TRACE-DDE: a Tool for Robust Analysis and Characteristic Equations for Delay Differential Equations

Dimitri Breda[1], Stefano Maset[2], and Rossana Vermiglio[3]

[1] Dipartimento di Matematica e Informatica, Università degli Studi di Udine, via delle Scienze, 206, I-33100 Udine, Italy dimitri.breda@dimi.uniud.it
[2] Dipartimento di Matematica e Informatica, Università degli Studi di Trieste, via Valerio 12, I-34127 Trieste, Italy maset@units.it
[3] Dipartimento di Matematica e Informatica, Università degli Studi di Udine, via delle Scienze, 206, I-33100 Udine, Italy rossana.vermiglio@dimi.uniud.it

Summary. In the recent years the authors developed numerical schemes to detect the stability properties of different classes of systems involving delayed terms. The base of all methods is the use of pseudospectral differentiation techniques in order to get numerical approximations of the relevant characteristic eigenvalues. This chapter is aimed to present the freely available Matlab package TRACE-DDE devoted to the computation of characteristic roots and stability charts of linear autonomous systems of delay differential equations with discrete and distributed delays and to resume the main features of the underlying pseudospectral approach.

1 Introduction

It is nowadays widely recognized that many real phenomena in physics, engineering, chemistry, biology, economics etc. are better modeled and simulated if time delays are taken into consideration. Delay systems are fundamental in control theory, where the effects of delays on stability are a crucial problem [19], [21], [18], but key applications can be found also in machining tool where the role of parameters such as spindle speed and feed are stability determining [23], [16]. Moreover, time delays are concerned in other fields involving different models whose stability characteristics are important, e.g. age-structured population dynamics [4], neutral and advanced-retarded functional differential equations [9] and partial differential equations with delays [10].

All delay systems share the common feature of being influenced, in their present evolution, by information on their past history. Much of their interest is concerned with the asymptotic stability analysis of the linear (or linearized) case. The lack of good estimates of the parameter values involved in system models (e.g. delays, but not only) leads to develop suitable criteria to determine not only whether a nominal system is stable or not, but an entire stability region of parameters due to this uncertainty (robust stability). In particular, when we deal with two variable parameters, we talk about *stability charts*.

J.J. Loiseau et al. (Eds.): Topics in Time Delay Systems, LNCIS 388, pp. 145–155.
springerlink.com

In order to tackle the two above questions (nominal and robust stability), several techniques have been proposed in the more or less recent literature. In the exhaustive monograph [19] many graphical and analytical tests to this aim are reported, but in general they work only for restricted classes of delay differential equations (DDEs), e.g. constant delays, single-delay dynamics, commensurate delays and so on. Here we consider a system of m linear DDEs with multiple discrete and distributed delays:

$$y'(t) = L_0 y(t) + \sum_{l=1}^{k} L_l y(t-\tau_l) + \int_{-\tau}^{0} M(\theta) y(t+\theta) d\theta, \ t \geq 0, \tag{1}$$

where $L_0, L_1, \ldots, L_k \in \mathbb{C}^{m\times m}$, $0 = \tau_0 < \tau_1 < \ldots < \tau_k = \tau$ and $M : [-\tau, 0] \to \mathbb{C}^{m\times m}$ is a piecewise smooth function. Indeed in the implementation of the method we write the distributed term as

$$\int_{-\tau}^{0} M(\theta) y(t+\theta) d\theta = \sum_{l=1}^{k_d} \int_{-\tau_l}^{-\tau_{l-1}} M_l(\theta) y(t+\theta) d\theta, \ t \geq 0,$$

where M_l, $l = 1, \ldots, k_d$, is smooth (without loss of generality one can assume $k_d = k$). However, here we restrict to the standard case (1) with smooth M.

In the last few years we devoted our research activity to the numerical computation of the characteristic roots of (1), i.e. the (infinitely many) roots of $\det(\Delta(\lambda)) = 0$ where

$$\Delta(\lambda) = \lambda I - \sum_{l=0}^{k} L_l e^{-\lambda\tau_l} - \int_{-\tau}^{0} M(\theta) e^{\lambda\theta} d\theta, \ \lambda \in \mathbb{C},$$

since it is well known that the zero solution of (1) is asymptotically stable if and only if these roots have strictly negative real part ([11]).

