Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Journal of Computational and Applied Mathematics 185 (2006) 212–224
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

The role of conditioning in mesh selection algorithms for first order systems of linear two point boundary value problems☆

J.R. Cash[a,*], F. Mazzia[b], N. Sumarti[a], D. Trigiante[c]

[a]*Department of Mathematics, Imperial College, South Kensington, London SW7, England*
[b]*Dipartimento di Matematica, Università di Bari, via Orabona 4, I-70125 Bari, Italy*
[c]*Dipartimento di Energetica, via C. Lombroso 6/17, 50134 Firenze, Italy*

Received 20 June 2003

Abstract

Codes for the numerical solution of two-point boundary value problems can now handle quite general problems in a fairly routine and reliable manner. When faced with particularly challenging equations, such as singular perturbation problems, the most efficient codes use a highly non-uniform grid in order to resolve the non-smooth parts of the solution trajectory. This grid is usually constructed using either a pointwise local error estimate defined at the grid points or else by using a local residual control. Similar error estimates are used to decide whether or not to accept a solution. Such an approach is very effective in general providing that the problem to be solved is well conditioned. However, if the problem is ill conditioned then such grid refinement algorithms may be inefficient because many iterations may be required to reach a suitable mesh on which to compute the solution. Even worse, for ill conditioned problems an inaccurate solution may be accepted even though the local error estimates may be perfectly satisfactory in that they are less than a prescribed tolerance. The primary reason for this is, of course, that for ill conditioned problems a small local error at each grid point may not produce a correspondingly small global error in the solution. In view of this it could be argued that, when solving a two-point boundary value problem in cases where we have no idea of its conditioning, we should provide an estimate of the condition number of the problem as well as the

☆The works has been supported by GNCS(INDAM) and MIUR.
*Corresponding author.
E-mail addresses: j.cash@ic.ac.uk (J.R. Cash), mazzia@dm.uniba.it (F. Mazzia).

0377-0427/$ - see front matter
doi:10.1016/j.cam.2005.03.007

numerical solution. In this paper we consider some algorithms for estimating the condition number of boundary value problems and show how this estimate can be used in the grid refinement algorithm.

MSC: 65L05; 65L06; 65L20

Keywords: Conditioning; Two-point boundary value problems; Mesh selection

1. Mathematical preliminaries

Many methods have been proposed for the numerical solution of nonlinear systems of first order, two-point boundary value problems of the form

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y), \quad a \leqslant x \leqslant b, \quad g(y(a), y(b)) = 0. \tag{1}$$

Typically these methods attempt to control the error in the solution either by estimating the pointwise local error at the mesh points [7,6,14,16] or else by controlling the residual, that is the amount by which a continuous solution fails to satisfy the differential equation and the boundary conditions [9]. In both cases the codes attempt to control an estimate of the local error on the assumption that the problem is well conditioned so that if the local error in the solution is, in some sense small, then the global error will also be small. This all works perfectly well when the differential equation is well conditioned. However for stiff or ill conditioned problems, at least two major difficulties arise with such an approach. Firstly there is the problem that a small local error in the accepted solution does not necessarily mean that this solution has the required global accuracy. This can be demonstrated by a backward error analysis which shows that the error in the solution is bounded by the product of the local error and the condition number of the problem. Secondly, a mesh choosing algorithm which refines a mesh using a strategy based on estimates of the local error may do a very poor job of putting the mesh points in the correct places.

The concept of a well conditioned problem is a very intuitive one. Broadly speaking, a problem is said to be well conditioned if small changes in the functions f and g appearing in (1) produce correspondingly small changes in the solution obtained. In order to study the conditioning of (1), the standard approach is to examine the behaviour of the solutions of (1) in the neighbourhood of an isolated solution. This leads us to consider the linear equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = A(x)y + q(x), \quad a \leqslant x \leqslant b, \quad B_a y(a) + B_b y(b) = \beta. \tag{2}$$

If we assume that the boundary value problem (2) has a unique solution $y(x)$ then this solution is given by

$$y(x) = Y(x)Q^{-1}\beta + \int_a^b G(x, t)q(t)\,\mathrm{d}t. \tag{3}$$

Here $Y(x)$ is a fundamental solution of (2), $G(x, t)$ is the Green's function for (2) and the matrix Q, which is related to the boundary conditions, is defined as

$$Q = B_a Y(a) + B_b Y(b).$$

We see immediately from (3) that our assumption that a unique solution of (2) exists, is valid if and only if the matrix Q is non-singular. The structure of the solution (3) is important since it shows explicitly the dependence of the solution $y(x)$ on the boundary condition β and on the inhomogeneous term $q(x)$. As a result of this we can use Eq. (3) to investigate the effect of making small changes in the data of Eq. (2) and this in turn will lead to the formulation of our condition number estimate.

