

**Numerical computation of stability and  
detection of Hopf bifurcations of steady  
state solutions of delay differential  
equations**

*Koen Engelborghs*

*Dirk Roose*

*Report TW 274, January 1998*



**Katholieke Universiteit Leuven**  
**Department of Computer Science**  
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

# Numerical computation of stability and detection of Hopf bifurcations of steady state solutions of delay differential equations

*Koen Engelborghs*

*Dirk Roose*

*Report TW 274, January 1998*

Department of Computer Science, K.U.Leuven

## **Abstract**

The characteristic equation of a system of delay differential equations (DDEs) is a nonlinear equation with infinitely many zeros. The stability of a steady state solution of such a DDE system is determined by the number of zeros of this equation with positive real part. We present a numerical algorithm to compute the rightmost, i.e. stability determining, zeros of the characteristic equation. The algorithm is based on the application of subspace iteration on the time integration operator of the system or its variational equations. The computed zeros provide insight in the systems behaviour, can be used for robust bifurcation detection and for efficient indirect calculation of bifurcation points.

**Keywords :** Delay differential equations, steady state solutions, stability.

**AMS(MOS) Classification :** Primary : 34K20, Secondary : 65J10.

## 1 Introduction

This paper is concerned with systems of delay differential equations (DDEs),

$$\dot{x} = f(x(t), x(t - \tau_1), \dots, x(t - \tau_m)), \quad (1)$$

with one or several fixed, discrete delays  $\tau_j \in \mathbb{R}_+$ ,  $j = 1 \dots m$ ,

$$x \in \mathbb{R}^n, f : \mathbb{R}^n \times \mathbb{R}^n \times \dots \times \mathbb{R}^n \rightarrow \mathbb{R}^n.$$

Note that we do not demand commensurability of the delays (which would, for instance, prove impractical when one wants to use one of several delays as a continuation parameter (see section 4.2)).

Stability analysis of steady state solutions of (1) is the subject of some recent papers [4, 1, 2]. Most of these studies consist of analytical determination of bifurcation curves (especially Hopf bifurcations) in a two-dimensional parameter space. The stability of a given equilibrium is then determined from its location with respect to these curves. The problem with this approach is that it is restricted to systems where the right hand side and its derivatives are known analytically. Moreover, the computational costs become unfeasible when the system has more than just a few dimensions.

In [13] an algorithm is described to compute the number of zeros of the characteristic equation in the right half plane. The method presented here is an extension of this, in the sense that it computes the rightmost zeros of the characteristic equation, instead of just their number. These computed zeros provide insight in the systems behaviour, can be used for robust detection and efficient calculation of bifurcations points.

## 2 Stability of steady state solutions of DDEs

The DDE system (1) has the same steady state solutions as the system of ordinary differential equations that is formed from (1) by putting all its delays to zero. An equilibrium  $x \in \mathbb{R}^n$  of (1) can therefore be computed as a zero of the  $n$ -dimensional nonlinear system

$$f(x, x, \dots, x) = 0.$$

Once a steady state solution  $x^*$  is found, its stability is determined by the zeros of the characteristic equation. If we define the  $n \times n$ -dimensional characteristic matrix  $\Delta$  as:

$$\Delta(x^*, \lambda) \triangleq \lambda I - A_0(x^*) - \sum_{j=1}^m A_j(x^*) e^{-\tau_j \lambda},$$

with  $I$  the  $n$  by  $n$  identity matrix,

$$A_0(x^*) \triangleq \frac{\partial f}{\partial x(t)}(x^*, x^*, \dots, x^*) \in \mathbb{R}^{n \times n} \quad (2)$$

and

$$A_j(x^*) \triangleq \frac{\partial f}{\partial x(t - \tau_j)}(x^*, x^*, \dots, x^*) \in \mathbb{R}^{n \times n}, \quad j = 1 \dots m, \quad (3)$$

then the characteristic equation reads

$$\det(\Delta(x^*, \lambda)) = 0. \quad (4)$$

The steady state solution  $x^*$  will be asymptotically stable if all the zeros  $\lambda_k$  of (4) have strict positive real part [6, p. 23]. Remark that, in general, (4) will have infinitely many zeros.

An alternative to system (4) is the  $n + 1$ -dimensional system

$$\begin{cases} \Delta(x^*, \lambda)v = 0 \\ c^T v - 1 = 0 \end{cases}. \quad (5)$$

At a zero  $\lambda_k$  of (4),  $\Delta(x^*, \lambda_k)$  is singular and  $(\lambda_k, v_k)$  is a solution of (5) with  $v_k$  the eigenvector, normalized by  $c^T v_k = 1$ . Note that  $c$  should not be orthogonal to  $v_k$ . Similar approaches have been used to construct determining systems for turning points and other branching points, see e.g. [15].

For complex  $\lambda = a + bi \in \mathbb{C}$ , we can avoid complex arithmetic using:

$$\begin{cases} \Re(\Delta(x^*, a + bi))v - \Im(\Delta(x^*, a + bi))w = 0 \\ \Im(\Delta(x^*, a + bi))v + \Re(\Delta(x^*, a + bi))w = 0 \\ c^T v - 1 = 0 \\ c^T w = 0 \end{cases} \quad (6)$$

The last two equations are normalisations of the real, respectively, the imaginary part of the null-vector. The vector  $c$  should not be orthogonal to the span of  $v$  and  $w$  at the solution point.

The extended equations have the same solutions for  $\lambda$  as (4). Double real respectively complex roots of (4) will become two single roots of (5) respectively (6) if there are two independent eigenvectors.

