University of Chester

This work has been submitted to ChesterRep – the University of Chester's online research repository

http://chesterrep.openrepository.com

Author(s): Neville J Ford; Patricia M Lumb

Title: An algorithm to detect small solutions in linear delay differential equations

Date: 2006

Originally published in: Journal of Computational and Applied Mathematics

Example citation: Ford, N. J., Lumb, P. M. (2006). An algorithm to detect small solutions in linear delay differential equations. *Journal of Computational and Applied Mathematics*, 193, 121-139

Version of item: Preprint

Available at: http://hdl.handle.net/10034/11816

An algorithm to detect small solutions in linear delay differential equations

Neville J Ford[1] and Patricia M Lumb

Mathematics Department, University College Chester, Parkgate Road, Chester, CH1 4BJ, UK

Abstract

This paper presents an algorithm that provides a simple reliable mechanism for the detection of small solutions in linear delay differential equations. We provide background that emphasises the importance of detecting small solutions, we review existing experimental results and provide a mathematical justification for our choice of algorithm. The paper concludes with some examples.

1 Introduction and Background

This paper focuses on the development of a computer program that will determine whether or not a linear delay differential equation of the form

$$x'(t) = \sum_{j=0}^{N} a_j(t)x(t - \tau_j) \text{ where } \tau_0 = 0, \tau_{j+1} > \tau_j, j = 0, \ldots, N-1 \qquad (1)$$

admits small solutions. Our aim is to produce an *automatic* detection algorithm so that the user does not need to understand the methodology underlying the process by which the decision is made. We developed the program using Matlab.

The detection of small solutions, that is solutions $x(t)$ for which $\lim_{t \to \infty} e^{kt}x(t) = 0$, for all $k \in \mathbb{R}$ (see [3,9,11]), is important in the qualitative analysis of delay differential equations ([1]). It has been shown that delay differential equations that have small solutions are particularly difficult to analyse and one needs to beware of applying certain standard analytical methods when small solutions are present. Unfortunately, the detection of small solutions by direct analysis is, in general, a hard problem and therefore the idea of using a numerical routine is attractive. In previous work ([4–7,9]) various theoretical and experimental results have been presented that justify our belief that the detection

---

[1] corresponding author email: njford@chester.ac.uk

of small solutions can be accomplished effectively through numerical methods and these techniques now form the basis for the algorithm that we present in this paper.

For some non-autonomous problems of the form (1) there exists an equivalent autonomous problem in the sense that the dynamics of the solutions of the two problems are the same (see [9]). It turns out that such an equivalent autonomous problem exists if and only if the underlying equation has no small solutions. In our previous work we built on this fundamental idea: We began by finding the candidate autonomous problem that may be equivalent to the given non-autonomous equation, then ([4–7]) we considered the eigenspectra of the solution maps of the two problems after they had each been subjected to numerical discretisation (using the trapezium rule, for example). We were able to deduce, by examining plots of these eigenspectra, that there were characteristic shapes in the figures that enabled us to identify correctly the presence, or otherwise, of small solutions, and hence determine whether or not an equivalent autonomous problem existed.

The task that we descibe in this paper is of automating the process to remove the visualisation step that requires human intervention/interpretation and provide a reliable and robust automatic procedure for determining whether a given linear delay differential equation has small solutions.

The problem of looking for small solutions is made harder by the fact that they cannot usually be seen when the solution of the equation is plotted. This is because the actual solution in any given case is a linear combination of the (generalised) eigenfunctions of the differential equation, and the coefficients are dependent upon the initial function. Therefore unless all the coefficients corresponding to *non-small* eigenfunctions are zero, one cannot expect the actual solution to the differential equation with a given initial function to be small even though the equation may *possess* small solutions

Small solutions of linear delay differential equations obviously correspond to eigenvalues close to the origin and therefore when we consider the eigenspectra our attention focuses on the eigenvalues near the origin. In our previous work we needed to vary the magnification of the eigenspectra near to the origin to suit the equation under consideration and we found that we could then draw reliable conclusions. Even for those equations close to critical parameter values where the property of having small solutions changes, our visualisation methods were effective. As would be expected, one needs to use quite a small step length in the numerical approximation to reflect accurately the true behaviour of the dynamical system. We experimented with even smaller step lengths but found that this did not improve detection even close to critical parameter values.

In section 2 we use the cartesian form of the eigenvalues in our consideration of a one to one mapping between two ordered sets of eigenvalues. We illustrate how differences between the results depend upon whether or not the problem admits small solutions. In section 3 we show how using the polar form of the eigenvalues provides greater reliability and motivates the the development of our algorithm.

In section 4 we introduce our algorithm and explain the underlying methodology. We

2

consider its use and reliability and provide illustrative examples.

2   Using the cartesian form of the eigenvalues

2.1   The basic delay equation

We first restrict ourselves to equations of the form (2), (a simple case of (1)), which we considered in our previous work (see [4]). It is well-known (see [11,13]) that if $b(t)$ does not change sign then equation (2) does not admit small solutions and in this case (2) and (3) are equivalent.

$$\dot{x}(t) = b(t)x(t-1) \text{ with } b(t+1) = b(t) \qquad \text{(a non-autonomous problem).} \qquad (2)$$

$$\dot{x}(t) = \hat{b}x(t-1) \text{ where } \hat{b} = \int_0^1 b(t)dt \qquad \text{(an autonomous problem).} \qquad (3)$$

2.2   Fundamentals of our approach

We will use (discrete) numerical approximations to derive information about the exact analytical properties of the underlying continuous equation. This is a normal approach in cases where direct analysis does not yield the required information. However in the analysis of delay equations in general and small solutions in particular, the approach poses particular challenges and therefore we shall spend a little time here reviewing what is already known and quoting a Theorem that justifies our methods.

As is well known, delay equations are infinite dimensional problems, requiring as they do the specification of a function over an initial interval to derive a unique solution. The use of a fixed step length numerical scheme to approximate the solution of the delay equation results in a reduction of the dimension of the problem to some fixed finite order (dependent on the step length chosen) and therefore the infinite dimensionality of the problem is sacrificed. The existence of small solutions to an equation is an infinite dimensional property and this cannot be precisely represented in the numerical approximation because of this loss of dimensionality. Therefore one might surmise that a numerical method cannot be used to detect the presence of small solutions. The major theme of the paper [8] is to show that (perhaps surprisingly) numerical methods of this type can be used efficiently in small solution detection.