We start off our perturbation analysis by defining a solution $u(x)$ to satisfy the equation

$$\frac{\mathrm{d}u}{\mathrm{d}x} = A(x)u + q(x) + \delta(x), \quad B_a u(a) + B_b u(b) = \beta + \varepsilon \tag{4}$$

for a given function $\delta(x)$ and a given constant ε. From (3) and (4) it follows immediately that

$$u(x) - y(x) = Y(x)Q^{-1}\varepsilon + \int_a^b G(x,t)\delta(t)\,\mathrm{d}t. \tag{5}$$

Taking norms we have

$$\|u(x) - y(x)\| \leqslant \|Y(x)Q^{-1}\varepsilon\| + \|\int_a^b G(x,t)\delta(t)\,\mathrm{d}t\| \tag{6}$$

and the maximum value of the error is bounded by

$$\max_{a\leqslant x\leqslant b}\|u(x) - y(x)\| \leqslant K_1\|\varepsilon\| + K_2 \max_{a\leqslant x\leqslant b}\|\delta(x)\| \leqslant K \max\left(\|\varepsilon\|, \max_{a\leqslant x\leqslant b}\|\delta(x)\|\right), \tag{7}$$

where

$$K_1 = \max_{a\leqslant x\leqslant b}\|Y(x)Q^{-1}\|, \quad K_2 = \sup_x \int_a^b \|G(x,t)\|\,\mathrm{d}t$$

and

$$K = \max_{a\leqslant x\leqslant b}\left(\|Y(x)Q^{-1}\| + \int_a^b \|G(x,t)\|\,\mathrm{d}t\right). \tag{8}$$

In this way we can link the error in the perturbed solution $u(x)$, that is $u(x) - y(x)$, to the size of the perturbations $\delta(x)$ and ε in the differential equation and boundary conditions. If the constant K is appropriately small then the problem is well conditioned. Conditioning theory for linear two-point boundary value problems is well understood and is studied extensively in [2]. Our aim is not to solve ill conditioned problems accurately, as this is generally not possible, but instead is to detect which level of accuracy can be reached for a given problem. Important related work on computability with respect to a given tolerance is given in [10].

The important consequence of a well conditioned problem is that small $\|\varepsilon\|$ or $\max_{a\leqslant x\leqslant b}\|\delta(x)\|$ will give correspondingly small changes in the solution obtained. The converse is also clear in that large K_1 or K_2 can lead to large perturbations in the solution even for small $|\varepsilon|$ and $\max_{a\leqslant x\leqslant b}\|\delta(x)\|$. It is therefore crucial that when computing the numerical solution of a two-point boundary value problem we should also provide an estimate of the condition number. We consider this problem,which is a particularly challenging one, in the next section.

2. Approximation of the condition number of the discrete problem

In this section we consider the problem of estimating the condition number of the discrete problem obtained by applying a discretization algorithm to solve (1). In this discussion we will follow very closely the approach of Ascher [2]. In particular we will confine our attention to the case where a one-step numerical method is used to solve (1). Using a modified Newton method to solve the nonlinear algebraic equations arising at each step we obtain a linear system of algebraic equations of the form

$$Ax = b, \tag{9}$$

to solve for the approximate solution x. The matrix A appearing in (9) has the special structured form

$$\mathbf{A} = \begin{pmatrix} S_1 & R_1 & & & \cdots & & \\ & S_2 & R_2 & & & & \\ & & S_3 & R_3 & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & S_N & R_N \end{pmatrix}. \tag{10}$$

As we will see later, it is not the actual content of the matrices R_i and S_i that is important here but rather the almost block triangular structure of A. If we define the matrices M and D as

$$D = \operatorname{diag}(R_1^{-1}, R_2^{-1}, \ldots, R_N^{-1}, I), \tag{11}$$

$$\mathbf{M}^{-1} = \begin{pmatrix} G(x_1, x_2) & \cdots & G(x_1, x_{N+1}) & Y(x_1)Q^{-1} \\ \cdots & \vdots & \cdots & \vdots \\ \cdots & \vdots & \cdots & \vdots \\ G(x_{N+1}, x_2) & \vdots & G(x_{N+1}, x_{N+1}) & Y(x_{N+1})Q^{-1} \end{pmatrix}, \tag{12}$$

then Ascher, Mattheij and Russell prove the important result that

$$A^{-1} = M^{-1}D + O(h), \tag{13}$$

where the grid x_i chosen by the numerical method is such that $h_i = x_{i+1} - x_i$ and $h = \max_i h_i$. Ascher, Russell and Mattheij further show that

$$\|M^{-1}D\|_\infty = \max_i \left(\sum_{j=1}^{N} h_j \|G(x_i, x_{j+1})\|_\infty + O(h) + \|Y(x_i)Q^{-1}\|_\infty \right). \tag{14}$$