### 3 The algorithm

One could seek roots of the characteristic equation using Newton's method. However even if a grid of starting values in a certain region is taken there is no guarantee that all zeros with a desired property will be found. For our purposes an iteration scheme that converges automatically to the rightmost roots is desired.

### 3.1 Outline of ideas

A straightforward method to check the asymptotic stability of a given equilibrium solution of (1) would be to start close to it and see whether time integration either converges or diverges. To avoid missing the possible small domain of attraction of an attracting fixed point one can work with the linearization of the system around the equilibrium because it enlarges the attraction domain of the fixed point to the whole space. This intuitive stability test based on time integration must, in some way, be related to the stability theory outlined in section 2. The correspondence can be seen in the following way.

The linearization of (1) around  $x^*$  is a linear  $n$ -dimensional DDE, called the variational equation, which reads:

$$\dot{y} = A_0(x^*)y(t) + \sum_{j=1}^m A_j(x^*)y(t - \tau_j). \quad (7)$$

When integrating DDEs one needs to specify an initial function segment instead of an initial vector. Let

$$\tau_{\max} = \max\{\tau_j, j = 1 \dots m\},$$

then a solution  $y(t)$  of (7) is uniquely defined by an initial function segment  $\varphi \in C([-\tau_{\max}, 0], \mathbb{R}^n)$ . We repeat a theorem from [5, p. 34]:

**Theorem 3.1** *Let  $y(t)$  be a solution of the linear DDE (7) corresponding to some initial function segment  $\varphi(\theta)$ ,  $\theta \in [-\tau_{\max}, 0]$ . For any  $\gamma \in \mathbb{R}$  such that  $\det(\Delta(x^*, \lambda)) \neq 0$  on the line  $\Re(\lambda) = \gamma$ , we have the asymptotic expansion:*

$$y(t) = \sum_{k=1}^l p_k(t)e^{\lambda_k t} + o(e^{\gamma t}) \text{ for } t \rightarrow +\infty \quad (8)$$

where  $\lambda_1, \dots, \lambda_l$  are the finitely many zeros of  $\det(\Delta(x^*, \lambda))$  with real part exceeding  $\gamma$  and where  $p_k(t)$  is a  $\mathbb{C}^n$ -valued polynomial in  $t$  of degree less than or equal to  $m_k - 1$  with  $m_k$  the multiplicity of  $\lambda_k$  as a zero of  $\det(\Delta(x^*, \lambda))$ .

For  $\gamma < 0$ , an immediate consequence is the stability theory given in section 2. For the sake of the argument we assume all roots  $\lambda_k$ ,  $k = 1 \dots l$  to have multiplicity  $m_k = 1$ , the polynomials  $p_k(t)$  are then just constants,  $p_k(t) \equiv c_k$ . The effect of time integration over a fixed length of time  $t_0$  can be seen from:

$$y(t_0 + t) = \sum_{k=1}^l \hat{c}_k e^{\lambda_k t} + o(e^{\gamma(t_0+t)}) \text{ for } t \rightarrow +\infty, \text{ with } \hat{c}_k = c_k e^{\lambda_k t_0} \quad (9)$$

Assuming  $o(e^{\gamma(t+t_0)})$  is negligible (because either  $(t+t_0)$  or  $l$  is large enough), we see that the coefficient  $p_k(t) = c_k$  of expansion (8) has been multiplied with  $\sigma_k = e^{\lambda_k t_0}$ . The exponential transformation  $\sigma = e^{\lambda t_0}$  maps eigenvalues  $\lambda_k$  with large positive real part to  $\sigma_k$  having large modulus and eigenvalues  $\lambda_k$  with large negative real part to  $\sigma_k$  having small modulus. Hence the terms of (9) corresponding to the rightmost zeros  $\lambda_k$  of the characteristic equation have become relatively more important. We could say that time integration of (7) automatically evolves towards the rightmost modes of the system. A conclusion which is in some sense trivial because, as seen in expansion (8) they correspond to the fastest growing or least decaying modes of (7).

If we can extract  $\lambda_1$  and perhaps also  $c_1$  from the asymptotic trajectory  $c_1 e^{\lambda_1 t}$ , we have found a method to automatically calculate the rightmost mode of (1). To find more than one rightmost mode we could integrate a set of functions and periodically keep them independent to prevent convergence to the same, rightmost mode.

This method can be seen as an infinite-dimensional form of subspace iteration, a numerical technique used to compute the largest (in modulus) eigenvalues of a matrix. In the next section we will formalize this equivalence and reformulate the traditional algorithm in our setting.

### 3.2 Subspace iteration

Let  $y(t; \phi)$  be the solution of (7) corresponding to the initial condition  $\phi \in C([- \tau_{\max}, 0], \mathbb{R}^n)$ . We define the function segment  $y_t(\phi) \in C([- \tau_{\max}, 0], \mathbb{R}^n)$  as follows:

$$y_t(\phi)(\theta) = y(t + \theta; \phi), \quad \theta \in [- \tau_{\max}, 0].$$

Let  $S(x^*, t_0)$  be the linear operator which transforms  $\phi$  to the function segment  $y_{t_0}(\phi)$ :

$$S(x^*, t_0)\phi \triangleq y_{t_0}(\phi). \quad (10)$$

We state the following result:

**Theorem 3.2** *If  $(\lambda \in \mathbb{C}, v \in \mathbb{C}^n)$  is a simple root of (5) then  $\phi = ve^{\lambda\theta}$ ,  $\theta \in [- \tau_{\max}, 0]$  is an eigenfunction of  $S(x^*, t_0)$  corresponding to a simple eigenvalue  $e^{\lambda t_0}$ .*