The key idea is to consider whether the equations (2) and (3) are equivalent as dynamical systems and we do this by considering how a numerical approximation would be used to solve each equation. This means that we need to be concerned with the extent to which we can rely on the numerical scheme to give a true representation of each of the dynamical systems. Once we can be sure that the numerical scheme represents the dynamical systems

3

faithfully then we can compare the eigenspectra of the two numerical schemes as a means of comparing the eigenspectra of the underlying continuous problems. Reassurance on this point is provided by the following Theorem:

Theorem 1 *(see Theorem 3.1 of [8] and Theorem 3.2 of [10]) Apply a strongly stable linear multistep method of order $p \geq 1$ to the autonomous delay differential equation*

$$y'(t) = \alpha y(t - \tau) \tag{4}$$

with characteristic roots that satisfy

$$\lambda - \alpha e^{-\tau\lambda} = 0. \tag{5}$$

For each fixed step length $h = \frac{\tau}{m} > 0$ the numerical method has a set S_h of $m + 1$ characteristic roots of the equation

$$\lambda^m \rho(\lambda) - h\alpha\sigma(\lambda) = 0, \tag{6}$$

where $\rho(\lambda)$ and $\sigma(\lambda)$ are, respectively, the first and second characteristic polynomials of the linear multistep method being used. Let λ be a root of equation (5) and define d_h to be the distance given by

$$d_h = min_{s \in S_h} |e^\lambda - s^m| \tag{7}$$

then d_h satisfies

$$d_h = O(h^p) \quad as \quad h \to 0. \tag{8}$$

This Theorem answers several key questions for us. Firstly it tells us that we should choose a strongly stable linear multistep method for our approximation. Following experimentation, we have chosen to use the trapezium rule. Secondly it tells us that characteristic roots with large negative real parts in the continuous case will show up as roots close to the origin for the discrete problem. Finally it tells us that, in the limit as $h \to 0$ we shall recover the countably infinite set of characteristic roots of the underlying continuous problem. As we shall see later, we use the idea of the limiting process in our decision-making by considering several different small values of h.

2.3   Examples of eigenspectra from our previous work

We have successfully detected the presence of small solutions to (2) by comparing the eigenspectra arising from discretisations of (2) and (3). In our diagrams we use '+' and '*' to indicate the eigenspectra arising from the non-autonomous problem and the autonomous problem respectively.

When small solutions are not present we expect the eigenvalue trajectory arising from the discretisation of the non-autonomous problem (2) to lie close to that arising from the equivalent autonomous problem (3). In fact, we can go further. When there are no small solutions, the (exact) characteristic values all lie on one curve (see [8]. Therefore

4

the existence of more than one trajectory of characteristic values can be taken to imply the presence of small solutions. The left-hand eigenspectra in Figure 1 are illustrative of the case where no small solutions are present. When (2) admits small solutions the two problems cannot be regarded as equivalent and we observe clear differences in the eigenspectra. We take the presence of *closed loops* to indicate that the equation admits small solutions and illustrate this in the right-hand eigenspectra of Figure 1. Further examples can be found in [4,5].
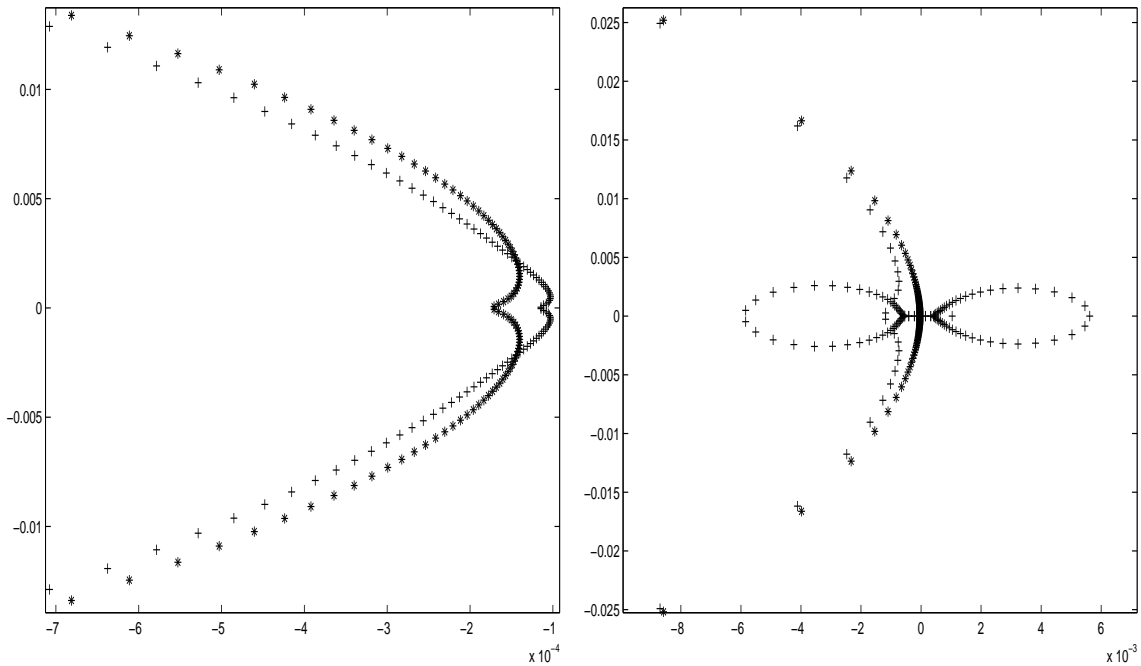

Fig. 1.
Left: The eigenspectra are very similar. The equation does not admit small solutions. The two problems are equivalent.
Right: Clear differences in the eigenspectra are visible. The equation admits small solutions. An equivalent autonomous problem does not exist

Our aim now is to see how the process of distinguishing the different cases from the figures can be automated.

2.4   Applying a numerical method

We obtained the eigenspectra in Figure 1 using the following approach: we applied a numerical method (the trapezium rule, in this case) with step-length $h = \frac{1}{N}$ to (2) to give an equation for y_{n+1} of the form

$$y_{n+1} = A(n)y_n, \tag{9}$$

where A(n), with $A(n) = A(n - N)$ for all $n > N$, is a companion matrix (see [4]). It follows that

$$y_{n+N} = Cy_n, \qquad \text{for } n = 1, 2, ... \qquad (10)$$

where

$$C = \prod_{i=1}^{N} A(N - i). \qquad (11)$$

In the autonomous problem (3) $A(n) = A$ is a constant matrix. This leads to a comparison of the eigenvalues of C with those of A^N.

We introduce
$\Lambda_1 = \{$eigenvalues of $C\} = \{z_{1,j},$ j=1, 2,...,N+1 $: z_{1,j}$ is an eigenvalue of C with $|z_{1,j}| \geq |z_{1,j+1}|$ and if $|z_{1,j}| = |z_{1,j+1}|$ then $arg(z_{1,j}) < arg(z_{1,j+1})\}$.
$\Lambda_2 = \{$eigenvalues of $A^N\} = \{z_{2,j},$ j=1, 2,...,N+1 $: z_{2,j}$ is an eigenvalue of A^N with $|z_{2,j}| \geq |z_{2,j+1}|$ and if $|z_{2,j}| = |z_{2,j+1}|$ then $arg(z_{2,j}) < arg(z_{2,j+1})\}$.

We examine whether the two (ordered) sets of eigenvalues, Λ_1 and Λ_2, arise from equivalent problems.

When the two problems are equivalent, that is equation (2) does not admit small solutions, then we expect each eigenvalue arising from discretisation of (2) to approximate an eigenvalue arising from discretisation of (3). The approximation should improve as the step size decreases (and the dimensionality of the problem increases).

