*European Society of Computational Methods in Sciences and Engineering (ESCMSE)*

*Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM)*
vol. 1, no. 1, 2006, pp. 81-90
ISSN 1790–8140

# Hybrid Mesh Selection Algorithms Based on Conditioning for Two-Point Boundary Value Problems

Jeff R. Cash* and Francesca Mazzia†

* *Dept. of Mathematics, Imperial College, South Kensington London SW7 2AZ*
† *Dipartimento di Matematica, Università di Bari, Via Orabona 4, I-70125 Bari (Italy)*

Received 13 December, 2005; accepted in revised form 6 January, 2006

*Abstract:* In this paper we demonstrate how hybrid mesh selection strategies based on conditioning can be used in codes designed for the numerical solution of singularly perturbed boundary value problems. The new mesh selection strategies are based on the estimation of two parameters which characterise the conditioning of the continuous problem as well as on standard estimates of the local discretization error. The code TOM was the first code where this kind of mesh selection was implemented. Subsequently, a different version of mesh selection, based on the same principles, was implemented in the well known code TWPBVP. We have now implemented a similar strategy in a deferred correction code based on Lobatto formulae and this has proved to be very efficient for stiff problems. The results obtained show that the modified code is often much more efficient than the original one which uses a standard mesh selection strategy. Furthermore the estimate of the conditioning of the problem, which is automatically provided by the code, gives information about the complexity of the differential equation being solved. This can be useful especially for singularly perturbed boundary value problems for which the complexity of the problem is related to a parameter which appears in the differential equation. Finally some numerical results are given to illustrate the greatly improved performance of the new algorithm.



*Keywords:* Boundary Value Problems, Mesh Selection, Conditioning

*Mathematics Subject Classification:* 65L10, 65L50

## 1 Introduction

In this paper we consider codes for the numerical solution of singularly perturbed boundary value ordinary differential equations having the general form

$$D(\epsilon)y'(x) = f(x,y), \quad a \le x \le b, \quad g(y(a), y(b)) = 0,$$

where $D(\epsilon)$ is a diagonal matrix whose elements depend linearly on $\epsilon$. Usually $D(\epsilon) = diag(1, \ldots, 1, \epsilon)$, $y$,$f$, $g \in R^m$ and $0 < \epsilon \ll 1$. When $\epsilon$ is small this class of problems is very difficult to solve because the solution exhibits narrow regions of very fast variation, such as boundary layer or turning

points for example. Codes for the numerical solution of this problem should therefore be able to automatically detect the regions of rapid variation and to put an appropriately fine mesh in these regions. Moreover, in the case of nonlinear problems, it is important to find a good initial guess to the solution which will lead to the rapid convergence of the modified Newton iteration method used to solve the discrete algebraic equations.

Recently hybrid mesh selection strategies based on conditioning have been introduced in [8, 13]. These new strategies are based on an estimation of two parameters which characterise the conditioning of the discrete problems and have been implemented in two codes for the solution of two point boundary value problems, namely TOM [13] and TWPBVPC [8]. The results obtained by these two codes show that the new mesh selection strategy makes the modified codes considerably more efficient than the originals, especially for singularly perturbed problems. Furthermore, as was explained in [8], it is necessary to have an estimate of the condition number if we are to establish a link between the local error, which we estimate, and the global error which we wish to control. In [8] an example is given where a condition number estimate is essential to stop a completely erroneous solution from being accepted.

To clarify these ideas we present a test problem and then solve it using TWPBVPC both with the standard mesh selection strategy based on the estimation of the local error and with the new mesh selection strategy which considers both conditioning and local error estimation.

**Example 1.** We examine the following problem [9] :

$$\begin{aligned} &\epsilon y'' + xy' = -\epsilon\pi^2\cos(\pi x) - \pi x\sin(\pi x), \\ &y(-1) = -2, y(1) = 0, \end{aligned} \tag{1}$$

whose exact solution is $\cos(\pi x)+\mathrm{erf}(x/\sqrt{2\epsilon})/\mathrm{erf}(1/\sqrt{2\epsilon})$, using the code TWPBVPC. The problem is first rewritten as a first order system with two components $(y, y')^T$ and the input parameters are $\epsilon = 10^{-7}$ and $tol(y) = 10^{-8}$. This means that we are seeking an approximation to the solution with an error $0.5 \cdot 10^{-8}$ or less in the $y$ component. The code gives a solution using a final mesh of 4192 points with a maximum error of $0.17 \cdot 10^{-8}$. The mesh sizes used in the intermediate steps of the computation are: 16, 31, 61, 121, 241, 481, 530, 1059, 1108, 2215, 2242, 4483, 8965, 19912, 4192. It seems that the code finds the problem very difficult to solve and it needs 19912 mesh points to get some worthwhile information about the solution. This is mainly due to the fact that the mesh selection algorithm adds extra points in the wrong place. If we use the same code with the mesh selection based on conditioning, the solution is obtained using 368 mesh points with maximum error of $0.85 \cdot 10^{-8}$. The meshes used are of sizes: 16, 31, 58, 85, 169, 368. The information given by the conditioning parameters allows the code to put the mesh points in the correct place and makes it considerably more efficient for the solution of this problem.