Several numerical approaches for characteristic roots computation have been proposed, which are based on the discretization of either the solution operator associated to (1) or the infinitesimal generator of the solution operator semigroup. We briefly recall that the solution operator $T(t)$, $t \geq 0$, associated to (1) is defined by $T(t)\varphi = y_t$, $\varphi \in X$, where $X = C([-\tau, 0], \mathbb{C}^m)$ is endowed with the maximum norm, y_t is the *state* of the system, i.e. the function $y_t(\theta) = y(t+\theta)$, $\theta \in [-\tau, 0]$, and y is the solution of (1) with initial data $y_0 = \varphi \in X$ at $t = 0$. The family $\{T(t)\}_{t\geq 0}$ is a C_0-semigroup with infinitesimal generator $\mathcal{A} : D(\mathcal{A}) \subseteq X \to X$ given by

$$\mathcal{A}\varphi = \varphi', \ \varphi \in D(\mathcal{A}), \tag{2}$$

with domain

$$D(\mathcal{A}) = \left\{ \varphi \in X \ : \ \varphi' \in X, \ \varphi'(0) = \sum_{l=0}^{k} L_l \varphi(-\tau_l) + \int_{-\tau}^{0} M(\theta)\varphi(\theta) d\theta \right\}. \tag{3}$$

So (1) can be restated as the following abstract Cauchy problem ([11]) on the state space X:

$$\begin{cases} \dfrac{du}{dt}(t) = \mathcal{A}u(t), \ t > 0, \\ u(0) = \varphi \in X, \end{cases}$$

whose solution is $u(t) = y_t$ whenever $\varphi \in D(\mathcal{A})$. The two following important results ([11], [15])

1. $\det(\Delta(\lambda)) = 0 \Leftrightarrow \forall t > 0,\ \lambda = \frac{1}{t}\ln\mu$ for some $\mu \in \sigma(T(t)) \setminus \{0\}$;
2. $\det(\Delta(\lambda)) = 0 \Leftrightarrow \lambda \in \sigma(\mathcal{A})$;

where $\sigma(\cdot)$ denotes the spectrum, suggest the idea to turn the characteristic roots approximation problem into a corresponding eigenvalue problem for suitable matrix discretization of either $T(t)$ (*solution operator* approach) or $\mathcal{A}$ (*infinitesimal generator* approach).

Engelborghs and Roose proposed in [13] the solution operator approach via linear multistep (LMS) time integration of (1) without distributed delay term. Their method computes approximations to the roots from a large, standard and sparse eigenvalue problem and it is implemented in the Matlab package DDE-BIFTOOL for DDEs bifurcation analysis [12]. The distributed delay case is considered in [17] by using LMS methods and in [3] by using Runge-Kutta (RK) methods. The complete development of the infinitesimal generator approach first appeared in [5] and [2] where a matrix approximation $\mathcal{A}_N$ of $\mathcal{A}$ is obtained discretizing the derivative in (2) by RK or LMS method, respectively, ending up with a large and sparse eigenvalue problem as in [13]. Finally, the infinitesimal generator approach via pseudospectral differencing methods has been proposed in [8]. The technique is based on the exact differentiation of interpolants at selected sets of $N + 1$ nodes. Although the resulting differentiation matrix is non-sparse, the advantage of the well-known "spectral accuracy" (see [24] and the bibliography therein) allows very accurate approximations with small matrix dimension. This behavior represents in fact, for sufficiently small tolerance, the outstanding advantage of this method compared to the above mentioned schemes. Among those methods for stability detection not based on the semigroup structure of the problem it is worthy to mention the Cluster Treatment of Characteristic Roots (CTCR) by Olgac and Sipahi (see e.g. [20]) and the Quasi Polynomial mapping based Root-finder (QPmR) by Vyhlídal and Zítek (see [25]).

The chapter is organized as follows. In Section 2 we present the Graphic User Interface (GUI) Matlab package TRACE-DDE, acronym for Tool for Robust Analysis and Characteristic Equations for Delay Differential Equations. The software allows an efficient and reliable determination of the characteristic roots and the stability chart of systems of DDEs like (1). Its usage is explained by following a tutorial example while the underlying methods are recalled in the forthcoming sections. In particular, in Section 3 we shortly recover from [8] the basic facts about the numerical computation of characteristic roots of (1) through the pseudospectral discretization of the associated infinitesimal generator. Then, in Section 4, we recall the main lines of a recent work of the authors on the approximation of level curves of surfaces ([6],[7]). The chapter is closed with several experimental results showing features, appearance and performances of the new tool.