*Proof:* Because the characteristic equation in the form (5) can be obtained from (7) by filling in the sample solution  $y(t) = ve^{\lambda t}$ , the fact that  $(\lambda \in \mathbb{C}, v \in \mathbb{C}^n)$  is a root of (5) means that  $y(t)$  is a solution of (7). This implies that for  $\theta \in [- \tau_{\max}, 0]$ ,

$$(S(x^*, t_0)y_t(\phi))(\theta) = y_{(t+t_0)}(\phi)(\theta) = ve^{\lambda t_0}e^{\lambda(t+\theta)} = (e^{\lambda t_0}y_t(\phi))(\theta), \quad (11)$$

with  $\phi$  the appropriate initial condition,

$$\phi(\theta) = ve^{\lambda\theta}, \theta \in [-\tau_{\max}, 0].$$

Equation (11) states that  $y_t(\phi)(\theta) = ve^{\lambda\theta}$  is an eigenfunction of  $S(x^*, t_0)$  corresponding to an eigenvalue  $e^{\lambda t_0}$ .

An equivalent theorem with proof for a more general class of equations and including the case of multiple roots can be found in [5, p. 126].

When an eigenvalue  $\sigma \in \mathbb{C}$  of  $S(x^*, t_0)$  is given, the real part of  $\lambda$  can easily be found provided  $t_0 \neq 0$ ,

$$\Re(\lambda) = \frac{\ln(\|\sigma\|)}{t_0}. \quad (12)$$

The imaginary part of  $\lambda$  is found modulo  $\pi/t_0$ , using

$$\Im(\lambda) \equiv \frac{\arccos\left(\frac{\Re(\sigma)}{\|\sigma\|}\right)}{t_0} \pmod{\frac{\pi}{t_0}}, \quad (13)$$

or,

$$\Im(\lambda) \equiv \frac{\arcsin\left(\frac{\Im(\sigma)}{\|\sigma\|}\right)}{t_0} \pmod{\frac{\pi}{t_0}}. \quad (14)$$

However, for the purposes of stability analysis and detection of bifurcations, the real part of  $\lambda$  suffices. If it is of interest, the full imaginary part can be extracted from the calculated eigenfunction (see section 3.3).

The dominant eigenvalues of  $S(x^*, t_0)$  will, due to transformation (12), correspond to the rightmost solutions of the characteristic equation. Subspace iteration [17] is a well known and well understood method to compute the largest eigenvalues of a matrix. We rewrite the basic subspace iteration algorithm with projection [17, p. 156] in our infinite-dimensional setting and notations.

### Algorithm 3.1

**Start:**

*Choose an initial set of functions:*

$$Y = \{y_1, \dots, y_l\}, y_i \in C([-\tau_{\max}, 0], \mathbb{R}^n).$$

*Orthonormalize the functions  $y_i$ ,  $i = 1 \dots l$ , using the inner product,*

$$(y_i, y_j) \triangleq \int_{-\tau_{\max}}^0 y_i(\theta)y_j(\theta)d\theta. \quad (15)$$

Choose  $t_0 > 0$ .

**Iterate:**

Until convergence do:

(a) Integrate functions  $y_i$  using (7),  $i = 1 \dots l$ :

$$Z = \{S(x^*, t_0)y_1, \dots, S(x^*, t_0)y_l\}.$$

(c) Compute  $B \in \mathbb{R}^{l \times l}$  (using the inner product (15)):

$$B_{i,j} = (y_i, z_j).$$

(d) Use the QR algorithm to compute eigenvectors  $\sigma_i$  and the Schur vectors  $u_i$  of  $B$ ,  $i = 1 \dots l$ :

$$U = [u_1, \dots, u_l] \in \mathbb{R}^{l \times l}; \quad \sigma_i \in \mathbb{C}, \quad i = 1 \dots l.$$

(e) Recombine the functions  $z_i$  into  $Y^{\text{new}} = \{y_1^{\text{new}}, \dots, y_l^{\text{new}}\}$ :

$$y_i^{\text{new}} = \sum_{j=1}^l u_{i,j} z_j, \quad i = 1 \dots l.$$

(b) Orthormalize the functions  $y_i^{\text{new}}$ ,  $i = 1 \dots l$ , using the inner product (15).

(f) Test for convergence. If not converged, repeat with  $Y := Y^{\text{new}}$ .

At convergence the eigenvalues  $\sigma_i$ ,  $i = 1 \dots l - 1$  of  $B$  will be good approximations of the  $l - 1$  most dominant eigenvalues of  $S(x^*, t_0)$ . Eigenvalue  $\sigma_l$  may not converge if it forms a complex pair of eigenvalues with  $\sigma_{l+1}$ . Because the convergence speed of the  $\sigma_i$  depends on its distance to  $\sigma_{l+1}$  one usually takes  $l$  a little larger than necessary to obtain good convergence of the eigenvalues that are of actual interest.

The method thus obtained is automatic and robust but expensive. Several time integrations are necessary before convergence is achieved. Some ways of improving the efficiency (and at the same time the accuracy of the results) are explained in the next section. We end this section with two remarks.

Although in actual computations the problem necessarily has to be discretized we prefer an infinite dimensional reformulation of the subspace iteration. Indeed, the advantage is that we do not need a global discretisation, instead we want the representation of the function segments to be able to change during the course of the computations. This is important because as more and more modes are calculated the imaginary parts tend to increase

drastically. When these modes are still of interest, adaptive time integration of the functions  $y_i$  decreases its steplength and the corresponding discretisation of the functions  $y_i$  should refine in accordance with the integration code.