This behaviour suggests that implementing the same strategy in other codes designed for the numerical solution of singularly perturbed BVP could give similar improvement. This prompted us to investigate the possibility of adding a hybrid mesh selection strategy to the code TWPBVPL, which is based on Lobatto formulae, and which is an updated version of the code presented in [3]. This is the motivation behind the present paper. The important thing about TWPBVPL is the fact that it is a deferred correction formula where the deferred corrections are implicit. This allows the code to be very efficient for stiff problems but it does present us with difficulties in defining a hybrid mesh. This problem is examined in the present paper.

The way in which this research has evolved is that the code TOM was the first to use both conditioning and local error estimation in a mesh choosing algorithm. The theory behind this was presented in [13], [11]. This analysis was then refined and extended to the code TWPBVP which uses deferred correction based on MIRK formulae and which is suitable for mildly stiff problems. The main advance here was to extend the mesh selection techniques so as to be applicable to deferred correction algorithms. This has resulted in a code, TWPBVPC, being produced. Finally,

in the present paper, the main contribution is to extend these mesh choosing algorithms to Lobatto codes which have the novel property that the deferred corrections are implicit. In this case we now have one Newton iteration scheme which solves the discrete finite difference equations and another Newton iteration scheme which computes the deferred corrections. Our aim is to develop a new mesh selection strategy which is a reasonably direct extension of previous ideas but which performs well when implemented in our Lobatto deferred correction code. In what follows we will outline the contents of this paper.

In section 2 we describe how the conditioning parameters are computed in the continuous as well as in the discrete case. In section 3 we recall how the hybrid mesh selection was implemented in the codes TOM and TWPBVPC and in section 4 we describe how this was extended to the code TWPBVPL. Finally in section 5 we report some numerical results that show the behaviour of the code for both linear and nonlinear problems.

## 2 Conditioning parameters

To introduce the conditioning parameters which will be used in the mesh selection strategy of our Lobatto deferred correction code, let us consider for simplicity the following linear boundary value problem:

$$\frac{dy}{dx} = A(x)y + q(x), \;\; a \le x \le b, \quad B_a y(a) + B_b y(b) = \beta,$$

whose solution is given by

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

Here $Y(x)$ is a fundamental solution, $Q = B_a Y(a) + B_b Y(b)$ is non singular and $G(x,t)$ is the Green's function. Using the $\infty$-norm we can compute the conditioning parameter by considering a perturbed equation:

$$\frac{du}{dx} = A(x)u + q(x) + \delta(x), \;\; a \le x \le b, \quad B_a u(a) + B_b u(b) = \beta + e.$$

Here $\delta(x)$ and $e$ are small perturbations of the data. The difference between the two solutions satisfies:

$$||u(x) - y(x)|| \le ||Y(x)Q^{-1}e|| + ||\int_a^b G(x,t)\delta(t)dt||.$$

After some algebraic manipulation we obtain:

$$\max_{a\le x\le b} ||u(x) - y(x)|| \le \kappa_1 \|e\| + \kappa_2 \max_{a\le x\le b} ||\delta(x)||,$$

and

$$\max_{a\le x\le b} ||u(x) - y(x)|| \le \kappa \max(\|e\|, \max_{a\le x\le b} ||\delta(x)||),$$

where

$$\kappa_1 = \max_{a\le x\le b} ||Y(x)Q^{-1}||, \qquad \kappa_2 = \sup_{a\le x\le b} \int_a^b ||G(x,t)||dt,$$

and

$$\kappa = \max_{a\le x\le b} (||Y(x)Q^{-1}|| + \int_a^b ||G(x,t)||dt).$$

Following the same procedure as above and using the 1-norm we obtain:

$$\frac{1}{b-a}\int_a^b ||u(x)-y(x)||dx \le \gamma_1\|e\| + \gamma_2 \max_{a\le x\le b}||\delta(x)||,$$

and

$$\frac{1}{b-a}\int_a^b ||u(x)-y(x)||dx \le \gamma \max(\|e\|, \max_{a\le x\le b}||\delta(x)||),$$

where

$$\gamma_1 = \frac{1}{b-a}\int_a^b ||Y(x)Q^{-1}||dx, \qquad \gamma_2 = \frac{1}{b-a}\int_a^b\int_a^b ||G(x,t)||dtdx,$$

and

$$\gamma = \frac{1}{b-a}\int_a^b ((||Y(x)Q^{-1}|| + \int_a^b ||G(x,t)||dt))dx.$$

We know that if the conditioning parameters $\kappa$ and $\gamma$ are both of moderate size we are dealing with a well conditioned problem. If both are large we have an ill conditioned problem. Using two different norms it is possible to be presented with the case where $\kappa$ is large and $\gamma$ is small. 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 fall into this class. Brugnano and Trigiante [4, 5], using $\kappa_1$ and $\gamma_1$, classified this class of problems as "stiff" two point boundary value problems.