2 The graphic user interface

TRACE-DDE is a Matlab package equipped with a GUI devoted to the robust analysis and the computation of the characteristic roots of delay systems entering the class (1). The software is publicly available (`http://users.dimi.uniud.it/~dimitri.breda/software.html`) and its core is made of the two algorithms briefly described in the forthcoming Sections 3 and 4.

In this section we limit ourselves to a brief description through the computation of the roots and of a stability chart for the system [14]

$$y'(t) = L_0 y(t) + L_1 y(t-a) + \int_{-b}^{0} M(\theta) y(t+\theta) d\theta \tag{4}$$

with coefficients matrices

$$L_0 = \begin{pmatrix} -3 & 1 \\ -24.646 & -35.430 \end{pmatrix}, L_1 = \begin{pmatrix} 1 & 0 \\ 2.35553 & 2.00365 \end{pmatrix}, M = \begin{pmatrix} 1 & -\theta \\ 1 & -\theta \end{pmatrix},$$

where a and b are uncertain parameters both with nominal value 1.

Once the source package has been downloaded, thanks to the graphical suite, the user is required only to type `tracedde` at the Matlab prompt and then to follow the self-contained messages through the forthcoming windows. The first couple of windows (Figure 1) help the user in choosing the type of computation (i.e. characteristic roots or stability chart of a given system) and in inserting the preliminary data such as m, k and k_d as given in (1). In particular, the menu "LOAD" in the second window allows to upload existing data saved in previous sessions.

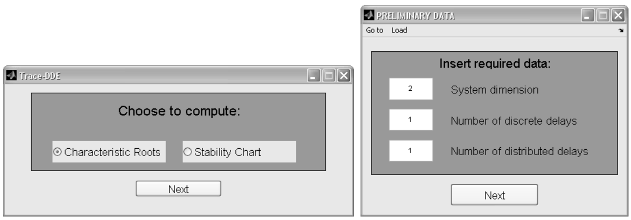

Fig. 1. Starting windows: problem choice (left) and system data (right).

Supposing to start with the roots computation (Section 3), the window in Figure 2, left, appears and the user is asked to fill the boxes with the required data by using the standard Matlab syntax for matrices (this is the only Matlab requirement). The notation follows the representation of the chosen DDE appearing at the bottom. The maximum number of roots $m(N+1)$ (with N the discretization parameter, see Section 3) is determined automatically since $N = 20$ is set by default. Obviously, the users requiring more roots or more accuracy may change this value in the source code. Once the computation is performed, the result window (Figure 2, right) is produced and several options (such as, print, plot, etc.) are available.

In order to switch to the stability chart computation (Section 4), one can run `tracedde` again or go directly to the appropriate window (Figure 3, left) by selecting the relevant choice in the "GO TO" menu in Figure 2, left. With this second choice all data appear automatically and the user is required only to choose the uncertain parameters using variables (a and b for (4)) and insert their ranges in the allowed boxes. Then the option to plot the stability boundaries (Figure 3, right, observe the triangulation structure along the boundary) or surface is provided.

We are not going to give here a full description of all the features provided in TRACE-DDE. The motivation is twofold: on the one hand, the GUI structure should be self-contained, on the other hand we want to focus on the general purpose of the package. Nevertheless, we would like to mention, among others, the presence of error-alerting messages, data saving and loading facilities and zooming of stability chart zones. Other details can be found in the relevant web site. Anyway, let us invite possible users to report their feedbacks to the authors.

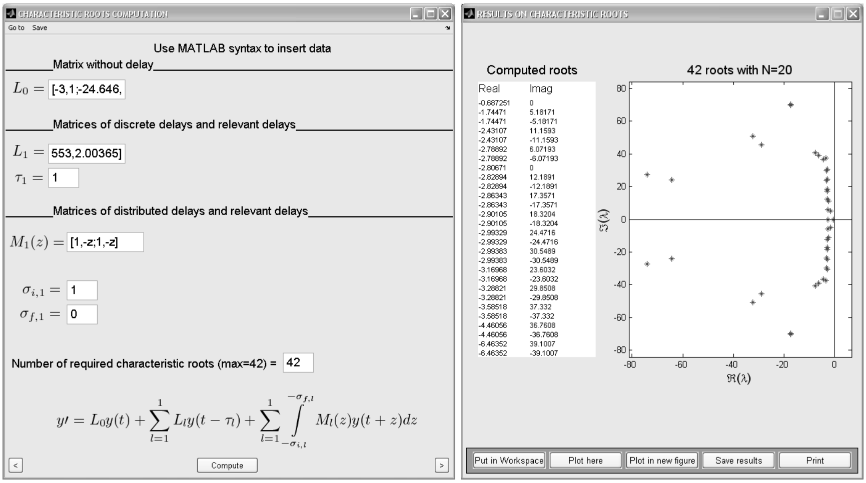

Fig. 2. Characteristic roots windows: data (left) and results (right).