We concentrate for the moment on the case where the two problems are equivalent. Let $z_{1,j} = x_{1,j} + iy_{1,j}$, $z_{2,j} = x_{2,j} + iy_{2,j}$. We set up a one-one mapping between these two sets of eigenvalues (after choosing the *ordering* as above) and, for $j = 1, ..., N + 1$ we evaluate the usual distances d_j where $d_j = \sqrt{(x_{2,j} - x_{1,j})^2 + (y_{2,j} - y_{1,j})^2}$. We expect the improvement in the approximation as the step size decreases to be reflected in measures of location and dispersion of the distribution of the d_j. As the step length decreases we expect the values of d_j to tend to zero.

Now we apply some basic statistical methods in our analysis of the d_j. We use the mean, standard deviation, skewness and kurtosis. Skewness reflects the degree to which a distribution is asymmetrical. Kurtosis reflects the degree to which a distribution is 'peaked', providing information about the *height* of the distribution relative to the value of its standard deviation. Now it is quite clear, from looking at the figures, that when there are no small solutions, the values of all the d_j should satisfy $d_j \to 0$ as $h \to 0$ while when there are small solutions present, the ordering will *match up the wrong pairs* and so $d_j \not\to 0$ for some j. We explore whether differences (in the shape of the distributions of the d_j) according to whether the problem admits small solutions can be identified easily through calculations of mean, standard deviation, skewness or kurtosis.

6

2.5 Examples

Example 2.1 *We consider first the distributions of the distances d_j for equation (2) with $b(t) = \sin(2\pi t) + c$ for different values of c and as $h = \frac{1}{N}$ varies. In this case small solutions are known to arise if and only if $b(t)$ changes sign on $[0, 1]$, that is, if and only if $|c| < 1$ ([11]). In Figure 2 the box plots illustrate the cases $c = 0.5$ and $c = 1.5$. In both cases we observe a decrease in the range of values of d_j and in the median value as the step size decreases. The interquartile range is seen to decrease steadily as the step length decreases when small solutions are not admitted but the situation is less clear when $c = 0.5$ and the equation admits small solutions.*
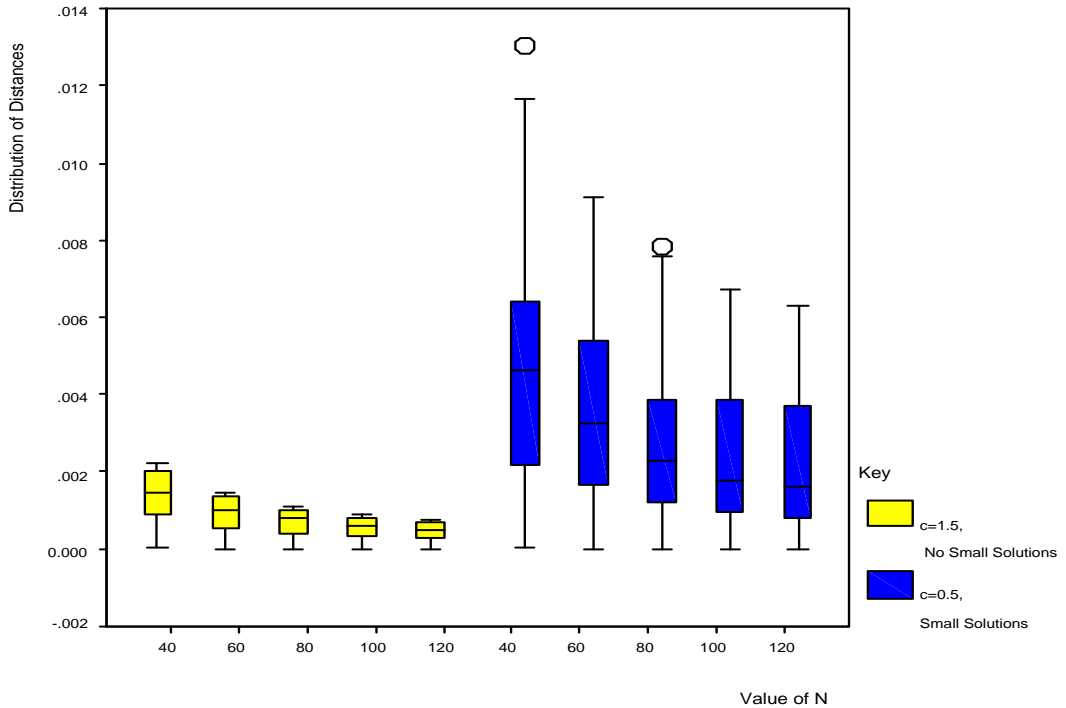

Fig. 2. A Comparison of the Distributions of the d_j for varying values of N, with (left) $c = 1.5$ and (right) $c = 0.5$

There is a prima facie case for arguing that the distributions show clearly distinct behaviour in the two cases. Where there are small solutions both the mean and standard deviation are much larger in every case than in the corresponding plots for the problem without small solutions. Therefore we propose to investigate whether one can impose a threshold (which may be dependent on N) leading to the automatic detection of small solutions in this way.

As the step length decreases we expect the mean and standard deviation of the distribution of the d_j to decrease. This is evidenced in Figure 3 which shows the ninety-five per cent confidence intervals for the mean value of the distance between *corresponding* eigenvalues in Λ_1 and Λ_2 for $b(t) = \sin(2\pi t) + c$ and $c = 0.5, 1.5$. We observe the much wider intervals

7

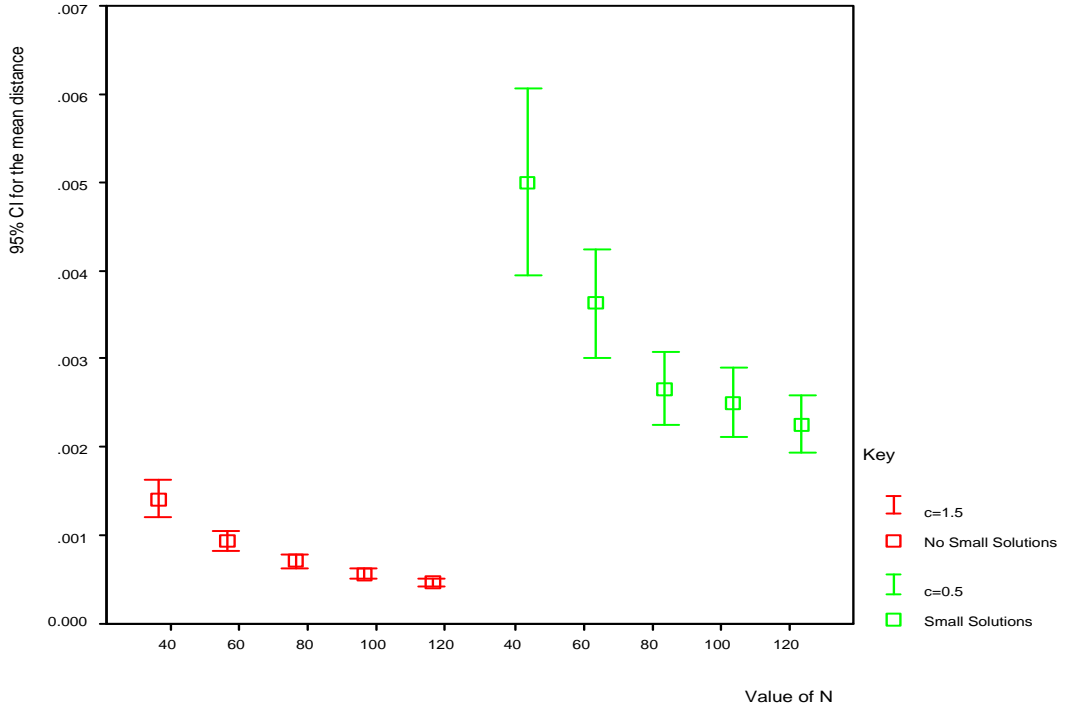and higher value for the mean when small solutions are present.