Problem (1) is a "stiff" problem; in fact $\kappa_1 \approx 2.5\cdot 10^3$, $\gamma_1 \approx 2.0$. The ratio between these two parameters, is often called the stiffness ratio, and is given by $\kappa_1/\gamma_1$. For our problem this ratio is $\approx 1.3\cdot 10^3$ and we consider this as being large. Moreover it is possible to compare the values of $\kappa_1$ and the value of $\kappa$. In this example $\kappa \approx 5.6\cdot 10^3$ and this is very close to $\kappa_1$. This means that the boundary conditions are appropriate for handling the decreasing and the increasing modes (see [2], sec. 3.4.2 and 3.4.3).

The main difficulty with the solution of this problem is not that it is extremely ill conditioned but that the mesh selection strategy is poor because it does not take conditioning into account.

## 3 Hybrid mesh selection strategies based on conditioning.

In order to use the conditioning parameters in a mesh selection strategy we first need to compute a discrete approximation to them which can be used in our numerical method. If in our algorithm we use a Newton iteration scheme to solve the nonlinear algebraic equations we need to solve a linear system of algebraic equations of the form $Mx = b$ for each iteration. The matrix $M$ depends on the numerical scheme and on the stepsize used. We use a grid $\pi = x_0, x_1, \ldots, x_N$ with stepsizes $h_i = x_i - x_{i-1}, i = 1, ..., N$ on which to solve the problem. The block matrix $M$ is set up so that the boundary conditions appear only in the first row block of $b$. Ascher, Mattheij and Russell [2] proved that for one-step schemes $\|M^{-1}\| \approx \kappa$. The computation of the first $m$ columns of $\|M^{-1}\|$ allows us to have an estimate of $\kappa_1$ and $\gamma_1$ (see [8, 11]).

We define the matrices $G = M^{-1}$ and $\Omega$ having elements $\Omega_{ij} = ||G_{ij}||$. The discrete conditioning parameters are defined on the grid $\pi$ in the following way:

$$\kappa_1(\pi) = \max_i \Omega_{i0}, \quad \gamma_1(\pi) = (\sum_{i=1}^{N} h_i \max(\Omega_{i-1,0}, \Omega_{i0}))/(b-a).$$

At present there exist two codes that implement hybrid mesh selection strategies, namely TOM and TWPBVPC. We need to structure our approach to mesh selection so that the strategy

developed in this paper shares exactly the same principles as those developed previously, namely: choose the mesh in order to have a discrete problem with conditioning parameters similar to those of the continuous problem. That is we wish to choose the mesh so that in some sense $\kappa_1(\pi) \approx \kappa_1$ and $\gamma_1(\pi) \approx \gamma_1$. The code TOM implements a hybrid mesh selection strategy which uses both the conditioning and the approximate global error in a dynamic way [13]. The solver TWPBVP has been updated and the new version [8], called TWPBVPC, implements a hybrid mesh selection strategy where we use information given by $\gamma_1(\pi)$, $\kappa_1(\pi)$ and the approximation of the local error. In both codes it is possible to choose between a standard mesh selection and a hybrid one.

In order to obtain an efficient mesh selection algorithm for TWPBVP, the procedure used in TOM was generalized. The two codes in fact use a very different strategy to compute the approximate solution of the continuous problem as we now explain.

TOM uses the Top Order Boundary Value Methods [6, 13] of order 2 and 6, and the conditioning parameters are associated with different matrices if the order changes. TWPBVPC starts with an order 4 mono-implicit Runge-Kutta method and tries to solve the generally nonlinear systems to compute approximations to the solution of order 4, 6, and 8 using a deferred correction procedure [10, 1]. The conditioning parameters used in the mesh selection are the ones associated with the order 4 method. In both codes, $\kappa_1(\pi)$ and $\gamma_1(\pi)$ are computed by solving $m$ linear systems for which the coefficient matrix has already been factorised. These two parameters and the vector $(\Omega_{00}, \cdots, \Omega_{N0})^T$ are essential to the new mesh selection algorithm and are computed each time we change the mesh. For the computation of $\kappa(\pi) = \|M^{-1}\|_\infty$ we use an algorithm that computes an estimate of the 1-norm of a matrix [12]. In contrast to the code TOM, in TWPBVPC the value of $\kappa(\pi)$ is not used in the mesh selection process. In order to improve the robustness of the code, we compute this value only at the end of the integration after the final solution has been computed. By comparing the values of $\kappa_1(\pi)$ and $\kappa(\pi)$ we can see whether the problem that has been solved has the correct dichotomy.