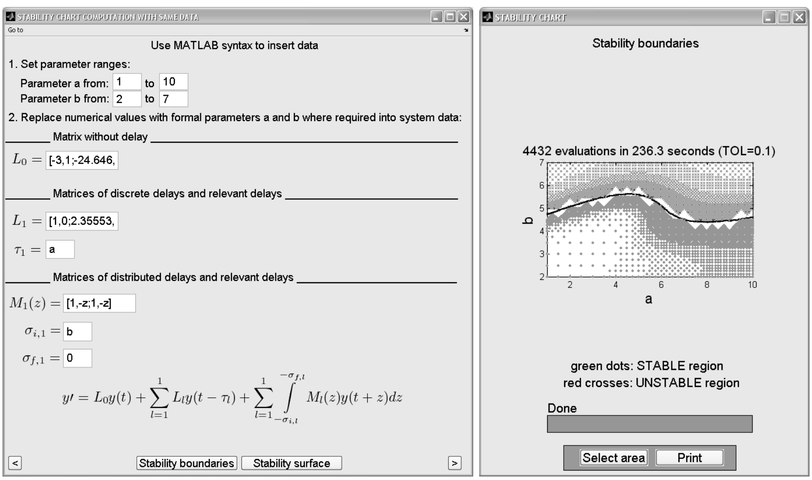

Fig. 3. Stability charts windows: data (left) and results (right).

3 The pseudospectral approach

For the sake of simplicity, let us rewrite the system of DDEs (1) as $y'(t) = f(y_t)$ where $f : X \to \mathbb{C}^m$ is defined by

$$f(\varphi) = \sum_{l=0}^{k} L_l \varphi(-\tau_l) + \int_{-\tau}^{0} M(\theta)\varphi(\theta)d\theta, \ \varphi \in X. \tag{5}$$

For a given N, N positive integer, let us consider the mesh

$$\Omega_N = \{\theta_{N,i},\ i = 0, 1, \ldots, N\}$$

of $N+1$ distinct nodes in $[-\tau, 0]$ with $0 = \theta_{N,0} > \theta_{N,1} > \ldots > \theta_{N,N} \geq -\tau$. We replace the continuous state space X by the space X_N of the discrete functions defined on the mesh Ω_N, i.e. any $\varphi \in X$ is discretized into the block-vector $x \in X_N$ of components $x_i = \varphi(\theta_{N,i}) \in \mathbb{C}^m$, $i = 0, 1, \ldots, N$.

Let now $\mathcal{L}_N x$, $x \in X_N$, be the unique $\mathbb{C}^m$-valued interpolating polynomial of degree $\leq N$ with $(\mathcal{L}_N x)(\theta_{N,i}) = x_i$, $i = 0, 1, \ldots, N$. Thus we approximate the infinitesimal generator $\mathcal{A}$ by the matrix $\mathcal{A}_N : X_N \to X_N$, called spectral differentiation matrix, defined as follows:

$$\begin{cases} (\mathcal{A}_N x)_0 = f_N(\mathcal{L}_N x), \\ (\mathcal{A}_N x)_i = (\mathcal{L}_N x)'(\theta_{N,i}),\ i = 1, \ldots, N, \end{cases} \tag{6}$$

where f_N is an approximation of f in which the distributed delay integral term in (5) is substituted by a suitable interpolatory quadrature rule or f_N is equal to f in the case of simple matrix function M for which the integral can be exactly computed. In particular, the above discretization follows from the discretization of the so-called *splicing condition* on $\varphi'(0)$ in (3) for what concerns the first block-row, while the remaining part is due to the derivative action (2) of the infinitesimal generator.