Fig. 3. The 95% Confidence interval for the mean distance between corresponding eigenvalues for varying values of N, with $c = 0.5$ and $c = 1.5$.

Example 2.2 *Figures 4 and 5 illustrate differences in the distributions of the d_j for different values of c, dependent upon whether or not $|c| < 1$. Again, a much greater variation in the values of d_j is observed for values of c for which the equation admits small solutions. When $c = 0$ almost all solutions are small. In Figure 4 we observe an increase in both the presence of outliers and in the mean distance as c approaches 0.*

However, Figures 4 and 5 also show the limitations of using the distribution of the d_j as the basis for making decisions. If we look at the values of c close to the critical values of ± 1 we observe that there are very similar distributions of d_j on either side of the boundary and that therefore it will be difficult to draw any reliable conclusions using either a threshold for the mean or one for the standard deviation of the d_j.

Example 2.3 *In Tables 1 and 2 we present the values of the kurtosis and skewness of the distribution of the d_j for varying values of c. Values of c equal to -1.5, 1.1. 1.5 and 3 correspond to problems which do not admit small solutions and in this case we observe similar values for different values of the step-length. Values of c of -0.5, 0.1 and 0.5 correspond to problems which admit small solutions. The situation here is very different. Considerable variation is observed for different step lengths. The evidence in favour of a one-one correspondence between the ordered sets of eigenvalues, Λ_1 and Λ_2 turns out to be weak.*

8

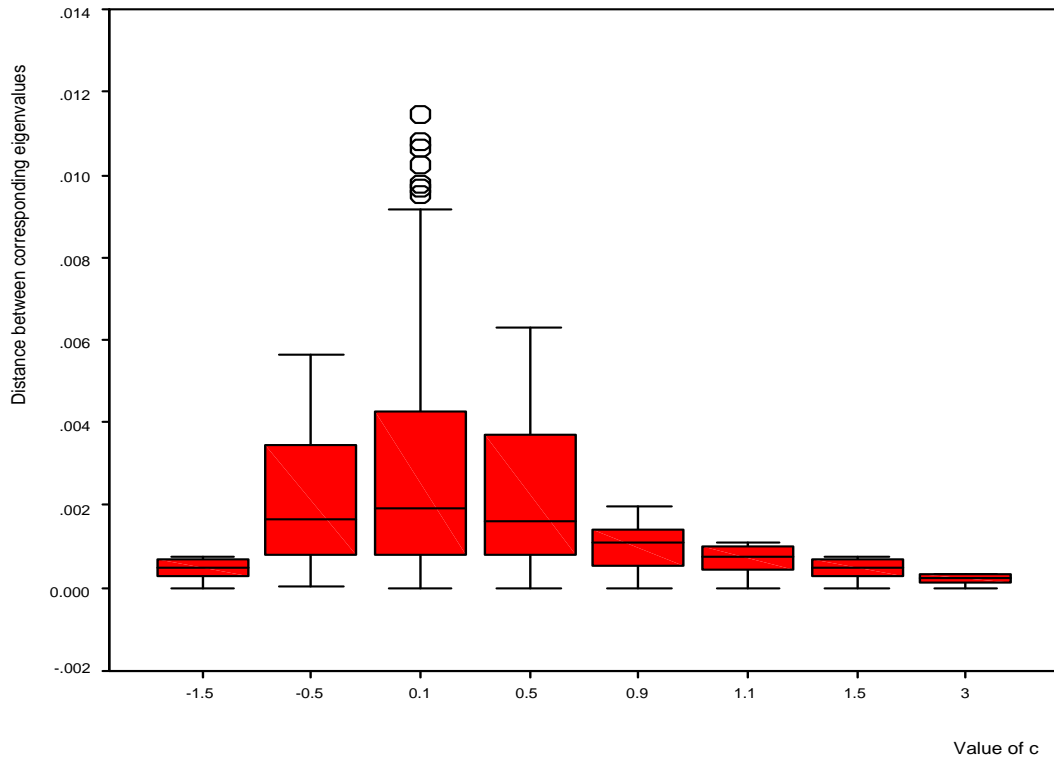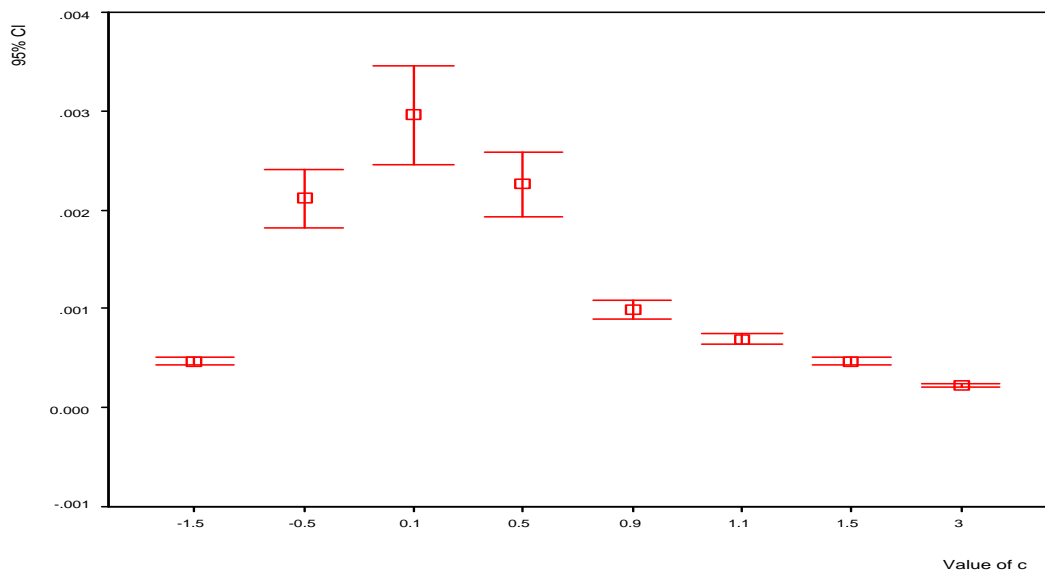Fig. 4. Distribution of d_j for different values of c

Fig. 5. 95% Confidence intervals for the mean distance between corresponding eigenvalues for different values of c

Remark 2.1 *When N is even we have chosen to disregard the one real eigenvalue of each of the matrices C and A^N since it is an outlier for the distribution and has the potential to affect the conclusions which we may be able to draw.*