The discrete monitor function used by both codes when the conditioning is taken into account is:

$$\psi_{\gamma_1}(x_i) = |\Omega_{i0} - \Omega_{i-1,0}| + \alpha \tag{2}$$

where $\alpha = \frac{p}{(1-p)(b-a)} \sum_{i=1}^{N} |\Omega_{i0} - \Omega_{i-1,0}|$. The parameter $p$ gives information concerning the number of points that will be concentrated in the regions of small change. We choose $p/(1-p) = 0.08$ and this means that roughly 93 % of the points will be concentrated in the region of rapid change if the standard equidistribution algorithm is used. This monitor function is used dynamically together with information on the local error in both codes.

## 4 Mesh selection for TWPBVPL

The code TWPBVPL is a deferred correction code based on Lobatto IIIa formulae of order 4,6 and 8. This code is an extension of the one presented in [3], and this is considerably more robust than TWPBVP, especially for singularly perturbed boundary value problems. For this reason these deferred correction algorithms have also been implemented in the code ACDC [7] in a continuation framework, in order to deal with extremely stiff problems. The deferred correction scheme implementated in TWPBVPL has some similarities with that used in TWPBVPC, and as a result the hybrid mesh selection strategy derived for TWPBVPC can be implemented for TWPBVPL using a similar procedure. This has been described in detail in [8]. However since the underlying numerical schemes used in TWPBVPC and TWPBVPL are very different it is important to set up some empirical parameters in order to have an efficient mesh selection for the new code. In what follows we will describe some of these empirical parameters which have been chosen as a result of extensive numerical experimentation. In particular the monitor function is as defined by (2), but now the parameter $p/(1-p) = 10^{-5}$ is used. This means that roughly 99 % of the mesh points

will be concentrated in the regions of rapid change if the standard equidistribution algorithm is used. The two parameters $\kappa_1(\pi)$ and $\gamma_1(\pi)$ are considered as having become stabilised using the same criterion as was defined in [8] that is if they change by less than 5 % from one mesh to another. Moreover a problem is considered stiff if the ratio of the two parameters $\kappa_1(\pi)$ and $\gamma_1(\pi)$ satisfy $\kappa_1(\pi)/\gamma_1(\pi) > 100$. This definition of stiffness is different from that used by TWPBVPC. However it does make sense to impose this change since the code TWPBVPL is already able to solve quite difficult stiff problems in an efficient and reliable way. So it is appropriate not to change this algorithm too much unless the problem is extremely stiff (then it might be appropriate to use continuation). In fact this was one of the main problems we encountered in deriving our modified code. The code TWPBVPL, unlike TWPBVPC, is already efficient for quite a large class of singular perturbation problems and it is important not to destroy this good behaviour. (Recall in TWPBVPC a problem is considered stiff if $\kappa_1(\pi)/\gamma_1(\pi) > 10$).

As in TWPBVPC, the algorithm for adding and removing points in TWPBVPL is based on the following two quantities associated with the monitor function $\psi_{\gamma_1}$:

$$r_1 = \max_{i=1,\ldots,N}(\psi_{\gamma_1}(x_i)h_i),$$

and

$$r_2 = \sum_{i=1}^{N}(\psi_{\gamma_1}(x_i)h_i)/N.$$

Here $h_i$ refers to the mesh spacing on the current mesh. We decide to add additional mesh points when $\psi_{\gamma_1}(x_i)h_i$ is sufficiently large and in our code we take this as being when it is greater than $\max(0.65r_1, r_2)$, This parameter differs from the one used in TWPBVPC because usually the Lobatto schemes give us more reliable information concerning the regions where it is appropriate for points to be added. The number of mesh points to be added depends on the number of intervals in which this relation is satisfied and in this case we use the same strategy as is used in TWPBVPC only if $r_1 > 1$. In this case the mesh control procedure puts points in the region of rapid variation of the monitor function and also makes sure that not too many points are added at any stage. If $r_1$ is less than one, we are usually working with a good mesh and the error is small. In this case, the number of points to be added is restricted to 2.

All the other computational techniques used are very similar in both TWPBVPC and TWPBVPL, with the only major difference being that we remove points when $\psi_{\gamma_1}(x_i)h_i$ is less than $10^{-4}r_2$. The condition used in TWPBVPC has been changed because now it is possible to use a less conservative condition for removing points and this, usually, does not to destroy the information already acquired concerning $\kappa_1(\pi)$ and $\gamma_1(\pi)$.

For nonlinear problems the strategy used is the same as used in TWPBVPC, but now, if the Newton iteration scheme does not converge, the partially converged solution is considered stiff if $\kappa_1(\pi)/\gamma_1(\pi) > 10$.