By using the Lagrange representation of $\mathcal{L}_N x$, i.e.

$$(\mathcal{L}_N x)(\theta) = \sum_{j=0}^{N} \ell_{N,j}(\theta) x_j,\ \theta \in [-\tau, 0],$$

where

$$\ell_{N,j}(\theta) = \sum_{\substack{i=0 \\ i \neq j}}^{N} \frac{\theta - \theta_{N,i}}{\theta_{N,j} - \theta_{N,i}},\ \theta \in [-\tau, 0],$$

are the Lagrange coefficients relevant to the nodes in Ω_N, we obtain

$$\mathcal{A}_N = \begin{pmatrix} a_0 & a_1 & \cdots & a_N \\ d_{10} & d_{11} & \cdots & d_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N0} & d_{N1} & \cdots & d_{NN} \end{pmatrix} \in \mathbb{C}^{m(N+1) \times m(N+1)}$$

with

$$\begin{cases} a_j = f_N\left(\ell_{N,j}(\cdot) I_m\right),\ j = 0, 1, \ldots, N, \\ d_{ij} = \ell'_{N,j}(\theta_{N,i}) I_m,\ i = 1, \ldots, N,\ j = 0, 1, \ldots, N, \end{cases}$$

where, by definition, $f_N\left(\ell_{N,j}(\cdot) I_m\right) = \left(f_N\left(\ell_{N,j}(\cdot) e^{(i)}\right)\right)_{i=1}^{m}$ with $e^{(i)}$'s the canonical vectors in $\mathbb{R}^m$. Explicit expressions of the d_{ij}'s for particular choices of Ω_N (e.g. Chebyshev extremal points) can be found in [24, §6].

Therefore the original infinite-dimensional problem of the numerical computation of the characteristic roots of (1) can be turned into the finite-dimensional eigenvalue problem for the matrix $\mathcal{A}_N$, i.e. eigenvalues of $\mathcal{A}_N$ directly approximate the characteristic roots. If one wonders how much accurate are these approximations, the following theorem, whose proof is detailed in [8] through a complete convergence analysis, should serve as an answer.

Theorem 1. *Assume to use the Chebyshev extremal nodes, i.e.*

$$\theta_{N,i} = \frac{\tau}{2}\left(\cos\left(i\frac{\pi}{N}\right) - 1\right),\ i = 0, 1, \ldots, N,$$

and that the function f_N in (6) satisfies $\sup_{N\in\mathbb{N}} \|f_N\| < +\infty$. Let $\lambda^* \in \mathbb{C}$ be a characteristic root of (1) with multiplicity ν. Then, for sufficiently large N, $\mathcal{A}_N$ has exactly ν eigenvalues λ_i, $i = 1,\dots,\nu$ (counted with their multiplicity), such that

$$\max_{i=1,\dots,\nu} |\lambda^* - \lambda_i| \le \left[C_2 \left(\varepsilon_N + \frac{1}{\sqrt{N}} \left(\frac{C_1}{N} \right)^N \right) \right]^{1/\nu} \tag{7}$$

holds where C_1 and C_2 are constants independent of N and ε_N is the error due to the approximate computation of the distributed term.

It is important to point out that the spectral accuracy (i.e. $O(N^{-N})$) of the method is preserved in the behavior of the final error (7), if the distributed term in (6) can be exactly computed, i.e. $\varepsilon_N = 0$. On the other hand, we also maintain the spectral accuracy if we make the further assumptions that the function M in (1) is C^∞ with equilimited derivatives and that a interpolatory quadrature rule based on Chebyshev nodes as given in Theorem 1 is used for the distributed term (Clenshaw-Curtis quadrature). Finally, we observe that if the distributed term is approximated by some software within a tolerance TOL, then the error bound in Theorem 1 is still valid with $\varepsilon_N = \mathrm{TOL}$ and thus the convergence follows spectral accuracy down to this value.

As to conclude the section, we give an explicit expression of the discretization matrix for the single constant delay scalar equation

$$y'(t) = ay(t) + by(t-\tau),\ a, b \in \mathbb{C}.$$

In this case, it is easy to see that $\mathcal{A}_N \in \mathbb{C}^{(N+1)\times(N+1)}$ is given by

$$\mathcal{A}_N = \begin{pmatrix} a & 0 & \cdots & 0 & b \\ \ell'_{N,0}(\theta_{N,1}) & \ell'_{N,1}(\theta_{N,1}) & \cdots & \ell'_{N,N-1}(\theta_{N,1}) & \ell'_{N,N}(\theta_{N,1}) \\ \vdots & \vdots & \ddots & \vdots & \\ \ell'_{N,0}(\theta_{N,N}) & \ell'_{N,1}(\theta_{N,N}) & \cdots & \ell'_{N,N-1}(\theta_{N,N}) & \ell'_{N,N}(\theta_{N,N}) \end{pmatrix}$$

whenever $\theta_{N,N} = -\tau$.