	Value of the constant c						
N	-1.5	-0.5	0.1	0.5	1.1	1.5	3
20	-0.9805	-1.3494	-0.8987	-1.0654	-0.7645	-1.0068	-1.1627
40	-1.1103	-0.8628	-1.1396	-0.3964	-1.0624	-1.1287	-1.1593
60	-1.1212	-0.1906	-1.2311	-0.7312	-1.0962	-1.1307	-1.1397
80	-1.1170	-0.4978	-0.5819	0.1171	-1.0981	-1.1227	-1.1256
100	-1.1110	0.1868	-0.5294	-0.8360	-1.0940	-1.1145	-1.1157
120	-1.1053	-0.8658	0.9663	-0.8150	-1.0890	-1.1080	-1.1080

Table 1
Values of the kurtosis of the distribution of d_j for different values of c and N

	Value of the constant c						
N	-1.5	-0.5	0.1	0.5	1.1	1.5	3
20	-0.4977	-0.1484	-0.4673	0.1665	-0.5702	-0.4754	-0.4248
40	-0.4647	0.4142	-0.3549	0.5712	-0.4866	-0.4565	-0.4487
60	-0.4682	0.6600	0.1157	0.4281	-0.4840	-0.4642	-0.4626
80	-0.4736	0.7745	0.6337	0.7630	-0.4883	-0.4713	-0.4710
100	-0.4780	0.8656	0.8861	0.6758	-0.4926	-0.4766	-0.4764
120	-0.4814	0.6019	1.3091	0.6755	-0.4963	-0.4804	-0.4801

Table 2
Values of the skewness of the distribution of d_j for different values of c and N

Finally in this section we consider the use of Spearman's rank-order correlation coefficient to help us determine the degree to which a monotonic relationship (increasing or decreasing) exists between the two variables $|z_{1,j}|$ and $|y_{1,j}|$ (see [12] for example). A visual comparison of the eigenspectra indicates that, for the equations we are considering, the relationship between the magnitude of the eigenvalue and the magnitude of the imaginary part would be expected to be monotonic when the equation does not admit small solutions, but not otherwise. We explore in the examples below whether the (automatic) calculation of an appropriate Spearman's rank correlation coefficient can reliably answer the question 'Does an equation admit small solutions?'

Remark 2.2 *This approach, if it was successful, would be extremely attractive. Notice that the calculation of the statistic involves calculations only in terms of features of the eigenspectrum of the original delay differential equation under discretisation, and no longer relies on the computation of eigenspectra for an* equivalent *autonomous problem. For more general equations, we might be unable to write down an* equivalent *autonomous problem (for many equations, the formula is unknown analytically) yet this type of method would remain applicable.*

In Table 3 we present values of Spearman's rank correlation coefficient between the mag-

nitude of the eigenvalue and the magnitude of its imaginary part for the non-autonomous equation (2), with $b(t) = t - 0.5 + c, b(t + 1) = b(t)$, and for the autonomous equation (3) with $\hat{b} = c$. For this example small solutions are admitted if $|c| < 0.5$. We observe that the relationship is monotonic when small solutions are not admitted. A similar pattern emerged for other $b(t)$, including $b(t) = \sin 2\pi t + c$, $b(t) = t(t - 0.5)(t - 1) + c$ and $b(t) = \sin 2\pi t + t(t - 0.5)(t - 1)$.

	(non-autonomous)	(autonomous)		(non-autonomous)	(autonomous)
c	r_s	r_s	c	r_s	r_s
-1	1	1	0.1	0.871913	1
-0.9	1	1	0.2	0.893179	1
-0.8	1	1	0.3	0.935066	1
-0.7	1	1	0.4	0.967120	1
-0.6	1	1	0.5	1	1
-0.5	1	1	0.6	1	1
-0.4	0.954099	1	0.7	1	1
-0.3	0.851343	0.961307	0.8	1	1
-0.2	0.845831	0.963283	0.9	1	1
-0.1	0.836725	0.962309	1.0	1	1
0	0.829479	1			

Table 3
Values of Spearman's rank-order correlation coefficient between the magnitudes of the eigenvalues and their imaginary part using the eigenvalues of (2) with $b(t) = t - 0.5 + c$ and c varying

For those cases where c is chosen far from the critical value where small solutions appear the calculations provide some indication of their presence. However we can see quite clearly that close to the boundary, Spearman's rank correlation co-efficient does not provide the sensitivity we need to make predictions.

In conclusion, in this section we have reviewed the elementary statistics that could be calculated to determine whether small solutions arise for a particular problem. While the approaches we have considered provide useful insight, they are (somewhat unexpectedly) poor tools for distinguishing cases close to the critical values, and therefore we explore a quite different approach in the next section.

3 Insight from visualisation: consideration of the eigenvalues in polar form

Based on our experimental results (see [4–7]), we believe that results arising from the use of the polar form of the eigenvalues might be more easily extended to other classes of

equation, in particular to equations of the form $\dot{x}(t) = \sum_{j=0}^{m} b_j(t)x(t-jw)$ and to those higher dimensional systems when the eigenvalues of $A(t)$ in equation $y'(t) = A(t)y(t-1)$ are always real. Therefore we are motivated, both by the disappointing outcome of the investigation described in the previous section and the desire to produce a widely applicable algorithm, to consider the polar form.

When the analytical theory tells us that there should be small solutions, we have observed consistently some of the eigenvalues arising from discretisation of the non-autonomous problem lying close to the real axis and others lying on the negative real axis ([9]). In our work, (see [4,5]), we used the presence of closed loops that cross the x-axis to be characteristic of the cases where small solutions arise. We observe that the sizes of the arguments of the eigenvalues whose representation forms the 'additional' trajectory lie closer to 0 or 2π than those represented in the trajectory lying close to that arising from the autonomous problem. We use this idea as a basis for developing our method:

We use $z_{1,j}$ and $z_{2,j}$ as defined in section 2.4 and introduce
$M_1 = \{\alpha_{1,j} : \alpha_{1,j} = arg(z_{1,j}), j = 1, 2, ..., N+1\}$.
$M_2 = \{\alpha_{2,j} : \alpha_{2,j} = arg(z_{2,j}), j = 1, 2, ..., N+1\}$.
$L_1 = \{\alpha : 0 \leq \alpha < 0.5, \alpha = |\alpha_{1,j}|, \alpha_{1,j} \in M_1\}$.
$L_2 = \{\alpha : 3 < \alpha \leq \pi, \alpha = |\alpha_{1,j}|, \alpha_{1,j} \in M_1\}$.

We focus our interest on the distribution of $\alpha = \{|\alpha_{1,j}| : \alpha_{1,j} \in M_1\}$ for α lying in the intervals $[0, 0.5], (0.5, 1.0], (1.0, 1.5], (1.5, 2.5], (2.5, 3.0], (3.0, \pi]$.

Decreasing the step length from $\frac{1}{N_1}$ to $\frac{1}{N_2}$ increases the dimensions of the matrices C and A^N and leads to the calculation of $N_2 + 1$ eigenvalues instead of $N_1 + 1$ eigenvalues. We consider the question 'Where does the larger set of eigenvalues lie in relation to the previous set of eigenvalues?'. We investigated a range of step-lengths, observing where the additional eigenvalues fitted into the distribution and whether this depended upon the presence, or otherwise, of small solutions.