## 5 Numerical Experiments

In this section we present some numerical experiments with the code TWPBVPL first using the classical mesh selection, which we denote by (TWPVPL(E)), and then using the hybrid mesh selection based on conditioning (which we denote by TWPBVPL(CE)). In order to make a fair comparison we have written a new code TWPBVPLC which has as an input parameter a logical variable asking whether conditioning is to be used or not. This means that the only difference between the two codes is that in one the conditioning is turned off and in the other the conditioning is used. The numerical experiments were carried out on an ALPHA server DS20E, with a 667MHz EV67 processor using the Digital Fortran compiler. For each problem we use an error tolerance

which is the same for all components, and we ran the codes for different values of this tolerance. For a problem for which we know the exact solution we compute the error using the following formula:

$$\max_{0\le i\le N}\left(\frac{|y(x_i)-y_\pi(x_i)|}{(1+|y(x_i)|)}\right), \tag{3}$$

where $x_i, i = 1, N$ are the mesh points in the final mesh, $y_\pi$ represents the numerical solution on the mesh and $y$ is the true solution of the differential equation. If we are dealing with a second order differential equation we convert it to first order form before we solve it since our codes are written to be applicable to first order systems only.

We have run the code on the challanging test problems described in [9] and in what follows we report the results on a representative subset of problems.

### 5.1 Test Problems

**Problem 1**

$$\epsilon y'' + xy' = -\epsilon\pi^2\cos(\pi x) - \pi x\sin(\pi x), \qquad y(-1) = -2, y(1) = 0.$$

This problem is a linear singularly perturbed problem. It has been chosen because, for $0 < \epsilon \ll 1$, the solution has a turning point at $x = 0$. The exact solution is $y(x) = \cos(\pi x) + \exp((x-1)/\sqrt{\epsilon}) + \exp(-(x+1)/\sqrt{\epsilon})$.

**Problem 2**

$$\epsilon y'' + \exp(y)y' - \frac{\pi}{2}\sin(\frac{\pi x}{2})\exp(2y) = 0, \quad y(0) = 0, y(1) = 0,$$

This problem has a boundary layer at $x = 0$. We use as an initial guess 0 for $y$ and $y'$.

**Problem 3**

$$\epsilon y'' + (y')^2 = 1, \quad y(0) = 1 + \epsilon\ln\cosh(-0.745/\epsilon), y(1) = 1 + \epsilon\ln\cosh(0.255/\epsilon).$$

The true solution is $1 + \epsilon\ln\cosh((x - 0.745)/\epsilon)$. We use as initial guess 0 for $y$ and $y'$.

**Problem 4**

$$\epsilon y'' + yy' - y = 0, \quad y(0) = 1, y(1) = 1/3.$$

The solution has a boundary layer at $x = 0$ and a corner layer at $x = 2/3$. We use as an initial guess 0.5 for $y$ and $y'$.

### 5.2 Discussion of the results

We consider first the results for Problem 1 (see Table 1). This is a linear singularly perturbed differential equation. In this case the deferred correction code TWPBVPL was already very efficient. The results are comparable, but we now have more information about the problem. It is well known in fact that the complexity of the problem changes with $\epsilon$. In this case the conditioning parameter $\kappa_1$ changes like $\sqrt{(25/40)\epsilon^{-1}}$ and the value of $\gamma_1 \approx 2$ remains constant with respect to $\epsilon$.

For nonlinear problems (see Tables 2, 3 and 4) the information given by the conditioning parameters normally allows the convergence of the nonlinear schemes using a small number of mesh points, especially when $\epsilon$ becomes small and the convergence becomes more difficult to achieve (a * in the table means that the code was not able to solve the problem using less than 50000 mesh points). Also in this case it is very easy to find a relation between $\kappa_1$ and $\epsilon$, in fact $\kappa_1 \approx 1.25/\epsilon$ for Problem 2 and $\kappa_1 \approx 1/\epsilon$ for Problems 3 and 4. The values of $\gamma_1$ remain constant with respect to $\epsilon$. Moreover for all the test problems the values of $\kappa$ and $\kappa_1$ are very close, showing that the problems have the correct dichotomy.

Table 1: Results for Problem 1.