If the derivatives of the right hand side (2), (3) cannot easily be computed one can avoid their use by an approximation of the action of  $S(x^*, t_0)$ , through time integration of the original system (similar to [12, section 3.2]):

$$(S(x^*, t_0)\phi)(\theta) \approx \frac{x_{t_0}(\Phi)(\theta) - x^*}{\epsilon}, \quad \theta \in [-\tau_{\max}, 0], \quad (16)$$

with  $\Phi(\theta) = x^* + \epsilon\phi(\theta)$ ,  $\theta \in [-\tau_{\max}, 0]$  and with  $x_{t_0}(\Phi)$  obtained from  $\Phi$  by integration of (1) over time  $t_0$ . Although this can be useful in some cases, one must be very careful because (16) can produce breakdown of convergence and, which is probably worse, the computation of faulty spurious modes.

### 3.3 Improving efficiency and accuracy

A common practice to improve the efficiency of algorithm 3.1 is the use of a technique called *locking* [17, section 3.1]. Because of the different convergence speeds of the different modes, the first modes will converge earlier than less dominant modes. The corresponding eigenfunction can be kept fixed because its behaviour under time integration is known. Only in the orthogonalization (step (b) of algorithm 3.1) we still have to orthogonalize with respect to the 'locked' modes.

The implementation of locking is usually done using the last iterate of the eigenfunction  $y_i$  together with its integration  $z_i$ . However in our situation we can do better because the form of the eigenfunctions is known analytically. If we have the corresponding zero  $(\lambda, v)$  of characteristic equation (5) we can construct the corresponding eigenfunction  $ve^{\lambda\theta}$ .

After deriving an approximation to  $(\lambda^{(0)}, v^{(0)})$  from  $y_i$  we can easily use Newton's method on the characteristic equation to obtain the pair  $(\lambda, v)$  with full accuracy. Indeed, while subspace iteration is robust and automatically converges to the modes of interest, Newton's iteration is much cheaper and has asymptotic quadratic convergence once we have a good enough approximation of a mode.

To extract the information  $(\lambda = a + bi, v)$  from  $(\sigma_i, y_i)$  we use formula (12) for the real part  $a$  and  $v$  can be obtained from

$$v = ve^{\lambda\theta} \Big|_{\theta=0} = y_i(0).$$

The imaginary part  $b$  can be obtained from a curve fitting procedure on  $y_i$ , or alternatively,  $b$  is proportional to the distance of its zeros and the sign can be extracted from  $\partial(y_i(\theta))/\partial\theta$ . If no two zeros in the function segment  $y_i(\theta)$ ,  $\theta \in [-\tau, 0]$  exist the formulas (13) and (14) can be used and they complement nicely provided  $t_0 \geq \tau_{\max}$ .

Remark that for roots where Newton's iteration fails to converge (because of nearby other roots or because the starting value was not good enough) we still extract the root from the subspace iteration, thereby keeping the robustness of the original algorithm.

To conclude this section we give some details of the Newton-Raphson iteration schemes.

When good starting values are available real, simple zeros of (5) can be found quadratically with the iteration scheme:

$$\begin{cases} \lambda^{(k+1)} = \lambda^{(k)} + \lambda_\epsilon \\ v^{(k+1)} = v^{(k)} + v_\epsilon \end{cases} \quad (17)$$

with  $[v_\epsilon \ \lambda_\epsilon]^T$  the solution of

$$\begin{bmatrix} \Delta(x^*, \lambda^{(k)}) & \Delta_\lambda(x^*, \lambda^{(k)})v^{(k)} \\ c^T & 0 \end{bmatrix} \begin{bmatrix} v_\epsilon \\ \lambda_\epsilon \end{bmatrix} = - \begin{bmatrix} \Delta(x^*, \lambda^{(k)})v^{(k)} \\ c^T v^{(k)} - 1 \end{bmatrix},$$

and

$$\Delta_\lambda(x^*, \lambda) \triangleq \frac{\partial \Delta}{\partial \lambda}(x^*, \lambda) = I + \sum_{j=1}^m \tau_j A_j(x^*) e^{-\tau_j \lambda}.$$

Complex zeros  $\lambda = a + bi$  can be found with:

$$\begin{cases} \lambda^{(k+1)} = \lambda^{(k)} + (a_\epsilon + b_\epsilon i) \\ v^{(k+1)} = v^{(k)} + v_\epsilon \\ w^{(k+1)} = w^{(k)} + w_\epsilon \end{cases}$$

with  $[v_\epsilon \ w_\epsilon \ a_\epsilon \ b_\epsilon]^T$  the solution of

$$\begin{bmatrix} \Re(\Delta) & -\Im(\Delta) & d_{11} & d_{12} \\ \Im(\Delta) & \Re(\Delta) & d_{21} & d_{22} \\ c^T & 0 & 0 & 0 \\ 0 & c^T & 0 & 0 \end{bmatrix} \begin{bmatrix} v_\epsilon \\ w_\epsilon \\ a_\epsilon \\ b_\epsilon \end{bmatrix} = - \begin{bmatrix} \Re(\Delta)v^{(k)} - \Im(\Delta)w^{(k)} \\ \Im(\Delta)v^{(k)} + \Re(\Delta)w^{(k)} \\ c^T v^{(k)} - 1 \\ c^T w^{(k)} \end{bmatrix}, \quad (18)$$

where the dependencies of  $\Delta$  on  $x^*$ ,  $\mu$  and  $a^{(k)} + b^{(k)}i$  have been omitted and

$$\begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} = \begin{bmatrix} \frac{\partial \Re(\Delta)}{\partial a} v^{(k)} - \frac{\partial \Im(\Delta)}{\partial a} w^{(k)} & \frac{\partial \Re(\Delta)}{\partial b} v^{(k)} - \frac{\partial \Im(\Delta)}{\partial b} w^{(k)} \\ \frac{\partial \Im(\Delta)}{\partial a} v^{(k)} + \frac{\partial \Re(\Delta)}{\partial a} w^{(k)} & \frac{\partial \Im(\Delta)}{\partial b} v^{(k)} + \frac{\partial \Re(\Delta)}{\partial b} w^{(k)} \end{bmatrix}.$$

The use of the extra vector  $v$  in (5) raises the dimension of (1) from one to  $n + 1$ , but the computational costs of the Newton iteration are reduced because almost everything can be precomputed (see (17) and (18)) and no determinants have to be evaluated.