3.1 Mathematical basis for the algorithm

There is a simple mathematical justification for our approach. It is straightforward to show that only one characteristic value (the real root itself) of the autonomous problem lies close to the real axis (see, for example, [2] p. 305-316). We apply the approach in [10] to show that for the numerical scheme, as $h \to 0$, there will be only a single characteristic root close to the real axis. Therefore an equation without small solutions should have characteristic roots all but one of which lie away from the real axis. Thus when we detect more than one characteristic root in a neighbourhood of the real axis, this is sufficient to indicate the presence of small solutions.

3.2   Numerical Results

In the case when (2), with $b(t) = \sin 2\pi t + c$, does not admit small solutions then, for $h \geq \frac{1}{300}$, all the additional eigenvalues have arguments whose magnitudes lie in the range 0.5 to 2.5. This is not the case when (2) admits small solutions and we illustrate this difference in Tables 4 and 5. We note also that in Table 4 where the problem does not admit small solutions we observe no values of $\alpha > 2.5$, but in Table 5, when small solutions are possible, we observe values of $\alpha > 2.5$ for all values of N.

N	$\alpha < 0.5$	$0.5 < \alpha < 1.0$	$1.0 < \alpha < 1.5$	$1.5 < \alpha < 2.5$	$2.5 < \alpha < 3.0$	$3 < \alpha \leq \pi$
30	1	2	26	2	0	0
60	1	2	56	2	0	0
90	1	4	52	34	0	0
120	1	4	48	68	0	0
150	1	4	48	98	0	0
300	1	4	48	248	0	0
500	3	2	50	446	0	0
1000	3	4	54	940	0	0

Table 4

Distribution of |argument| of the eigenvalues for $c = -1.4$. No small solutions are present.

N	$\alpha < 0.5$	$0.5 < \alpha < 1.0$	$1.0 < \alpha < 1.5$	$1.5 < \alpha < 2.5$	$2.5 < \alpha < 3.0$	$3 < \alpha \leq \pi$
30	15	0	0	0	2	14
60	30	0	0	2	18	11
90	25	18	0	18	20	10
120	20	38	0	40	14	9
150	19	40	12	54	18	8
300	18	26	98	136	12	11
500	16	24	196	240	16	9
1000	18	20	432	498	24	9

Table 5

Distribution of the |argument| of the eigenvalues for c=0.1. Small solutions are present.

We now consider equation (2) with $b(t) = \sin 2\pi t + c$ for a range of values of c. In this case the *critical* values of c are when $c = \pm 1$. In Table 6 we present the number of eigenvalues of C for which the magnitude of the argument lies in each specified range and, in brackets, the corresponding figure for A^N. The divisions in the table effectively discriminate between the middle section where $|c| < 1$ and the non-autonomous equation admits small solutions and the other cases where small solutions are not present. It is clear that for equations of the form (2) which admit small solutions then the two sets of

figures are very dissimilar. We observe that (using $h = \frac{1}{128}$):

(1) $n(L_2) = 0$ and $n(L_1) = 1$ except near the critical functions when $c = \pm 1$.
(2) Near the critical functions when $c = \pm 1$ at least one of the statements $n(L_2) = 0, n(L_1) = 1$ is true.

Further detailed experimentation leads us to present the following tool as the basis on which our program makes the decision between *small solutions* and *no small solutions.*

Decision Tool 3.1 *Let M_1 be the set of eigenvalues arising from discretisation of $x'(t) = b(t)x(t-1), b(t+1) = b(t)$ using the trapezium rule (as in section 2.4). and define*
$L_1 = \{\alpha : \alpha \in M_1, 0 \leq |\alpha| < 0.5\}$,
$L_2 = \{\alpha : \alpha \in M_1, 3 < |\alpha| \leq \pi\}$.

When the equation $x'(t) = b(t)x(t-1), b(t+1) = b(t)$ does not admit small solutions then at least one of the following statements is true:

(1) $L_2 = \phi$ (or $n(L_2) = 0$)
(2) $n(L_1) = 1$.

It is worth mentioning that we have also considered the distribution of the magnitudes of the arguments of the eigenvalues after discretisation using the backward Euler and forward Euler methods. The shape of the distributions differed from that obtained using the trapezium rule but distinguishing between problems which admitted small solutions and those for which an equivalent autonomous problem exists can be achieved using a similar approach to that described here and it is just as effective.

4   Introducing the Program

The program 'smallsolutiondetector1' is written to answer the question 'Does an equation of the form
$$x'(t) = b(t)x(t-p), b(t+p) = b(t) \tag{12}$$
admit small solutions?' The program allows the user to detect small solutions to equations of the form (12) but actually transforms that equation to an equation of the form

$$y'(t) = b_1(t)y(t-1), b_1(t+1) = b_1(t) \tag{13}$$

using the transformation $b_1(t) = pb(pt)$. This transformation is internal to the program and transparent to the user.

The methodology underlying the algorithm is based on Decision Tool 3.1. We use the term *critical function* to refer to a function at the bifurcation point when the behaviour of the equation changes from admitting small solutions to not admitting small solutions and vice-versa. The program consists of the following stages:

(1) The user is asked to state the period/delay and to input their function $b(t)$.
(2) The eigenvalues of the matrix C, with C as defined in section 2.4, are calculated.
(3) The numbers of these eigenvalues with arguments lying in the intervals $[0, 0.5)$ and $(3, \pi]$ are calculated. The algorithm refers to these numbers as $n1$ and $n6$ respectively.
(4) If $n6 = 0$ we conclude that the equation does not admit small solutions.
(5) If $n6 > 0$ we also consider the value of $n1$.
    (a) If $n6 > 0$ and $n1 = 1$ we conclude that the equation does not admit small solutions but the user is warned that their function is near to a critical function.
    (b) If $n6 > 0$ and $n1 > 1$ we conclude that the equation admits small solutions.
    (c) We note that, to date, we have not experienced the situation when $n6 > 0$ and $n1 = 0$. If this case does arise then the user is informed that a decision cannot be made using the algorithm.

We have considered the reliability of our algorithm with paricular reference to the decisions made near a critical function. In Table 7 we show, for three different $b(t)$, the value of c at which the algorithm's decision changes and the absolute difference between that value and the *theoretically correct value* to eight decimal places.
We make the following observations for the step lengths that we have considered:

(1) For $b(t) = \sin(2\pi t) + c$ the error is zero to 8 decimal places.
(2) For $b(t) = t - 0.5 + c$ the reduction in the error as the step length h decreases is of order h.
(3) For $b(t) = t(t - 0.5)(t - 1) + c$ the error is at most of the order of 10^{-5}.

The algorithm we present is based on months of experimentation and refinement. It would be attractive to base the algorithm purely on the number of eigenvalues with magnitude lying in $(3, \pi]$, a result of 0 implying that the equation does not admit small solutions and a value > 0 implying that the equation admits small solutions. The magnitude of the errors was considered in a similar way to that in Table 7. However, including the number of eigenvalues with magnitudes less than 0.5 in the decision-making process led to a significant increase in the reliability of our algorithm in detecting the presence of small solutions.