It now follows immediately from (11) and (12) that

$$\|A^{-1}\|_\infty \approx \max_{a \leqslant x \leqslant b} \left(\int_a^b \|G(x,t)\|_\infty \, \mathrm{d}t + \|Y(x)Q^{-1}\|_\infty \right). \tag{15}$$

The right-hand side of this expression is exactly the conditioning constant that appears in (6). It therefore follows that if we can estimate $\|A^{-1}\|_\infty$ then this will provide us with an estimate of the condition number K of the discrete problem. Note that one step numerical methods not only form A but also solve a linear system of algebraic equations, given by (9), with A as the coefficient matrix. This means that, not only do we have the matrix A available, we also have its LU factorization at hand. Several algorithms have been proposed for estimating $\|A^{-1}\|_\infty$ and we consider these in the next section.

3. Estimation of condition numbers

In this section we consider some codes which, as well as integrating general two-point boundary value problems of the form (1), also provide estimates of the condition number K appearing in Eq. (8). Although the basic algorithm used by all these codes to estimate K is based on an estimate of $\|A^{-1}\|_\infty$, the way in which they estimate this quantity and the way in which they interpret the results is different. The first algorithm we consider is one due to Shampine and Muir [15]. The starting point of their investigation was to consider the so called Bratu problem [8],

$$y'' = \lambda \exp(y), \quad y(0) = y(1) = 0, \tag{16}$$

using the two codes MIRKDC and Bvp4c that seek to control the residual of the solution. It is well known that for $\lambda = \lambda^* = 3.51383\ldots$ there is a unique solution to (16). It is also well known that for $\lambda < \lambda^*$ Eq. (16) has two solutions and for $\lambda > \lambda^*$ it has no solutions. Shampine and Muir applied their codes to the solution of (16) for the case $\lambda = 3.55$ (for this value of λ there is no solution) and both gave a numerical solution which looked perfectly reasonable with there being no indication that this solution was totally incorrect. In an attempt to explain this disturbing behaviour, Shampine and Muir suggested estimating $\|A^{-1}\|_\infty$ to see if the condition number associated with (16) is large. They did this by applying to A^{-T} an algorithm of Higham and Tisseur [11] which estimates the 1-norm of a matrix. What Shampine and Muir found was that as λ increased past λ^* there was a large increase in the estimated value of $\|A^{-1}\|_\infty$. For example, when the code Bvp4c, with a relative tolerance of 10^{-3} and an absolute tolerance of 10^{-6}, was applied to (16) with $\lambda = 3.45$ the condition number was estimated to be $3.4 \cdot 10^3$ and the solution was computed using 10 mesh points. However when (16) was solved with $\lambda = 3.55$ the code required 179 mesh points and the condition number estimate was 10^6. In this latter case the product of the condition number and the tolerance is $O(1)$ and, referring to (6)–(8), this would indicate that even though a 'solution' had been computed by the code there is no guarantee that it had any correct digits.

This technique of using the condition number estimate a posteriori could in theory be used with any one step integration method. The procedure would be to compute the numerical solution in the usual way and to provide an estimate of the condition number of the problem using the algorithm of Higham and Tisseur. If the condition number was larger than the inverse of the tolerance then this would be a warning that the solution may not be particularly accurate. In particular the global error in the solution may not be of the same order of magnitude as the local solution.

An obvious drawback with the above approach is that the numerical algorithm may be badly conditioned even though the original continuous problem may be perfectly well conditioned. With this problem in mind, an alternative way of estimating the condition number for a more restricted class of problems was proposed by Brugnano and Trigiante [4] and inserted in a general purpose solver for BVPs by Mazzia and Trigiante [13,14]. They considered the case where the problem to be solved is linear and homogeneous. In this case the system of algebraic equations corresponding to (9) takes the form

$$Ax = b, \tag{17}$$

where, because of the homogeneity of the problem being solved, the vector b depends only on the boundary conditions. To make their analysis slightly easier to formulate they set up the block matrix A so that the