### 3.4 Continuation and bifurcation analysis

If equation (1) contains some physical parameters,

$$\dot{x} = f(x(t), x(t - \tau_1), \dots, x(t - \tau_m), \mu), \mu \in \mathbb{R}^p,$$

one is interested in the movement of the eigenvalues of the characteristic equation as a function of these parameters.

The above procedure can be used for every point  $x^*(s)$ ,  $\mu(s) \in \mathbb{R}$  of a branch of steady state solutions, parameterized by some arclength  $s$ , to compute the stability evolution on the branch and in particular to detect Hopf bifurcations. One could also follow a branch of Hopf bifurcations in a two-dimensional parameter space  $\mu \in \mathbb{R}^2$  using [13], and compute the rightmost eigenvalues  $\lambda_k$  to detect double Hopf points (complicated dynamics can arise in the neighbourhood of double Hopf points, see [3] and for ODEs [10, section 8.6]).

The use of locking as described in the previous section is of special interest in a continuation context because the modes of the previous point  $x^*(s_{k-1})$  are approximations of the modes of the current point  $x^*(s_k)$  and can be used as starting values in the Newton process (17). The values that converge can already be put in the locked subspace.

To detect bifurcations we are interested in parameter values for which a real zero or a complex pair of zeros of the characteristic equation crosses the imaginary axis. We can, for instance, define the following scalar test function  $\rho(s)$  based on the real parts of the computed rightmost zeros  $\lambda_k(s)$ ,  $k = 1, \dots, l$ , of the characteristic equation at  $s$ ,

$$\rho(s) \triangleq \min\{\Re(\lambda_k(s)), k = 1, \dots, l\}.$$

Although this function has discontinuities with sign change it will be continuous in a neighbourhood of a regular bifurcation point where one can safely use linear interpolation to calculate its zeros.

## 4 Test examples

### 4.1 A model for recurrent neural feedback

As a first example we study a model developed by Plant [16] of which a bifurcation diagram and periodic solutions are given in [4]. The model is based on the system

$$\begin{cases} v'(t) = h(v(t)) - w(t) + \mu(v(t - \tau) - v^*) \\ w'(t) = \rho(v(t) + a - bw(t)) \end{cases} \quad (19)$$

with

$$h(v) = v - \frac{v^3}{3}; \quad a = 0.7, \quad b = 0.8, \quad \rho = 0.08, \quad \tau = 25$$

and  $v^* \approx -1.1994$  the unique real root of  $h(v^*) - (v^* + a)/b = 0$  which, together with  $w^* = (v^* + a)/b$  corresponds to the unique equilibrium of the system.  $\mu$  is used as the continuation parameter.

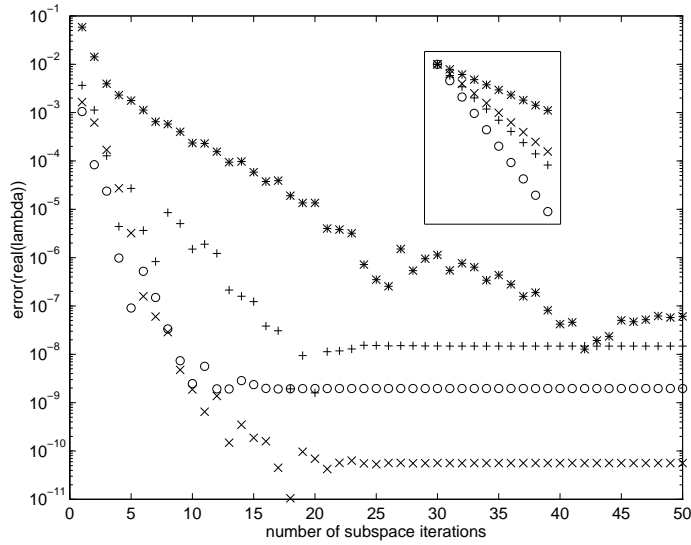


Figure 1: The convergence during subspace iteration of the real parts of the 8 rightmost roots of the characteristic equation of (19) at  $(v^*, w^*)$ ,  $\mu = -0.5$ . Since all roots consist of complex pairs only four lines are indicated. The symbols are as follows: 'o' for  $\lambda_{1,2}$ , '+' for  $\lambda_{3,4}$ , 'x' for  $\lambda_{5,6}$  and '\*' for  $\lambda_{7,8}$ . The theoretical asymptotic convergence rate is indicated in the upper right part of the picture.