A modified algorithm is also available. It provides even greater reliability than before, but at the cost of additional time in calculating the outcome. Essentially it is based on the idea that if the decision is not very clear for a particular equation, then it is worth recalculating for neighbouring problems to see whether the combination of results provides greater clarity of decision. The modified algorithm repeats the decision-making process outlined above, but this time with each of the three functions $b(t)$ and $b(t) \pm \epsilon$. For each of the three functions the program decides whether the equation admits small solutions. Three decisions are possible for each of the three functions. We will refer to these decisions as:

Yes: The equation admits small solutions.
No: The equation does not admit small solutions.
No/Near: It is unlikely that the equation admits small solutions but you are near to a critical function.

The algorithm considers all 27 possibilities and a decision is made for the function $b(t)$ dependent on the decisions using the nearby functions $b(t) \pm \epsilon$. The user can choose their own value of ϵ, referred to in the program as the tolerance, or use the pre-selected value of ϵ. The decisions made by the algorithm are reflected in Table 8.

If the user chooses to run the modified algorithm the program then compares the two answers produced. A re-run of the modified algorithm with a reduced tolerance, (pre-selected or of the user's own choice), is advised when appropriate. The user can elect whether or not to accept the advice.

4.1 Illustrative Examples

Example 4.1 *Input: period = 1, $b(t) = \frac{t(t-0.5)(t-1)}{1000}$*
The algorithm decides that the equation admits small solutions. Running the modified algorithm with the specified tolerance results in the advice to re-run the modified algorithm with a reduced tolerance. Re-running the modified algorithm with the tolerance reduced by a factor of 10 results in confirmation of the first decision. We can see from figure 6 that adjusting the function by a constant amount of 0.0001 will result in a function which does not change sign, hence the advise to reduce the tolerance.

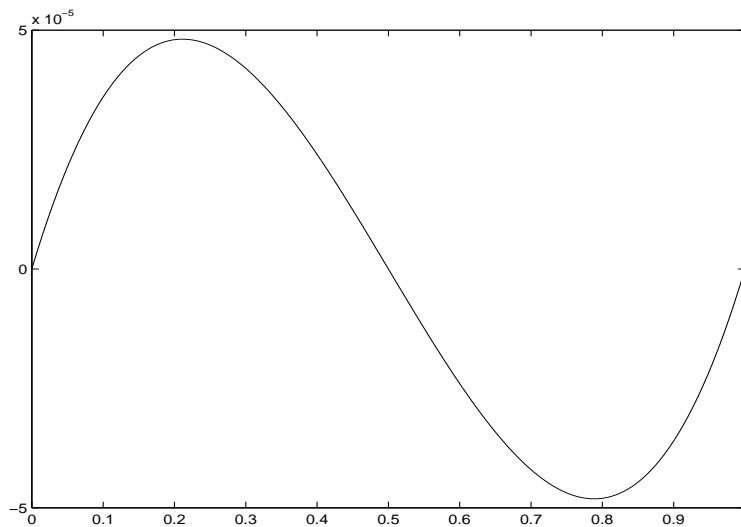Fig. 6. Graph of $b(t) = \frac{t(t-0.5)(t-1)}{1000}$ on $[0,1]$

Example 4.2 *Input: period = 4, $b(t) = t - 3.5$, the decision is that the equation admits small solutions.*
Input: period = 3, $b(t) = t - 3.5$, the decision is that the equation does not admit small solutions.
In this case $b(t)$ changes sign when t=3.5, hence when we force the period to equal 3 there is no change of sign.

16

Example 4.3 *In examples 4.1 and 4.2 the decision was easily predictable. If*

$$b(t) = \sin(\pi t) - e^{0.4t} + log(2.6t + 0.1) - \frac{t}{2 + 4t}$$

the decision is less obvious. The algorithm returns a decision that the equation admits small solutions. This result is confirmed by the graph of b(t) in figure 7.
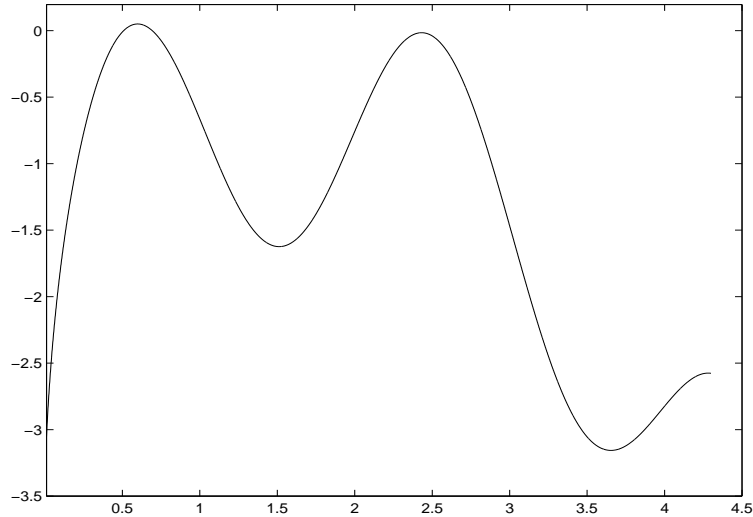

Fig. 7. Graph of $b(t) = \sin(\pi t) - e^{0.4t} + \log(2.6t + 0.1) - \frac{t}{2+4t}$ on $[0, 4.3]$

5   Summary

We have developed and tested an algorithm which automates the decision concerning the existence, or otherwise, of small solutions to the equation $x'(t) = b(t)x(t - p)$, with $b(t + p) = b(t)$. Consideration has been given to its reliability and any reservations about the decision are communicated to the user.

The algorithm extends immediately to the multi-delay equation of the form

$$x'(t) = \sum_{j=0}^{m} b_0(t)x(t - jw). \tag{14}$$

The mathematical justification for this extension is based on the Floquet theory which can be used to show that the underlying dynamics of the multi-delay equation are equivalent to those of a single delay equation (see, for example, [6]).

Good progress has also been made in the extension of the program to certain systems of delay equations of the form

$$y'(t) = A(t)y(t - 1). \tag{15}$$

However further work is needed before this can be fully automated.

6 Acknowledgements

The authors are pleased to acknowledge the helpful comments from the referees.

References

[1] K. L. Cooke, G. Derfel, On the Sharpness of a Theorem by Cooke and Verduyn Lunel, *J. Math. Anal. Appl.* **197** (1996), 379-391.

[2] O. Diekmann, S.A. van Gils, S. M. Verduyn Lunel, H.-O. Walther, *Delay Equations*, Springer Verlag, New York, 1995.

[3] Y. A. Fiagbedzi, Characterization of Small Solutions in Functional Differential Equations, *Appl. Math. Lett.* **10** (1997), 97-102.

[4] N. J. Ford and P. M. Lumb, *Numerical approaches to delay equations with small solutions*, E. A. Lipitakis(Ed), Proceedings of HERCMA 2001, 1, 101-108.

[5] N. J. Ford and P. M. Lumb, *Systems of delay equations with small solutions: A numerical approach*, In J. Levesley, I. J. Anderson & J. C. Mason (Eds), Algorithms for Approximation IV, University of Huddersfield, 2002, 94-101.