boundary conditions appear only in the first row block of b. If we now define $G = A^{-1}$ and denote the elements of G by G_{ij} we can define a matrix Ω having elements

$$\Omega_{ij} = \|G_{ij}\|, \tag{18}$$

which is of size $(N+1) * (N+1)$. If the boundary conditions are now perturbed by a small amount $\delta\eta$ then there will be a corresponding perturbation δy in the solution satisfying

$$\|\delta y\| \leqslant \Omega_{*0}\|\delta\eta\|, \tag{19}$$

where Ω_{*0} denotes the first column of the matrix Ω. The advantage of this approach is that it is very cheap to implement, in that it requires only one block column of A^{-1} to be computed, with the disadvantage being that strictly speaking it is applicable only to homogeneous equations. However this does not stop us from applying this algorithm to inhomogeneous equations, so as to make some use of the conditioning information and we give an example of this in the numerical results section. The application of this algorithm to inhomogeneous equations, does have theoretical support if we solve a boundary value problem where the boundary conditions are appropriate for handling the decreasing and the increasing modes, using a numerical method that properly handles the exponential growth and decay of the solution. In this case the information provided by K_1 in the continuous case and by the first block column of the matrix A in the discrete case, is sufficient to handle inhomogeneous problems as well (see [2] Section 3.4.3 for the continuous case). Brugnano and Trigiante in [5], page 262 and [4] analyzed the nonhomogeneous case, when the function $q(x)$ is itself rapidly varying, and considered a procedure for handling the inhomogeneity. However this procedure has not yet been included in a general purpose code that solves BVPs and it would clearly be desirable to be able to handle inhomogeneous problems in a cheap and reliable way and we regard this as a topic for future research.

4. Mesh selection based on conditioning

Earlier in this paper we established the important result that, by estimating $\|A^{-1}\|$, it is possible to compute an estimate of the condition number of a problem. While most codes for boundary value problems do not provide such an estimate, it would in most cases be possible to incorporate into a code the algorithm of Higham and Tisseur. The question remains as to what we should do if the condition number of the matrix is large. The approach of Shampine and Muir was simply to warn of a large condition number, the implication being that the required accuracy may not be achieved. In contrast, Mazzia and Trigiante examined the possibility of building the information on conditioning into the mesh selection procedure. In what follows we will explain their approach.

Mazzia and Trigiante consider the numerical solution of the linear, homogeneous problem

$$\frac{\mathrm{d}y}{\mathrm{d}x} = L(x)y, \quad y \in \Re^m, \tag{20}$$

$$B_a y(a) + B_b y(b) = \eta \tag{21}$$

in the range $a \leqslant x \leqslant b$. If this boundary value problem has a unique solution then we can write this solution as

$$y(x) = Y(x)Q^{-1}\eta. \tag{22}$$

Appealing to (5) with $\delta x = 0$ it follows that if we make a perturbation $\delta\eta$ in the boundary condition then this will cause a perturbation δy in the solution which satisfies the bound

$$\|\delta y(x)\| \leqslant \|Y(x)Q^{-1}\|\|\delta\eta\|. \tag{23}$$

The approach of Mazzia and Trigiante is to choose the mesh so that the condition number of the discrete problem is similar to that of the continuous problem. Estimating the condition number of the continuous problem is, of course, the difficult bit and this is why we need to restrict our analysis to (20). In investigating the conditioning of (20) we consider two norms:

$$\|\delta y\|_\infty = \max_{a\leqslant x\leqslant b}\|\delta y(x)\| \quad \text{and} \quad \|\delta y\|_1 = \left(\int_a^b \|\delta y(x)\|\,\mathrm{d}x\right)\Big/(b-a). \tag{24}$$

Using these two definitions we can define two conditioning constants κ_c and γ_c which satisfy the inequalities

$$\|\delta y\|_\infty \leqslant \kappa_c\|\delta\eta\|, \quad \|\delta y\|_1 \leqslant \gamma_c\|\delta\eta\|, \tag{25}$$

where

$$\kappa_c = \max_{a\leqslant x\leqslant b}\|Y(x)Q^{-1}\|, \quad \gamma_c = \int_a^b \|Y(x)Q^{-1}\|\,\mathrm{d}x. \tag{26}$$

If these two quantities are both of moderate size we are dealing with a well conditioned problem. Conversely if both are large we have an ill conditioned problem. A rather different case is when only κ_c is large and γ_c is small. This means that the problem is ill conditioned using the maximum norm and well conditioned using the 1 norm. The problems that fall into this class are typically those possessing different time scales, for which the growth or decay rates of some fundamental solution modes are very rapid compared to others. Many singularly perturbed BVPs are in this class. Brugnano and Trigiante in [3,5], classified this class of problems as "stiff". The solution of a stiff problem is clearly difficult to obtain efficiently if we do not use an appropriate mesh and an appropriate numerical method. Usually problems with a large κ_c are considered ill-conditioned. Using information given by γ_c, we are able to further split this class into ill-conditioned and stiff problems, according to the size of γ_c.