| | | TWPBVPL(CE) | | | | | | TWPBVPL(E) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | Tol | Error | $N_{max}$ | $\kappa_1$ | $\gamma_1$ | $\kappa$ | Time | Error | $N_{max}$ | Time |
| $10^{-5}$ | $10^{-4}$ | 0.34e-6 | 89 | 0.25e3 | 0.22e1 | 0.51e3 | 0.293e-2 | 0.34e-6 | 89 | 0.195e-2 |
| | $10^{-6}$ | 0.76e-9 | 171 | 0.25e3 | 0.21e1 | 0.51e3 | 0.390e-2 | 0.76e-9 | 171 | 0.390e-2 |
| | $10^{-8}$ | 0.31e-11 | 360 | 0.25e3 | 0.21e1 | 0.51e3 | 0.781e-2 | 0.31e-11 | 360 | 0.683e-2 |
| $10^{-7}$ | $10^{-4}$ | 0.34e-6 | 149 | 0.25e4 | 0.22e1 | 0.51e4 | 0.488e-2 | 0.34e-6 | 160 | 0.390e-2 |
| | $10^{-6}$ | 0.18e-8 | 319 | 0.25e4 | 0.21e1 | 0.51e4 | 0.976e-2 | 0.18e-8 | 335 | 0.878e-2 |
| | $10^{-8}$ | 0.985e-11 | 709 | 0.25e4 | 0.21e1 | 0.51e4 | 0.195e-1 | 0.99e-11 | 735 | 0.185e-1 |
| $10^{-9}$ | $10^{-4}$ | 0.16e-6 | 175 | 0.25e5 | 0.21e1 | 0.51e5 | 0.586e-2 | 0.34e-6 | 296 | 0.586e-2 |
| | $10^{-6}$ | 0.27e-9 | 732 | 0.25e5 | 0.21e1 | 0.50e5 | 0.303e-1 | 0.18e-8 | 591 | 0.137e-1 |
| | $10^{-8}$ | 0.17e-11 | 1692 | 0.25e5 | 0.21e1 | 0.51e5 | 0.586e-1 | 0.99e-11 | 1562 | 0.420e-1 |
| $10^{-11}$ | $10^{-4}$ | 0.39e-7 | 522 | 0.25e6 | 0.22e1 | 0.51e6 | 0.146e-1 | 0.39e-7 | 615 | 0.156e-1 |
| | $10^{-6}$ | 0.13e-10 | 1221 | 0.25e6 | 0.21e1 | 0.51e6 | 0.303e-1 | 0.18e-10 | 1576 | 0.351e-1 |
| | $10^{-8}$ | 0.53e-12 | 1649 | 0.25e6 | 0.20e1 | 0.51e6 | 0.478e-1 | 0.14e-12 | 2305 | 0.586e-1 |
| $10^{-13}$ | $10^{-4}$ | 0.18e-7 | 441 | 0.25e7 | 0.21e1 | 0.51e7 | 0.137e-1 | 0.35e-9 | 685 | 0.176e-1 |
| | $10^{-6}$ | 0.15e-9 | 881 | 0.25e7 | 0.20e1 | 0.51e7 | 0.264e-1 | 0.353e-10 | 1369 | 0.351e-1 |
| | $10^{-8}$ | 0.21e-11 | 3521 | 0.25e7 | 0.20e1 | 0.51e7 | 0.102e0 | 0.25e-12 | 5473 | 0.139e0 |
| $10^{-15}$ | $10^{-4}$ | 0.26e-6 | 461 | 0.25e8 | 0.21e1 | 0.51e8 | 0.146e-1 | 0.48e-7 | 793 | 0.215e-1 |
| | $10^{-6}$ | 0.14e-8 | 1841 | 0.25e8 | 0.20e1 | 0.51e8 | 0.537e-1 | 0.12e-8 | 3169 | 0.820e-1 |
| | $10^{-8}$ | 0.43e-9 | 3681 | 0.25e8 | 0.20e1 | 0.51e8 | 0.106e0 | 0.20e-9 | 6337 | 0.163e0 |

Table 2: Results for Problem 2.

| | | TWPBVPL(CE) | | | | | TWPBVPL(E) | |
|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | Tol | $N_{max}$ | $\kappa_1$ | $\gamma_1$ | $\kappa$ | Time | $N_{max}$ | Time |
| $10^{-3}$ | $10^{-4}$ | 65 | 0.125e4 | 0.24e1 | 0.17e4 | 0.781e-2 | 87 | 0.98e-2 |
| | $10^{-6}$ | 92 | 0.125e4 | 0.23e+1 | 0.17e4 | 0.781e-2 | 86 | 0.107e-1 |
| | $10^{-8}$ | 167 | 0.125e4 | 0.23e1 | 0.17e4 | 0.19e-1 | 155 | 0.13e-1 |
| $10^{-4}$ | $10^{-4}$ | 145 | 0.125e5 | 0.28e1 | 0.17e5 | 0.273e-1 | 303 | 0.400e-1 |
| | $10^{-6}$ | 145 | 0.125e5 | 0.25e1 | 0.17e5 | 0.283e-1 | 303 | 0.439e-1 |
| | $10^{-8}$ | 242 | 0.125e5 | 0.24e1 | 0.17e5 | 0.381e-1 | 303 | 0.547e-1 |
| $10^{-5}$ | $10^{-4}$ | 395 | 0.126e6 | 0.31e1 | 0.17e6 | 0.839e-1 | 3787 | 0.562e0 |
| | $10^{-6}$ | 395 | 0.125e6 | 0.25e1 | 0.17e6 | 0.878e-1 | 3787 | 0.573e0 |
| | $10^{-8}$ | 435 | 0.125e6 | 0.23e1 | 0.17e6 | 0.107e0 | 3787 | 0.578e0 |
| $10^{-6}$ | $10^{-4}$ | 1734 | 0.125e7 | 0.24e1 | 0.17e7 | 0.305e0 | 3289 | 0.461e0 |
| | $10^{-6}$ | 1734 | 0.125e7 | 0.24e1 | 0.17e7 | 0.327e0 | 6577 | 0.761e0 |
| | $10^{-8}$ | 1734 | 0.125e7 | 0.23e+1 | 0.17e7 | 0.364e0 | 6577 | 0.817e0 |
| $10^{-7}$ | $10^{-4}$ | 3177 | 0.125e8 | 0.25e1 | 0.17e8 | 0.793e0 | * | * |
| | $10^{-6}$ | 3179 | 0.125e8 | 0.23e1 | 0.17e8 | 0.49e0 | * | * |
| | $10^{-8}$ | 6257 | 0.125e8 | 0.22e+1 | 0.17e8 | 0.192e1 | * | * |