[6] N. J. Ford and P. M. Lumb, *Small solutions to periodic delay differential equations with multiple delays*, submitted for publication.

[7] N. J. Ford and P. M. Lumb, *Detecting small solutions for delay differential equations with delay and period commensurate: a numerical approach,* In preparation.

[8] N. J. Ford and S. M. Verduyn Lunel, *Characterising small solutions in delay differential equations through numerical approximations*, *Applied Mathematics and Computation*, **131** (2002) 253-270.

[9] N. J. Ford, S. M. Verduyn Lunel, *Numerical approximation of delay differential equations with small solutions*, Proceedings of 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, Lausanne 2000, paper 173-3, New Brunswick, 2000 (ISBN 3-9522075-1-9)

[10] N. J. Ford, *Numerical approximation of the characteristic values for a delay differential equation*, MCCM Numerical Analysis Report No 350, Manchester University 1999 (ISSN 1360 1725).

[11] J. K. Hale and S. M. Verduyn Lunel, *Introduction to Functional Differential Equations*, Springer Verlag, New York, 1993.

[12] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall\CRC,(2000).

[13] S. M. Verduyn Lunel, Small Solutions and Completeness for Linear Functional and Differential Equations, in John R. Graef, Jack K. Hale, editors, *Oscillation and Dynamics in Delay Equations*, American Mathematical Society, 1992.

c	$\alpha < 0.5$	$0.5 < \alpha < 1.0$	$1.0 < \alpha < 1.5$	$1.5 < \alpha < 2.5$	$2.5 < \alpha < 3.0$	$3 < \alpha \leq \pi$
-1.5	1 (1)	4 (4)	46 (40)	78 (84)	0 (0)	0 (0)
-1.4	1 (1)	4 (4)	48 (42)	76 (82)	0 (0)	0 (0)
-1.3	1 (1)	4 (4)	52 (44)	72 (80)	0 (0)	0 (0)
-1.2	1 (1)	4 (4)	60 (44)	64 (80)	0 (0)	0 (0)
-1.1	1 (1)	4 (4)	68 (48)	56 (76)	0 (0)	0 (0)
-1.0	4 (1)	6 (4)	74 (48)	45 (76)	0 (0)	0 (0)
-0.9	16 (1)	4 (4)	60 (48)	30 (76)	0 (0)	19 (0)
-0.8	24 (1)	4 (4)	62 (50)	12 (74)	14 (0)	13 (0)
-0.7	30 (1)	4 (4)	62 (52)	0 (72)	22 (0)	11 (0)
-0.6	28 (1)	14 (6)	50 (52)	0 (70)	28 (0)	9 (0)
-0.5	26 (1)	20 (6)	40 (54)	12 (68)	22 (0)	9 (0)
-0.4	26 (3)	26 (4)	30 (56)	18 (66)	20 (0)	9 (0)
-0.3	24 (3)	32 (4)	22 (60)	26 (62)	16 (0)	9 (0)
-0.2	20 (3)	38 (4)	16 (68)	28 (54)	20 (0)	7 (0)
-0.1	20 (5)	44 (2)	6 (78)	34 (44)	18 (0)	7 (0)
0	18 (1)	46 (0)	0 (0)	40 (128)	20 (0)	5 (0)
0.1	18 (1)	42 (0)	0 (0)	42 (126)	18 (2)	9 (0)
0.2	18 (1)	38 (0)	0 (0)	44 (126)	18 (2)	11 (0)
0.3	20 (1)	32 (0)	0 (0)	50 (126)	16 (2)	11 (0)
0.4	20 (1)	28 (0)	0 (0)	48 (126)	22 (2)	11 (0)
0.5	22 (1)	16 (0)	0 (0)	52 (126)	26 (2)	13 (0)
0.6	22 (1)	16 (0)	0 (0)	52 (126)	26 (2)	13 (0)
0.7	30 (1)	4 (0)	0 (0)	64 (126)	20 (2)	11 (0)
0.8	28 (1)	0 (0)	0 (0)	76 (126)	14 (2)	11 (0)
0.9	20 (1)	0 (0)	0 (0)	92 (126)	4 (2)	13 (0)
1.0	1(1)	0 (0)	0 (0)	123 (126)	2 (2)	3 (0)
1.1	1(1)	0 (0)	0 (0)	126 (126)	2 (2)	0 (0)
1.2	1(1)	0 (0)	0 (0)	126 (126)	2 (2)	0 (0)
1.3	1(1)	0 (0)	0 (0)	126 (126)	2 (2)	0 (0)
1.4	1(1)	0 (0)	0 (0)	126 (126)	2 (2)	0 (0)
1.5	1(1)	0 (0)	0 (0)	126 (126)	2 (2)	0 (0)

Table 6
The distribution of the magnitudes of the arguments of the eigenvalues, α, arising from discretisation of (2) and (3) with $b(t) = \sin 2\pi t + c$ for different values of c

	$b(t) = sin(2\pi t) + c$		$b(t) = t - 0.5 + c$		$b(t) = t(t - 0.5)(t - 1) + c$	
CV	$c = 1$		$c = \frac{1}{2}$		$c = \frac{\sqrt{3}}{36}$	
N	Actual	$\|Error\|$	Actual	$\|Error\|$	Actual	$\|Error\|$
32	1	0	0.46875000	0.03125	0.04806519	0.00004733
64	1	0	0.48437500	0.015625	0.04806519	0.00004733
96	1	0	0.48958333	0.01041667	0.04810475	0.00000777
128	1	0	0.49218750	0.0078125	0.04811239	0.00000013
160	1	0	0.49375000	0.00625	0.04811133	0.00000119

Table 7

Values of c at which the decision changes

NB. CV = the value of c which gives the critical function.

$b(t) - \epsilon$	$b(t)$	$b(t) + \epsilon$	Decision: Does the equation admit small solutions?	Re-run algorithm with a reduced tolerance?
Yes	Yes	Yes	Yes	
No/Near	Yes	Yes	Very Likely	
No	Yes	Yes	Very Likely	
Yes	Yes	No/Near	Very Likely	
Yes	Yes	No	Very Likely	
No/Near	Yes	No/Near	Likely	
No	Yes	No/Near	Likely	Yes
No/Near	Yes	No	Likely	Yes
No	Yes	No	Likely	Yes
Yes	No/Near	No	Unlikely	Possibly
No/Near	No/Near	No	Very Unlikely	
No/Near	No/Near	NoNear	Unlikely	
No/Near	No/Near	Yes	Unlikely	
Yes	No/Near	Yes	Very Unlikely	Yes
Yes	No/Near	No/Near	Very Unlikely	
No	No/Near	Yes	Very Unlikely	
No	No/Near	No/Near	Very Unlikely	
No	No/Near	No	Unlikely	Yes
No	No	No	No	No
No/Near	No	No	Very Unlikely	Yes
No	No	No/Near	Unlikely	Yes
Yes	No	No	Very Unlikely	Yes
No	No	Yes	Unlikely	Yes
Yes	No	No/Near	Unlikely	Yes
No/Near	No	No/Near	Very Unlikely	No
No/Near	No	Yes	Unlikely	Yes
Yes	No	Yes	Unlikely	Yes

Table 8
Decisions made using the modified algorithm