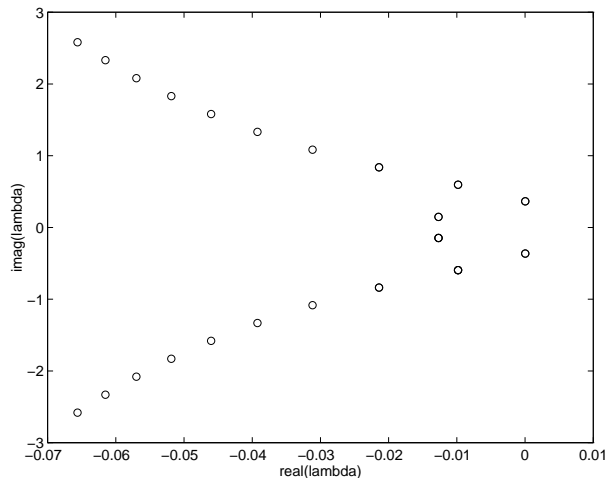


Figure 2: The location of the rightmost roots of the characteristic equation of (19) at  $(v^*, w^*)$ ,  $\mu = -0.5$ . The computation of the 8 rightmost roots of this picture is shown in figure 1.

To illustrate the convergence of the subspace iteration we fixed  $\mu = -0.5$  and used a basis of size  $l = 8$  and  $t_0 = \tau = 25$ . Each subspace iteration thus requires 8 time integrations over an interval of length  $t_0$ . The convergence of the real parts of the computed rightmost  $\lambda_k$ ,  $k = 1 \dots 8$ , is depicted in figure 1. Only four different lines are seen because all computed eigenvalues consist of complex pairs. The asymptotic convergence rate is given by  $\frac{\|\sigma_{l+1}\|}{\|\sigma_l\|} = \exp((\Re(\lambda_{l+1}) - \Re(\lambda_l))t_0)$  and is depicted in the upper right part of figure 1 using the corresponding symbols. The error of the eigenvalues at each step was computed using the corresponding eigenvalue obtained through Newton iteration. We clearly see that the maximal accuracy obtained by subspace iteration differs for different eigenvalues. This is not only due to the different position of the eigenfunctions in the subspaces, through which they depend on each other; but also due to the discretisation error made in the representation of the function segments and the time integration. In particular our example seems to indicate a correspondence between the imaginary part of  $\lambda$  and the maximal accuracy obtained by subspace iteration (compare figures 1 and 2 and note that the maximal accuracy of  $\lambda_{5,6}$  is smaller than those of  $\lambda_{1,2}$  and  $\lambda_{3,4}$ ).

In figure 3 the real parts of the rightmost roots of the characteristic equation are plotted along the constant equilibrium branch  $(v^*, w^*)$  for varying

$\mu$ . A solid line indicates a complex pair of roots, hence when such a line crosses the dashed-dotted null-line a Hopf bifurcation occurs. The resulting branches of periodic solutions can be found in [4, figure 4.1]. When  $\mu$  goes to zero the delay term of (19) disappears. The resulting ODE system has only two modes which can easily be calculated as the eigenvalues of the Jacobian,

$$\lim_{\mu \rightarrow 0} \lambda_{1,2}(\mu) \approx -0.2513 \pm 0.2119i.$$

The real parts of all other modes tend to  $-\infty$  when  $\mu$  decreases towards zero.

The left part of picture 3 was computed with a fixed steplength  $\Delta\mu = 0.01$  and a basis of size  $l = 24$ . At each point the iteration was halted when the relative change of the real parts of the first 18 modes between two subspace iteration steps was less than 0.01. For every point but the first one we used the last subspace basis of the previous point as initial set in step (a) of algorithm 3.1. This resulted in 9 subspace iterations for the first point and a mean of 2.26 subspace iterations for all subsequent points, giving a total mean of  $2.33 \times 24 \approx 56$  time integrations of length  $t_0 = 25$  per point. If we denote the unknown absolute error at the  $k$ -th subspace iteration as  $e_k$ , the halting condition can be written as,  $i = 1 \dots 18$ ,

$$\frac{(\Re(\lambda_i) + e_{k-1}) - (\Re(\lambda_i) + e_k)}{\Re(\lambda_i)} \leq 0.01. \quad (20)$$

Assuming the asymptotic convergence rate has been reached,

$$e_k \approx \exp((\Re(\lambda_{l+1}) - \Re(\lambda_i))t_0)e_{k-1},$$

we can rewrite (20) as

$$\frac{e_k}{\Re(\lambda)} \leq \frac{0.01}{\exp(-(\Re(\lambda_{l+1}) - \Re(\lambda_i))t_0) - 1} \leq \frac{0.01}{\exp(-(\Re(\lambda_{l-1}) - \Re(\lambda_i))t_0) - 1}.$$

This error estimate depends on  $\mu$  and can be monitored during continuation. In our situation we conclude that the largest relative error of  $\Re(\lambda_{1,2})$ ,  $\Re(\lambda_{10,11})$  and  $\Re(\lambda_{17,18})$  in the interval  $\mu \in [-1, -0.01]$  is smaller than respectively  $2.3e-3$ ,  $7.1e-3$  and  $2.8e-2$ .