## 6 Conclusion

The hybrid strategy for the mesh selection used in the code TWPBVPLC has been found to work efficiently for both linear and nonlinear problems. Moreover the information concerning the conditioning gives more insight into the problem being solved. The very good results obtained using TWPBVPLC, show clearly that it is important to use the conditioning parameters when implementing a mesh choosing algorithm. What we have attempted to do is to change the mesh selection algorithm in the two deferred correction codes TWPBVP and TWPBVPL so that in both codes the mesh selection algorithms are based on very similar principles. This has resulted in the completion of two efficient codes, namely TWPBVPLC and TWPBVPC which are available on the web page of one of the authors [9]. In order to solve extremely difficult singular perturbation problems it is normally necessary to resort to continuation. Two efficient continuation codes ACDC and COLMOD are already available [9] and our aim was to change these so that the mesh selection algorithm considered conditioning as well as local error estimation. However this has proved to be

Table 3: Results for Problem 3.

| | | TWPBVPL(CE) | | | | | | TWPBVPL(E) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | Tol | Error | $N_{max}$ | $\kappa_1$ | $\gamma_1$ | $\kappa$ | Time | Error | $N_{max}$ | Time |
| $10^{-1}$ | $10^{-4}$ | 0.14e-5 | 10 | 0.91e1 | 0.34e1 | 0.14e2 | 0.976e-3 | 0.14e-5 | 10 | 0.976e-3 |
| | $10^{-6}$ | 0.35e-8 | 20 | 0.10e2 | 0.30e1 | 0.15e2 | 0.195e-2 | 0.35e-8 | 20 | 0.976e-3 |
| | $10^{-8}$ | 0.78e-11 | 43 | 0.10e2 | 0.27e1 | 0.15e2 | 0.195e-2 | 0.78e-11 | 43 | 0.293e-2 |
| $10^{-2}$ | $10^{-4}$ | 0.18e-5 | 37 | 0.10e3 | 0.40e1 | 0.15e3 | 0.390e-2 | 0.18e-5 | 37 | 0.488e-2 |
| | $10^{-6}$ | 0.85e-9 | 160 | 0.10e3 | 0.35e1 | 0.15e3 | 0.234e-1 | 0.34e-11 | 307 | 0.390e-1 |
| | $10^{-8}$ | 0.26e-10 | 72 | 0.10e3 | 0.31e1 | 0.15e3 | 0.878e-2 | 0.26e-10 | 72 | 0.683e-2 |
| $10^{-3}$ | $10^{-4}$ | 0.39e-7 | 649 | 0.98e3 | 0.44e1 | 0.15e4 | 0.309e0 | 0.79e-9 | 4033 | 0.172e1 |
| | $10^{-6}$ | 0.31e-10 | 649 | 0.10e4 | 0.31e1 | 0.15e4 | 0.310e0 | 0.14e-10 | 4033 | 0.174e1 |
| | $10^{-8}$ | 0.89e-13 | 649 | 0.10e4 | 0.31e1 | 0.15e4 | 0.311e0 | 0.51e-13 | 4033 | 0.176e1 |

Table 4: Results for Problem 4.