This deals with the task of defining condition numbers for the continuous problem and we wish to choose our mesh so that the appropriate constants for the discrete problem are similar to these. Of course it will be a difficult task in general to compute these conditioning constants and there is a need to derive an algorithm which will allow us to do this.

If we now turn our attention to the discrete problem, the discrete solution vector y satisfies

$$My = e_1 \otimes \eta. \tag{27}$$

We saw in the previous section that a perturbation $\delta\eta$ in the boundary conditions produces a corresponding perturbation δy in the solution which satisfies

$$\|\delta y\| \leqslant \Omega_{*0}\|\delta\eta\|, \tag{28}$$

where $\|\delta y\| = (\|\delta y_0\|, \|\delta y_1\|, \ldots, \|\delta y_N\|)^{\mathrm{T}}$ and Ω_{*0} is the first column of the matrix Ω. We now introduce some discrete norms which are defined on the grid π in the following way:

$$\kappa_{\mathrm{d}}(\pi) = \max_i \Omega_{i0}, \quad \gamma_{\mathrm{d}}(\pi) = \left(\sum_{i=1}^{N} h_i \max(\Omega_{i-1,0}, \Omega_{i0}) \right) \Big/ (b-a). \tag{29}$$

The numbers $\kappa_{\mathrm{d}}(\pi)$, $\gamma_{\mathrm{d}}(\pi)$ are norms associated with the discrete problem (27) and these are the counterparts of the norms κ_{c}, γ_{c} which are associated with the continuous problem. The key point in the Mazzia and Trigiante approach is that they attempt to choose the mesh so that both the discrete and the continuous problems have condition numbers which are of the same order of magnitude. We note in particular that the constants $\kappa_{\mathrm{d}}(\pi)$ and $\gamma_{\mathrm{d}}(\pi)$ depend on the mesh and this is not the case for κ_{c}, γ_{c}. Our aim is to choose the mesh so that

$$\kappa_{\mathrm{c}} \approx \kappa_{\mathrm{d}}(\pi) \quad \text{and} \quad \gamma_{\mathrm{c}} \approx \gamma_{\mathrm{d}}(\pi). \tag{30}$$

In what follows we summarize the approach of Mazzia and Trigiante. The first important point to note is that the first column of G is an approximation to $Y(x)Q^{-1}$ to order of accuracy p, where p is the order of the integration formula being used. It now follows that

$$\begin{aligned} \gamma_{\mathrm{d}}(\pi) &= \left(\sum_{i=1}^{N} h_i \max(\Omega_{i-10}, \Omega_{i0}) \right) \Big/ (b-a) \\ &= \left(\sum_{i=1}^{N} h_i \max(\phi(t_{i-1}), \phi(t_i) \right) \Big/ (b-a) + O(h^p), \end{aligned}$$

where

$$\phi(x) = \|Y(x)Q^{-1}\|. \tag{31}$$

We now wish to choose the mesh that minimizes the error between the numerical quadrature formula and γ_{c}. That is we wish to minimize

$$E = \left| \sum_{i=1}^{N} h_i \max(\phi(t_{i-1}), \phi(t_i)) - \int_a^b \phi(t)\,\mathrm{d}t) \right| \Big/ (b-a)|. \tag{32}$$

It can be shown that E satisfies the bound

$$E = \frac{1}{b-a} \sum_{i=1}^{N} h_i (h_i \|\phi'(\xi_i)\| + O(h_i^2)), \tag{33}$$

where $\xi_i = x_i$ if $\phi(x_i) > \phi(x_{i-1})$ and $\xi_i = x_{i-1}$ otherwise. Since we only have discrete approximations to the function $\phi(x)$ we can further bound E as

$$E \leqslant \left(\sum_{i=1}^{N} h_i (|\omega_{i0} - \omega_{i-10}|) \right) \Big/ (b-a). \tag{34}$$

We therefore have the situation that we are seeking: to choose the mesh based on obtaining a desired accuracy and also on satisfying (30). Mazzia and Trigiante suggest a way of achieving this. While their approach works well we wish to describe their process in some generality since some of the components of their approach could be changed.