When applying locking as described in section (3.3) the purpose of the subspace iteration is reduced to finding starting values for the Newton iteration. The left part of picture 3 was again computed with the same fixed steplength  $\Delta\mu = 0.01$  and a basis of unlocked vectors of size  $l = 6$ . Each time a computed root satisfies (20) it is used as a starting value for the Newton iteration. Upon success it is removed from the active basis of unlocked

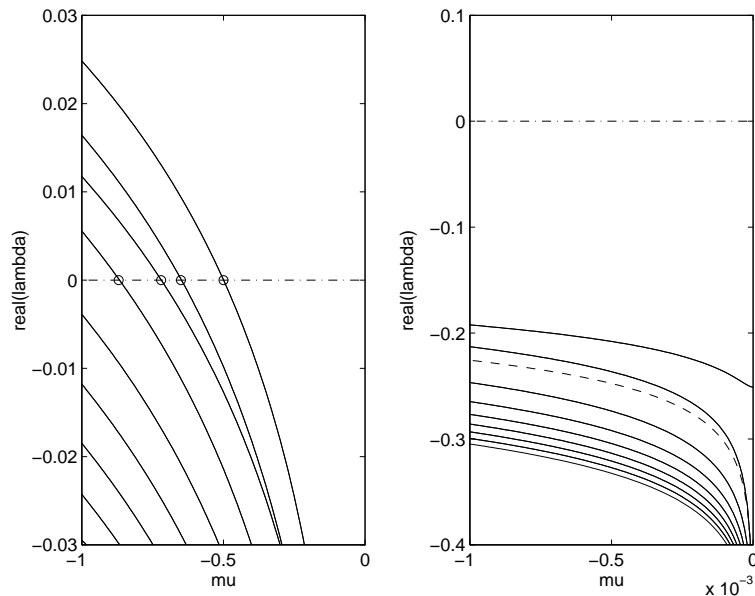


Figure 3: The real parts of the 19 rightmost roots of the characteristic equation of (19) along the constant solution branch  $(v^*, w^*)$  for varying  $\mu$  (left:  $\mu \in [-1, 0]$ , right:  $\mu \in [-0.001, 0]$ ). Real roots are indicated as dashed lines, complex pairs of roots as solid lines, Hopf bifurcations are indicated with a 'o'.

vectors. In such a way, the rightmost 18 roots of the first point are computed. For subsequent points the computed roots of the previous point can already be used as starting values for the Newton iteration. The purpose of the subspace iteration is then further reduced to watch for new upcoming modes. If the first unlocked mode satisfies (20) and its real part is smaller than all locked modes, we go on to the following point. If its real part is larger, it is refined using the Newton iteration and interchanged with the leftmost locked mode. This scheme resulted in 29 subspace iterations for the first point and a mean of 1.98 subspace iterations for all subsequent points, giving a total mean of  $2.25 \times 6 \approx 14$  time integrations of length  $t_0 = 25$  per point. The roots are now obtained to full accuracy.

## 4.2 A two-delay model

Our second test case is a model that was used to study the influence of multiple negative delayed feedback loops [1]. It is a scalar delay differential

equation with two delays and a unique equilibrium at  $x^* = 0$ .

$$\dot{x}(t) = -\tanh(x(t - \tau_1)) - A \tanh(x(t - \tau_2)) \quad (21)$$

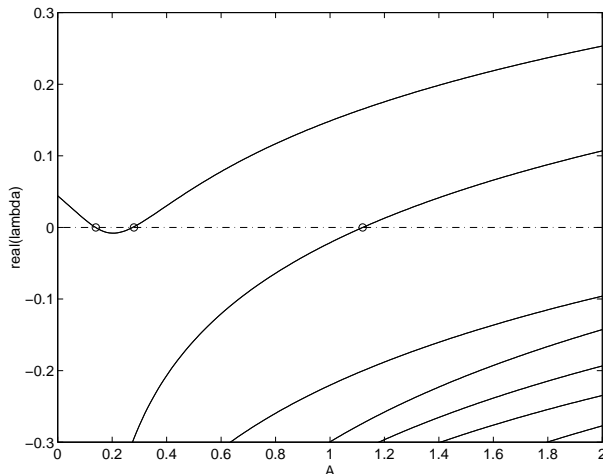


Figure 4: The real parts of the rightmost roots of the characteristic equation of (21) along the constant solution branch  $x^* = 0$  for  $\tau_1 = 1.75$ ,  $\tau_2 = 5$  and varying  $A \in [0, 2]$ . All calculated roots consist of complex pairs, Hopf bifurcations are indicated with a 'o'.

In figure 5 we see regions of linearized stability of the fixed point of decreasing size. As such regions get smaller they are easily overlooked when monitoring only the number of unstable modes. A similar region of linearized stability is seen in figure 4. The different movement of the eigenvalues  $\lambda_k$  suggests different dynamical behaviour in the neighbourhood of this region compared to those of figure 5.

## 5 Brief discussion of alternatives

In [11] a method is proposed to calculate a scalar function  $\nu(r) : \mathbb{R} \rightarrow \mathbb{R}$  that has a pole at every  $r = \Re(\lambda_i)$  with  $\lambda_i$  a root of characteristic equation (4). Although the calculation of  $\nu(r)$  for specific  $r$  is cheap when  $n$  is not very large this method has certain disadvantages. Finding all rightmost poles of a function is not easily done automatically. Determining double or near double poles is a difficult task. Also, when following a branch of Hopf points in two-parameter space the detection of double Hopf points is hindered by the constant pole at  $r = 0$ . The method is further restricted

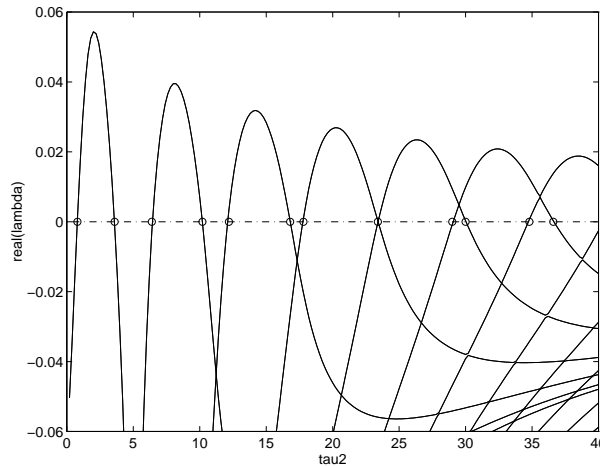


Figure 5: The real parts of the rightmost roots of the characteristic equation of (21) along the constant solution branch  $x^* = 0$  for  $A = 0.15$ ,  $\tau_1 = 1.5$  and varying  $\tau_2 \in [0, 40]$ . All calculated roots consist of complex pairs, Hopf bifurcations are indicated with a 'o'.

to commensurate delays which is impractical in cases of multiple delays, for instance when using one of the delays as a continuation parameter. On the other hand when one wants to calculate a large number of roots of the characteristic equation, especially with large imaginary parts our method becomes infeasible because of the accuracy needed in the time integration.