4 Detection of stability boundaries

In this section we briefly describe an adaptive strategy which can be employed in order to detect the stability boundaries of a system of DDEs in the plane of two uncertain parameters. The problem can be viewed as a particular instance of the more general question of computing the level curves of a surface function $z = f(x, y)$ where, possibly, f does not have an explicit form but rather it can be evaluated for any choice of x and y in a given rectangular region of the (x, y)-plane. In fact, the complete stability map in the parameters plane is the set of level curves $f(x, y) = 0$ where f is the function giving the real part of the rightmost eigenvalue governing the system dynamics. Hence, in our case, f corresponds to an eigenvalue problem, possibly of large dimension (e.g. space-discretized partial differential equations), and its computation at one point (x, y) could be heavy.

Here we describe only the underlying idea of the algorithm while we refer the interested reader to [6] and [7] for a fully detailed version of the methodology which is implemented in TRACE-DDE.

A first rude (but simple) technique is used in the Matlab's `contour` plot. It is based on a uniform grid dividing the plane region into rectangular cells at whose vertices f is evaluated. If the signs at the vertices of a cell are not equal, a segment of level curve crosses the cell. Crossing points are approximated by linear interpolation and joint together they reproduce the searched boundary. The final accuracy depends only on the grid size. This tool inevitably calculates many values of f which are useless and in order to reduce the amount of computation an adaptive refinement is proposed.

In alternative to the obvious rectangular refinement (each cell is divided into four rectangles by the mid points), we consider a triangular refinement by which each cell is divided at the first step into four triangles by the cell center and any of these latter into two triangles in all the forthcoming steps by the height foot. Triangulation is more efficient in reducing the overall number of evaluation points. In fact, every triangular refinement requires one new evaluation (the cell center or the height foot) while the rectangular one requires five new evaluations (the cell center plus the mid point of each edge). Moreover it can be accompanied with a couple of other tricks (described in [7]) to further reduce the computational cost while preserving the desired accuracy.

5 A test set

In this section we collect a set of results which we believe it might be useful for comparison purposes. Analytical knowledge of exact characteristic roots and stability boundaries is, in fact, restricted to rather exceptional cases. To this aim we cite [15]: "*For the autonomous matrix equation...*" $x'(t) = Ax(t) + Bx(t-r)$ "*...the exact region of stability as an explicit function of A, B and r is not known and probably will never be known. The reason is simple to understand because the characteristic equation is so complicated. It is therefore worthwhile to obtain methods for determining approximations to the region of stability.*" Hence we report in the sequel a list of case-studies with relevant plots of the rightmost part of their (approximated) spectrum, a stability chart and the value of the righmost root accurate to machine precision. All results are obtained by using the core algorithms in TRACE-DDE and concern the following systems and equations:

Example 1. [13]

$$y'(t) = ay(t) + by(t-1).$$

Example 2. [20]

$$y'(t) = \begin{pmatrix} -6.45 & -12.1 \\ 1.5 & -0.45 \end{pmatrix} y(t) + \begin{pmatrix} -6 & 0 \\ 1 & 0 \end{pmatrix} y(t-\tau_1) + \begin{pmatrix} 0 & 4 \\ 0 & -2 \end{pmatrix} y(t-\tau_2)$$

Example 3. [14]

$$y'(t) = L_0 y(t) + L_1 y(t-1) + \int_{-\tau_1}^{-0.1} M_1 y(t+\theta)d\theta + \int_{-\tau_2}^{-0.5} M_2 y(t+\theta)d\theta$$

with coefficients matrices

$$L_0 = \begin{pmatrix} -3 & 1 \\ -24.646 & -35.430 \end{pmatrix}, L_1 = \begin{pmatrix} 1 & 0 \\ 2.35553 & 2.00365 \end{pmatrix},$$

$$M_1 = \begin{pmatrix} 2 & 2.5 \\ 0 & -0.5 \end{pmatrix}, M_2 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Example 4. [1]

$$y'(t) = L_0 y(t) + L_1(y(t-\tau_1) + y(t-\tau_2)) + \\ + L_2(y(t-2\tau_1) + y(t-2\tau_2)) + L_3 y(t-\tau_1-\tau_2)$$

whose coefficients matrices $L_l \in \mathbb{C}^{8\times 8}$, $l = 0, 1, 2, 3$, can be found in [1].