What Mazzia and Trigiante do is to define a monitor function based on both the local accuracy and on the conditioning estimate and they then use equidistribution based on this monitor function. Mazzia and Trigiante describe a code top order methods (TOM), which is based on TOM [1,13,14] and show that this gives very good results on some ill conditioned and stiff problems. Indeed the MATLAB code TOM is a very powerful one and we refer the interested reader to [1,12–14]. However we regard their ideas as a basis for an approach which is capable of extension. For example with nonlinear problems, or homogeneous problems, we could simply add points where the condition number is large and derive a monitor function based on this approach and on the control of local errors. We would also be interested to see how this approach extends to other classes of integration methods. We feel however that the approach of Mazzia and Trigiante is described in sufficient generality that we are able to apply it to other methods in a straightforward fashion. In the next section we will extend their approach to the code TWPBVP and, as we will see, the results are quite striking.

5. Numerical results

In this section we present some numerical results using the code TWPBVP updated using the conditioning parameters in the stepsize choosing algorithm. This code has some additional problems not present with the TOM code when solving ill conditioned or stiff problems because it is a deferred correction code and we must be very careful how these deferred corrections are applied. We do not claim that we have solved all of the implementation problems, in fact our algorithm is far from optimal but we do feel that our numerical results indicate clearly the way forward. In particular we will show that the conditioning constants play an important role in determining the accuracy of the solution and we feel that this adds weight to our belief that we should provide an estimate of the condition number of a problem when delivering a numerical solution.

It is important at this stage to recall exactly how the code TWPBVP operates. We recall that the deferred correction algorithm computes a fourth order solution using a mono-implicit Runge–Kutta method and then applies corrections of order 6 and 8 to get a more accurate solution. When estimating the condition number using the Mazzia–Trigiante approach it makes sense, while the conditions numbers are settling down, to confine our attention to the fourth order case since there is normally nothing to be gained by computing the deferred corrections. However once the condition numbers have settled down then we should compute the deferred corrections to get better accuracy by using the full power of the eighth order method. It is of course important to get the "timing" right since computing the deferred corrections too early will result in an ill conditioned problem being solved and if the deferred corrections are computed too late then extra work will be required since we will be working with a fourth order method. It is important that we understand exactly how deferred correction will behave when the problem is ill conditioned. In what follows we give some numerical results for Bratu's problem for two different values of λ (one where a solution does exist and one for where it does not) and for two problems where TWPBVP accepted a totally inaccurate solution when applied without modification. The problems we

consider are

(i) $y'' = \lambda e^y,\ \ y(0) = y(1) = 0.$
(ii) $\varepsilon y'' + xy' = 0,\ y(-1) = 0,\ y(1) = 2,\ \ \varepsilon = 10^{-6},\ \ \mathrm{Tol} = 10^{-6}.$
(iii) $\varepsilon y'' = y - \varepsilon\pi^2 \cos \pi x - \cos \pi x,\ y(-1) = 0,\ y(1) = 0,\ \ \varepsilon = 10^{-5},\ \ \mathrm{Tol} = 10^{-6}.$

For completeness we report also the numerical results obtained with the code TOM. The first point we wish to make is that for many of the test problems considered the code TWPBVP performs very well. However in some cases it performs extremely poorly because it does not detect that the problem is stiff or ill-conditioned. It is this flaw that we aim to rectify in this paper. The same happens for the code TOM when it is used with a mesh selection based only on the approximation of the error without using the conditioning parameters. In fact, one of the most important things that we noticed is that TOM and TWPBVP share many of the same problems when trying to solve ill conditioned boundary value problems. We first of all consider Bratu's problem [8]. We recall that the code Bvp4c, with optional input parameters, gives a 'solution' to this problem even when none exists. The purpose of our first set of results is to see whether the codes TWPBVP and TOM exhibit the same behaviour. We do this by performing exactly the same experiment described by Shampine and Muir in that we solve (16) using (a) $\lambda = 3.45$ for which the problem has a unique solution and (b) $\lambda = 3.55$ for which there is no solution. The results obtained were as follows. For $\lambda = 3.45$ with an initial mesh of 13 equally spaced points the code TWPBVP computed a symmetric solution to an accuracy of 10^{-6} on the given mesh. For the code TOM with an initial mesh of 11 equally spaced points (the number of stepsizes used by the code TOM must be a multiple of 5), relative tolerance 10^{-3} and absolute tolerance 10^{-6}, the computed solution was obtained with an accuracy of $7.8 \cdot 10^{-3}$ on the given mesh. The solver also gave a warning, that the conditioning parameters are not stabilized. This means that the solution could be inaccurate. With the same input parameter we forced the code to give the solution only when the conditioning parameters are stabilized. In this case the solution was obtained on a mesh of 41 points, with a maximum relative error of $1.5 \cdot 10^{-4}$, the conditioning parameters are $\kappa = 9.2$ and $\gamma = 8.2$. For the case where $\lambda = 3.55$ the Newton iteration scheme for TWPBVP failed to converge and successively doubled the mesh until it reached 385 points which exceeded the storage space allocated. Similar results were obtained with TOM using a relative tolerance of 10^{-3} and an absolute tolerance of 10^{-6}. If we try to compute a numerical solution using an increased absolute and relative tolerances, for example 10^{-2} for both the tolerances, without forcing the conditioning parameters to be stabilized, the solver finds a pseudo solution on the given mesh, with the warning that the conditioning parameters are not stabilized. If we force the solver to find also the conditioning parameters, it fails to give any solution. Similar results have been obtained on other problems having no/multiple solutions.

