New functionality in DDE-BIFTOOL v. 2.03. Addendum to the manual of DDE-BIFTOOL v. 2.00 (and v. 2.02).

Koen Verheyden

March 2007

Summary

This document describes briefly the new functionality in DDE-BIFTOOL v. 2.03 compared to the previous version, DDE-BIFTOOL v. 2.02. The functionality of the previous version is described in the manual [2]. See also the paper [1].

The new functionality added to DDE-BIFTOOL v. 2.03 is the computation of the rightmost characteristic roots of steady state points using the *new steplength heuristic* (including the use of the *linear multistep methods of maximal order*) presented in [4]. In the previous versions of DDE-BIFTOOL another steplength heuristic, which is presented in [3], was implemented. Remark that it is still possible to use the latter heuristic in version 2.03 of DDE-BIFTOOL; however note that the new heuristic is the default choice. Users of this new version of DDE-BIFTOOL do *not* have to change their existing code which was implemented for a previous version.

The software package DDE-BIFTOOL v. 2.03 can be downloaded from the web page `http://www.cs.kuleuven.ac.be/~twr/research/software/delay/ddebiftool.shtml`. This web page contains also links to the manual [2] and this addendum.

Examples of use

The next section lists all the functions which were changed or added compared to the previous version of DDE-BIFTOOL (v. 2.02). Only the following changes/additions are important for the end user:

- The addition of a second, optional argument to `df_mthod.m` which can be used to choose explicitly the old steplength heuristic, presented in [3], over the new steplength heuristic, presented in [4] (which is the default).
- The new function `stst_stabil.m` which computes the rightmost characteristic roots using the new steplength heuristic.

Note that users of this new version of DDE-BIFTOOL do *not* have to change their existing code which was implemented for a previous version.

1) If the new steplength heuristic is chosen

In this case, one does not have to change the way DDE-BIFTOOL functions are called. This is illustrated by the first excerpt of code below, which was written for the first version of DDE-BIFTOOL.

The following excerpt of code (taken from the file `system/demo.m` which is included in the DDE-BIFTOOL distribution). The important part of the code is the fourth line (in a framed box) where the new heuristic is chosen by default.

```
stst.kind='stst';
stst.parameter=[1/2 -1 1 2.34 0.2 0.2 1.5];
stst.x=[0 0]'
method=df_mthod('stst')
method.stability.minimal_real_part=-1;
[stst,success]=p_correc(stst,[],[],method.point)
stst.stability=p_stabil(stst,method.stability);
figure(1); clf;
p_splot(stst);
```

It is also possible to make the choice for the new steplength heuristic explicitly visible, as follows. The important part of the code are now the fourth and fifth lines (in framed boxes).

```
stst.kind='stst';
stst.parameter=[1/2 -1 1 2.34 0.2 0.2 1.5];
stst.x=[0 0]'
flag_newhheur=1;
method=df_mthod('stst',flag_newhheur)
method.stability.minimal_real_part=-1;
[stst,success]=p_correc(stst,[],[],method.point)
stst.stability=p_stabil(stst,method.stability);
figure(1); clf;
p_splot(stst);
```

2) If the old steplength heuristic is preferred

In this case, one has to change the fourth and fifth lines (in framed boxes) of the code above as follows:

```
stst.kind='stst';
stst.parameter=[1/2 -1 1 2.34 0.2 0.2 1.5];
stst.x=[0 0]'
flag_newhheur=0;
method=df_mthod('stst',flag_newhheur)
method.stability.minimal_real_part=-1;
[stst,success]=p_correc(stst,[],[],method.point)
stst.stability=p_stabil(stst,method.stability);
figure(1); clf;
p_splot(stst);
```

Detailed list of changes and additions

List of files which were changed compared to DDE-BIFTOOL v. 2.02 :

`p_stabil.m` Changed so that both the old and the new steplength heuristic can be used. If the new steplength heuristic is chosen, this function calls `stst_stabil.m`.

`df_mthod.m` A second, optional argument is added to this function. This allows the user to choose either the old steplength heuristic or the new heuristic. The new steplength heuristic is the default choice in the case that this second argument is omitted. If a second argument is present and it equals zero, the old steplength heuristic is chosen.

`time_lms.m` This function can now also return the coefficients of the linear multistep methods of maximal order.

List of files which were added compared to DDE-BIFTOOL v. 2.02 :

`stst_stabil.m` A function for the computation of the rightmost characteristic roots by using the new steplength heuristic. This function assumes that the `method` structure (returned by `df_mthod.m`) contains the correct information which is necessary for the new steplength heuristic. This is the case if `df_mthod.m` is called *without* a second argument or if this argument is nonzero. (In `stst_stabil.m`, this is checked by looking at the imaginary part of `method.lms_parameter_rho`. If this is zero, an error message is displayed which includes some more explanation on how to call `df_mthod.m`.)

`help_stst_stabil.m` A helper function to obtain the eigenvalues of an approximate matrix eigenvalue problem obtained by the linear multistep method. This function is called by `stst_stabil.m`.

`stst_stabil_nwt_corr.m` A helper function which uses a Newton iteration to correct the approximate characteristic roots computed from the eigenvalues of the approximate matrix eigenvalue problem. This function is called by `stst_stabil.m`.

`get_pts_h_new.m` A helper function which computes points in the complex plane used in the estimation of the set $\Omega(\cdot)$ which is used in the new steplength heuristic, cf. [4]. This function is called by `stst_stabil.m`.

`get_lms_ellipse_new.m` A helper function which returns the a_{ell} and b_{ell} values determining the ellipse used in the new steplength heuristic, cf. [4]. This function is called by `df_mthod.m`.

`get_S_h_row.m` A helper function which returns part of the approximate matrix eigenvalue problem. This function is called by `help_stst_stabil.m`.

`mu_to_lambda.m` A helper function which computes the approximate characteristic roots ($\tilde{\lambda}$) given the approximate characteristic exponents ($\tilde{\mu}$). This function is called by `stst_stabil.m`.

References

[1] K. Engelborghs, T. Luzyanina, and D. Roose, *Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL*, ACM Trans. Math. Softw., 28 (2002),

pp. 1–21.

[2] K. Engelborghs, T. Luzyanina, and G. Samaey, *DDE-BIFTOOL v. 2.00: a Matlab package for numerical bifurcation analysis of delay differential equations*, Report TW 330, Department of Computer Science, K.U.Leuven, Leuven, Belgium, 2001. Available from http://www.cs.kuleuven.be/∼twr/research/software/delay/ddebiftool.shtml.

[3] K. Engelborghs and D. Roose, *On stability of LMS methods and characteristic roots of delay differential equations*, SIAM J. Numer. Anal., 40 (2002), pp. 629–650.

[4] K. Verheyden, T. Luzyanina, and D. Roose, *Efficient computation of characteristic roots of delay differential equations using LMS methods*, J. Comput. Appl. Math., (2007). accepted (DOI:10.1016/j.cam.2007.02.025).