An alternative to our approach would be to approximate the variational equations (7) by a high dimensional system of ordinary differential equations. The Jacobian of the right hand side of this system evaluated in an equilibrium solution has a sparse structure that depends on the number and the relative position of the delays. One still seeks the rightmost roots of this large matrix (in contrast with our approach). Techniques have been developed for this purpose [14], but when there is a large 'tail' of eigenvalues with large negative real parts the robustness of these methods is limited if no good guesses of the wanted eigenvalues are available. Because the actual delay differential equation has a tail of eigenvalues with real parts going to  $-\infty$ , the situation gets worse when the number of discretisation points is increased. On top of this, approximating a DDE by a high-dimensional ODE may be dangerous, specifically because stability of time integration codes of ODEs and DDEs differs [8, 7, 9].

## 6 Conclusions

In this paper we presented a robust and automatic numerical method to compute the rightmost roots of the characteristic equation of a fixed point of a DDE. These roots determine the stability of the fixed point, provide insight into the system's behaviour, can be used for accurate bifurcation detection and efficient indirect calculation of bifurcation points. Examples were presented that demonstrate the usefulness of the algorithm. We also believe this method can be extended to other types of functional differential equations and we intend to use the algorithm to study the influence of delay on the stability of equilibria of nonlinear feedback loops.

## 7 Acknowledgements

The authors wish to thank K. Lust for many interesting discussions and helpful comments. This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture (IUAP P4/02) and of the research project OT/94/16, funded by the Research Council K.U.Leuven. The scientific responsibility rests with its authors. K. Engelborghs is a Research Assistant of the Fund for Scientific Research (Flanders).

## References

- [1] J. Bélair and S. A. Campbell. Stability and bifurcations of equilibria in a multiple-delayed differential equation. *SIAM Journal of Applied Mathematics*, 54(5):1402–1424, October 1994.
- [2] J. Bélair, S. A. Campbell, and P. Van Den Driessche. Frustration, stability and delay-induced oscillations in a neural network model. *SIAM Journal of Applied Mathematics*, 56(1):245–255, February 1996.
- [3] S. Campbell and V. G. LeBlanc. Resonant Hopf-Hopf interactions in delay differential equations. *Journal of Dynamics and Differential Equations*, 1996. Submitted.
- [4] A. M. Castelfranco and H. W. Stech. Periodic solutions in a model of recurrent neural feedback. *SIAM Journal of Applied Mathematics*, 47(3):573–588, June 1987.

- [5] O. Diekmann, S. A. van Gils, S. M. Verduyn Lunel, and H.-O. Walther. *Delay Equations: Functional-, Complex-, and Nonlinear Analysis*, volume 110 of *Applied Mathematical Sciences*. Springer-Verlag, 1995.
- [6] Jack K. Hale. *Theory of Functional Differential Equations*, volume 3 of *Applied Mathematical Sciences*. Springer-Verlag, 1977.
- [7] T. Hong-Jiong and K. Jiao-Xun. The numerical stability of linear multistep methods for delay differential equations with many delays. *SIAM Journal of Numerical Analysis*, 33(3):883–889, June 1996.
- [8] K. J. in 't Hout. A new interpolation procedure for adapting Runge-Kutta methods to delay differential equations. *BIT*, 32:634–649, 1992.
- [9] K. J. in 't Hout. Stability analysis of Runge-Kutta methods for systems of delay differential equations. *IMA Journal of Numerical Analysis*, 17:17–27, 1997.
- [10] Yuri A. Kuznetsov. *Elements of Applied Bifurcation Theory*, volume 112 of *Applied Mathematical Sciences*. Springer-Verlag, 1995.
- [11] J. Louisell. Numerics of the stability exponent and eigenvalue abscissas of a matrix delay system. Technical report, 1997. Preprint.
- [12] T. Luzyanina, K. Engelborghs, K. Lust, and D. Roose. Computation, continuation and bifurcation analysis of periodic solutions of delay differential equations. *International Journal of Bifurcation and Chaos*, 7(11), November 1997.
- [13] T. Luzyanina and D. Roose. Numerical stability analysis and computation of Hopf bifurcation points for delay differential equations. *Journal of Computational and Applied Mathematics*, 72:379–392, 1996.
- [14] K. Meerbergen and D. Roose. Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems. *IMA Journal of Numerical Analysis*, 16:297–346, 1996.
- [15] G. Moore and A. Spence. The calculation of turning points of nonlinear equations. *SIAM Journal of Numerical Analysis*, 17(4):567–576, August 1980.
- [16] R. E. Plant. A FitzHugh differential-difference equation modeling recurrent neural feedback. *SIAM Journal of Applied Mathematics*, 40(1):150–162, February 1981.

- [17] Youcef Saad. *Numerical Methods for Large Eigenvalue Problems*.  
Manchester University Press, 1992.