The second set of test results are for problems where the standard TWPBVP code performs very poorly. They are very difficult singular perturbation problems where the code 'does not have time' to recognize the poor conditioning of the problem. The two problems we consider are equations (ii) and (iii) above and the results are given in Tables 1 and 3. The most striking thing about the results presented in 1 and 3 is that the conditioning constants predict very accurately how the deferred corrections will behave. In particular we note that the deferred corrections give very little or no increase in accuracy while the condition numbers are still varying. Here the errors in the fourth order method are those obtained by subtracting the fourth and sixth order solutions and similarly for the sixth order error. Since the condition numbers are computed using the basic fourth order mono-implicit method, and in particular do not involve the

Table 1
Results for TWPBVP on problem (ii)

κ	γ	Error in fourth order	Error in sixth order
$0.18d+2$	$0.39d+1$	$0.32d-4$	$0.37d+0$
$0.36d+2$	$0.40d+1$	$0.22d-3$	$0.15d+1$
$0.71d+2$	$0.40d+1$	$0.12d-2$	$0.62d+1$
$0.14d+3$	$0.40d+1$	$0.27d-2$	$0.58d+1$
$0.25d+3$	$0.41d+1$	$0.14d-2$	$0.18d+2$
$0.38d+3$	$0.38d+1$	$0.38d-1$	$0.27d+2$
$0.40d+3$	$0.21d+1$	$0.17d-1$	$0.74d-1$
$0.40d+3$	$0.17d+1$	$0.57d-3$	$0.19d-1$
$0.40d+3$	$0.16d+1$	$0.14d-3$	$0.67d-2$
$0.40d+3$	$0.16d+1$	$0.80d-5$	$0.30d-5$

Table 2
Results for TOM on problem (ii)

κ	γ	Error in sixth order
$0.52d+1$	$0.28d+1$	$0.24d+1$
$0.31d+2$	$0.20d+1$	$0.86d-1$
$0.28d+3$	$0.15d+2$	$0.10d+2$
$0.19d+3$	$0.21d+2$	$0.90d+1$
$0.34d+3$	$0.62d+1$	$0.63d+1$
$0.40d+3$	$0.16d+1$	$0.32d-3$
$0.40d+3$	$0.16d+1$	$0.32d-3$
$0.40d+3$	$0.16d+1$	$0.10d-2$
$0.40d+3$	$0.15d+1$	$0.26d-3$
$0.40d+3$	$0.15d+1$	$0.53d-5$

deferred corrections in any way, the most efficient approach is to confine our attention to the fourth order method until the conditioning constants settle down and only then to compute the deferred corrections and test for convergence. The final point to make about these results is that the solution is computed to the requested accuracy which is not the case for the unmodified code TWPBVP. We also point out that the grid choosing algorithm clearly does the wrong thing in the sixth grid refinement for problem (iii) where the value of gamma increases from $0.13 \cdot 10^1$ to $0.17 \cdot 10^2$. We would anticipate that a successful mesh choosing algorithm would reject meshes which cause a large increase in the condition constants where they had previously shown signs of settling down to a limit and we regard this as (yet another) area for future investigation.

The results for the code TOM using the conditioning in the mesh selection are reported in Tables 2–4. The behaviour of the conditioning parameters is the same as TWPBVP. We see from the Tables of results that we have presented that the results obtained on these very difficult problems are satisfactory.