For every example we report in Table 1 the parameters nominal values, the corresponding rightmost root (or imaginary couple) λ_r, the tolerance at which this latter is approximated (which determines the number of significant digits) and the discretization index N at which such an accuracy is met first. Almost full machine precision is performed for Examples 2 and 3 with very low N. As for Example 1, the accuracy is halved since $\lambda_r = 1$ is a double root as it can be easily verified through the characteristic equation (as stated in Theorem 1). As for the last Example 4, the final accuracy level is rather low due to ill-conditioning of the coefficients matrices: this may serve as an interesting case-study to test a root-finder performances on dramatically ill-posed problems.

Table 1. Nominal values and righmost roots.

example	nominal values	N	λ_r	TOL
1	$a = 2, b = e$	10	0.9999999 ± 0.0000002i	10^{-7}
2	$\tau_1 = 0.2, \tau_2 = 0.3$	14	$-1.3352684760607 \pm 9.1194833474024$i	10^{-13}
3	$\tau_1 = 0.3, \tau_2 = 1$	13	-1.24623812459204	10^{-14}
4	$\tau_1 = \tau_2 = 5 \times 10^{-4}$	17	-22.616 ± 3617.030i	10^{-3}

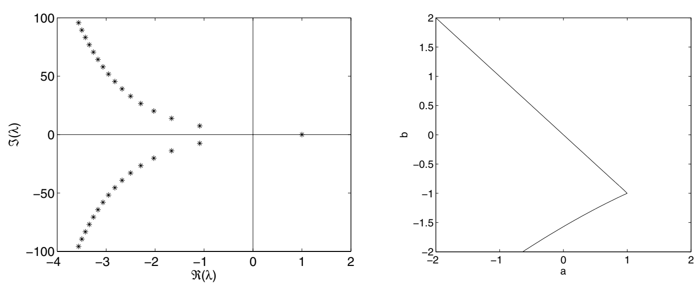

Fig. 4. Spectrum and stability chart for Example 1.

6 Acknowledgments

The authors are grateful to Daniele Sechi who built the first GUI version of TRACE-DDE for Macintosh platforms as part of his B.Sc. thesis in computer science at University of Udine [22].

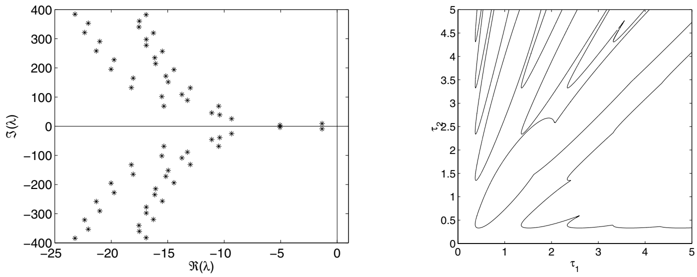

Fig. 5. Spectrum and stability chart for Example 2.

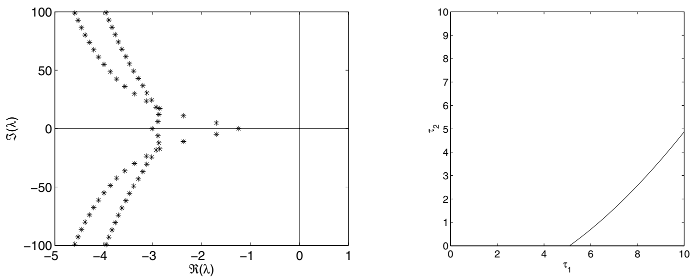

Fig. 6. Spectrum and stability chart for Example 3.

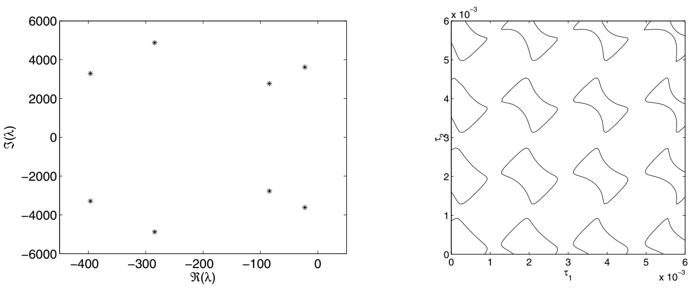

Fig. 7. Spectrum and stability chart for Example 4.

