the number of algebraic variables then the number of polynomial coefficients to be determined is

$$\underbrace{MK(d+2)}_{\text{Differential variables}} + \underbrace{NK(d+1)}_{\text{Algebraic variables}} \tag{4.2}$$

We must add to this the $K(d+2)+K(d+1)$ coefficients for the extra coordinate transform variables and the unknown scale L. Thus, the total number of unknowns is

$$(M+1)K(d+2) + (N+1)K(d+1) + 1 \tag{4.3}$$

5 The system of non-linear equations

Within each subinterval, *collocation points* are defined by the Gauss–Legendre quadrature points (rescaled to the subinterval). Denote by S the set of all collocation points.

Evaluate the system $\mathbf{f}(\mathbf{u}, s; \mathbf{p})$ and the extra equations for the coordinate transform at the collocation points S to get a vector $\mathbf{F}$ of dimension $(N + M + 2)K(d + 1)$. Let

$$\mathbf{C} = (p_{i,k+1}(s_i) - p_{i,k}(s_k))_{i=1,\ldots,M+1\ k=1,\ldots,K-1} \tag{5.1}$$

be the vector of continuity conditions of dimension $(M + 1)(K - 1)$ (which are only evaluated for differential variables). Also, evaluate the boundary conditions $\mathbf{b}(\mathbf{u}, s; \mathbf{p})$ plus the extra boundary conditions to get a vector $\mathbf{B}$ of dimension $M + 2$.

Let

$$G(\mathbf{c}; \mathbf{p}) = \begin{pmatrix} \mathbf{F} \\ \mathbf{C} \\ \mathbf{B} \end{pmatrix} \tag{5.2}$$

be the concatenation of these vectors where $\mathbf{c}$ contains the polynomial coefficients and the scale L. Then the problem is reduced to the non-linear system

$$G(\mathbf{c}; \mathbf{p}) = 0 \tag{5.3}$$

of dimension

$$(N+M+2)K(d+1)+(M+1)(K-1)+M+2=(M+1)K(d+2)+(N+1)K(d+1)+1 \quad (5.4)$$

The user has a choice of non-linear solvers including a global Newton method [3] and the Levenberg–Marquardt method [2].

References

[1] U. Ascher, R. Mattheij, and R. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations.* Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, January 1995.

[2] J. Dennis and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, January 1996.

[3] Peter Deuflhard. Systems of Equations: Global Newton Methods. In *Newton Methods for Nonlinear Problems*, Springer Series in Computational Mathematics, pages 109–172. Springer, Berlin, Heidelberg, 2011.