Table 3
Results for TWPBVP on problem (iii)

κ	γ	Error in fourth order	Error in sixth order
$0.320d+3$	$0.19d-3$	$0.82d+0$	$0.26d+2$
$0.318d+3$	$0.13d+3$	$0.69d+0$	$0.22d+2$
$0.317d+3$	$0.11d+3$	$0.60d+0$	$0.20d+2$
$0.317d+3$	$0.14d+2$	$0.27d+0$	$0.29d+0$
$0.317d+3$	$0.13d+1$	$0.36d-3$	$0.44d-3$
$0.317d+3$	$0.17d+2$	$0.95d-1$	$0.12d+0$
$0.317d+3$	$0.16d+1$	$0.79d-3$	$0.13d-3$
$0.317d+3$	$0.11d+1$	$0.67d-6$	$0.99d-7$

Table 4
Results for TOM on problem (iii)

κ	γ	Error in sixth order
$0.552d+3$	$0.12d+3$	$0.30d+1$
$0.317d+3$	$0.10d+2$	$0.20d+1$
$0.317d+3$	$0.15d+1$	$0.49d-2$
$0.317d+3$	$0.11d+1$	$0.68d-3$
$0.317d+3$	$0.10d+1$	$0.74d-4$
$0.317d+3$	$0.10d+1$	$0.12d-5$

6. Conclusion

The purpose of this paper was to give an overview of the role of conditioning in the solution of two-point boundary value problems. In particular we have shown by referring to previous work of Mazzia and Trigiante, how to estimate the condition number of a problem and how to incorporate this estimate into a mesh selection algorithm. We have also established the important concept that we need to wait until the condition constants have settled down before testing for accuracy and that this applies both to TOM and TWPBVP. Although these principles are now firmly established the precise way in which they should be applied is not clear. For this reason we do not regard our algorithm as being finished but we feel that a framework has now been established to allow us to tackle this problem. We hope that further research will settle these issues and that powerful codes which provide reliable condition number estimates, at least for an important subclass of problems, will soon be widely available.

Acknowledgements

The authors are grateful to the referee for several helpful comments.

References

[1] L. Aceto, F. Mazzia, D. Trigiante, On the performance of the code Tom on the Holt problem, modeling, simulation and optimization of integrated circuits, in: K. Antreich, R. Bulirsch, A. Gilg, P. Rentrop (Eds.), International Series of Numerical Mathematics, vol. 146, 2003, pp. 349–360.
[2] U.M. Ascher, R.M.M. Mattheij, R.D. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, SIAM, Philadelphia, 1995.
[3] L. Brugnano, D. Trigiante, On the characterization of stiffness for odes, Dyn. Continuous Discrete Impulsive Systems 2 (3) (1996) 317–335.
[4] L. Brugnano, D. Trigiante, A new mesh selection strategy for ODEs, Appl. Numer. Math. 24 (1) (1997) 1–21.
[5] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon & Breach, Amsterdam, 1998.
[6] J.R. Cash, G. Moore, R.W. Wright, An automatic continuation strategy for the solution of singularly perturbed linear two point boundary value problems, J. Comput. Phys. 122 (2) (1995) 266–279.
[7] J.R. Cash, M.H. Wright, A deferred correction method for nonlinear two point boundary value problems: implementation and numerical evaluation, SIAM J. Sci. Statist. Comput. 12 (4) (1991) 971–989.
[8] H.T. Davis, Introduction to Nonlinear Differential and Integral Equations, Dover, New York, 1962.
[9] W.H. Enright, P.H. Muir, Runge–Kutta software with defect control for boundary value ODEs, SIAM J. Sci. Comput. 17 (2) (1996) 479–497.
[10] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Introduction to adaptive methods for differential equations, Acta Numer. (1995) 105–158.
[11] N.J. Higham, F. Tisseur, A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra, SIAM J. Matrix Anal. 21 (2000) 1185–1201.
[12] F. Mazzia, I. Sgura, Numerical approximation of nonlinear BVPs by means of BVMs, Appl. Numer. Math. 42 (1–3) (2002) 337–352.
[13] F. Mazzia, D. Trigiante, A mesh selection strategy for boundary value problems, Report no. 36/2003, Dipartimento di Matematica, Università di Bari.
[14] F. Mazzia, D. Trigiante, A hybrid mesh selection strategy based on conditioning for boundary value ODE problems, Numer. Algorithms 36 (2) (2004) 169–187.
[15] L.F. Shampine, P.M. Muir, Estimating conditioning for BVPs of ODEs, Math. Comput. Model., 40(11-12) (2004) 1309–1321.
[16] R. Wright, J. Cash, G. Moore, Mesh selection for stiff two point boundary value problems, Numer. Algorithms 7 (1994) 205–224.