| | | TWPBVPL(CE) | | | | | TWPBVPL(E) | |
|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | Tol | $N_{max}$ | $\kappa_1$ | $\gamma_1$ | $\kappa$ | Time | $N_{max}$ | Time |
| $10^{-3}$ | $10^{-4}$ | 61 | 0.10e4 | 0.27e1 | 0.10e4 | 0.586e-2 | 87 | 0.586e-2 |
| | $10^{-6}$ | 88 | 0.10e4 | 0.25e1 | 0.10e4 | 0.683e-2 | 87 | 0.878e-2 |
| | $10^{-8}$ | 121 | 0.10e4 | 0.25e1 | 0.10e4 | 0.976e-2 | 151 | 0.137e-1 |
| $10^{-4}$ | $10^{-4}$ | 181 | 0.10e5 | 0.31e1 | 0.10e5 | 0.127e-1 | 303 | 0.293e-1 |
| | $10^{-6}$ | 181 | 0.10e5 | 0.27e1 | 0.10e5 | 0.156e-1 | 303 | 0.322e-1 |
| | $10^{-8}$ | 181 | 0.10e5 | 0.25e1 | 0.10e5 | 0.185e-1 | 303 | 0.361e-1 |
| $10^{-5}$ | $10^{-4}$ | 537 | 0.10e6 | 0.28e1 | 0.10e6 | 0.478e-1 | 1167 | 0.114e0 |
| | $10^{-6}$ | 537 | 0.10e6 | 0.27e1 | 0.10e6 | 0.498e-1 | 1181 | 0.117e0 |
| | $10^{-8}$ | 537 | 0.10e6 | 0.25e1 | 0.10e6 | 0.595e-1 | 1181 | 0.121e0 |
| $10^{-6}$ | $10^{-4}$ | 1711 | 0.10e7 | 0.33e1 | 0.10e7 | 0.144e0 | 4637 | 0.503e0 |
| | $10^{-6}$ | 1711 | 0.10e7 | 0.28e1 | 0.10e7 | 0.154e0 | 4637 | 0.515e0 |
| | $10^{-8}$ | 1711 | 0.10e7 | 0.26e1 | 0.10e7 | 0.162e0 | 4637 | 0.522e0 |
| $10^{-7}$ | $10^{-4}$ | 5345 | 0.10e8 | 0.36e1 | 0.10e8 | 0.486e0 | 18475 | 0.190e1 |
| | $10^{-6}$ | 5345 | 0.10e8 | 0.28e1 | 0.10e8 | 0.501e0 | 18475 | 0.192e1 |
| | $10^{-8}$ | 5345 | 0.10e8 | 0.25e1 | 0.10e8 | 0.522e0 | 18475 | 0.197e1 |
| $10^{-8}$ | $10^{-4}$ | 18027 | 0.10e9 | 0.26e1 | 0.10e9 | 0.549e1 | * | * |
| | $10^{-6}$ | 18027 | 0.10e9 | 0.26e1 | 0.10e9 | 0.561e1 | * | * |
| | $10^{-8}$ | 18027 | 0.10e9 | 0.26e1 | 0.10e9 | 0.570e1 | * | * |

much more difficult than anticipated and we defer this to a future time. Finally we note that, as well as making the codes much more efficient, the computation of information concerning conditioning is vital if we are to estimate the global error in the approximation [8]. As was explained in that paper we expect the next generation of boundary value codes to routinely provide information about the conditioning so that we can have some confidence in the global accuracy of the solution produced.

## 7 Acknowledgement.

This work was partially supported by G.N.C.S. (INdAM) and COFIN-PRIN 2004 (project "Metodi numerici e software matematico per le applicazioni").

## References

[1] J. R. Cash and 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):971–989, 1991.

[2] Uri M. Ascher, Robert M. M. Mattheij and Robert D. Russell. *Numerical solution of boundary value problems for ordinary differential equations*, volume 13 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995.

[3] Z. Bashir-Ali, J. R. Cash, and H. H. M. Silva. Lobatto deferred correction for stiff two-point boundary value problems. *Comput. Math. Appl.*, 36(10-12):59–69, 1998.

[4] L. Brugnano and D. Trigiante. On the characterization of stiffness for odes. *Dynamics of Continuous, Discrete and Impulsive Systems*, 2(3):317–335, 1996.

[5] L. Brugnano and D. Trigiante. A new mesh selection strategy for odes. *Appl. Numer. Math.*, 24:1–21, 1997.

[6] L. Brugnano and D. Trigiante. *Solving Differential Problems by Multistep Initial and Boundary Value Methods*. Gordon & Breach,Amsterdam, 1998.

[7] J. Cash, G. Moore, and R. Wright. An automatic continuation strategy for the solution of singularly perturbed nonlinear boundary value problems. *ACM Transaction on Mathematical Software*, 27(2):245–266, 2001.

[8] J. R. Cash and F. Mazzia. A new mesh selection algorithm, based on conditioning, for two-point boundary value codes. *J. Comput. Appl. Math.*, 184(2):362–381, 2005.

[9] J. R. Cash and Margaret H. Wright. Bvp software. `http://www.ma.ic.ac.uk/~jcash/BVP_software/readme.html`.

[10] J.R. Cash. On the numerical integration of nonlinear two-point boundary value problems using iterated deferred corrections. II: The development and analysis of highly stable deferred correction formulae. *SIAM J. Numer. Anal.*, 25(4):862–882, 1988.

[11] J.R. Cash, F. Mazzia, N. Sumarti, and D. Trigiante. The role of conditioning in mesh selection algorithms for first order systems of linear two point boundary value problems. *J. Comput. Appl. Math.*, 185(2):212–224, 2006.

[12] Nicholas J. Higham. FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation. *ACM Trans. Math. Software*, 14(4):381–396 (1989), 1988.

[13] Francesca Mazzia and Donato Trigiante. A hybrid mesh selection strategy based on conditioning for boundary value ODE problems. *Numer. Algorithms*, 36(2):169–187, 2004.