References

1. Altintas, Y., Engin, S., Budak, E.: Analytical stability prediction and design of variable pitch cutters. ASME J. Manuf. Sci. E.-T. 121, 173–178 (1999)
2. Breda, D.: The infinitesimal generator approach for the computation of characteristic roots for delay differential equations using BDF methods. Technical Report RR17/2002, Department of Mathematics and Computer Science, University of Udine (2002)
3. Breda, D.: Solution operator approximation for characteristic roots of delay differential equations. Appl. Numer. Math. 56(3-4), 305–317 (2006)

4. Breda, D., Iannelli, M., Maset, S., Vermiglio, R.: Stability analysis of the Gurtin-MacCamy model. J. Numer. Anal. 46(2), 980–995 (2008)
5. Breda, D., Maset, S., Vermiglio, R.: Computing the characteristic roots for delay differential equations. IMA J. Numer. Anal. 24(1), 1–19 (2004)
6. Breda, D., Maset, S., Vermiglio, R.: Efficient computation of stability charts for linear time delay systems. In: Proc. ASME-IDETC/CIE 2005, Long Beach, USA (2005)
7. Breda, D., Maset, S., Vermiglio, R.: An adaptive algorithm for efficient computation of level curves of surfaces. To appear in Numer Algorithms (2009) (preprint)
8. Breda, D., Maset, S., Vermiglio, R.: Pseudospectral differencing methods for characteristic roots of delay differential equations. SIAM J. Sci. Comput. 27(2), 482–495 (2005)
9. Breda, D., Maset, S., Vermiglio, R.: Pseudospectral approximation of eigenvalues of derivative operators with non-local boundary conditions. Appl. Numer. Math. 56(3-4), 318–331 (2006)
10. Breda, D., Maset, S., Vermiglio, R.: Numerical approximation of characteristic values of partial retarded functional differential equations. Numer. Math. (2009), doi:10.1007/s00211-009-0233-7
11. Diekmann, O., van Gils, S.A., Verduyn Lunel, S.M., Walther, H.O.: Delay Equations - Functional, Complex and Nonlinear Analysis. In: AMS series, vol. 110. Springer, New York (1995)
12. Engelborghs, K., Luzyanina, T., Roose, D.: Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL. ACM T. Math. Software 28(1), 1–21 (2002)
13. Engelborghs, K., Roose, D.: On stability of LMS methods and characteristic roots of delay differential equations. SIAM J. Numer. Anal. 40(2), 629–650 (2002)
14. Fattouh, A., Sename, O., Dion, J.M.: h_∞ controller and observer design for linear systems with point and distributed delays. In: Proc. 2nd IFAC workshop on Linear Time Delay Systems, Ancona, Italy (2000)
15. Hale, J.K., Verduyn Lunel, S.M.: Introduction to functional differential equations. AMS series, vol. 99. Springer, New York (1993)
16. Insperger, T., Stépán, G.: Updated semi-discretization method for periodic delay-differential equations with discrete delay. Int. J. Numer. Meth. Eng. 61, 117–141 (2004)
17. Luzyanina, T., Engelborghs, K., Roose, D.: Computing stability of differential equations with bounded distributed delays. Numer. Algorithms 34(1), 41–66 (2003)
18. Michiels, W., Niculescu, S.I.: Stability and stabilization of time-delay systems. An eigenvalue based approach. In: Advances in Design and Control, vol. 12. SIAM, Philadelphia (2007)
19. Niculescu, S.-I.: Delay Effects on Stability: A Robust Control Approach. In: TLNCIS Monograph, vol. 269. Springer, London (2001)
20. Olgac, N., Sipahi, R.: An exact method for the stability analysis of time delayed LTI systems. IEEE T. Automat. Contr. 47(5), 793–797 (2002)
21. Richard, J.-P.: Time-delay systems: an overview of some recent advances and open problems. Automatica 39, 1667–1694 (2003)
22. Sechi, D.: Sviluppo di interfaccia grafica per lo studio della stabilità di sistemi differenziali con ritardo (in italian). Master's thesis, University of Udine (2005)
23. Stépán, G.: Retarded dynamical systems. Longman, Harlow (1989)
24. Trefethen, L.N.: Spectral methods in MATLAB. Software - Environment - Tools series. SIAM, Philadelphia (2000)
25. Vyhlídal, T., Zítek, P.: Mapping the spectrum of a retarded time-delay system utilizing root distribution features. In: Proc. 6th IFAC workshop on Linear Time Delay Systems, L'Aquila, Italy (2006)