A mesh selection strategy for boundary value problems

Francesca Mazzia, Donato Trigiante

Rapporto n. 36/2003

Dipartimento di Matematica
Università degli studi di Bari
Via E. Orabona 4
70125 BARI (ITALIA)

A mesh selection strategy for boundary value problems *

Francesca Mazzia [a] Donato Trigiante [b]

[a] *Dipartimento di Matematica, via Orabona 4, 70125 Bari, Italy*
E-mail: mazzia@dm.uniba.it
[b] *Dipartimento di Energetica, Via C. Lombroso 6/17, 50134 Firenze, Italy*
E-mail: trigiant@unifi.it

An appropriate mesh selection strategy is one of the fundamental tools in designing robust codes for differential problems, especially if the codes are required to work for difficult multi scale problems. Most of the existing codes base the mesh selection on an estimate of the error (or the residual). Our strategy, based on the estimation of two parameters characterizing the conditioning of the continuous problem, as well as on an estimate of the error, not only permits us to obtain a *well adapted,* thus reducing the cost of the code, but also provides a measure of the conditioning of both continuous and discrete problems.

Keywords: boundary value problems, singularly perturbation problems, mesh selection, equidistribution

AMS Subject classification: 65L10, 65L50

1. Introduction

In this paper we will be concerned with the numerical solution of nonlinear two-point boundary value problems (BVPs)

$$\begin{aligned} &y' = f(t,y), \quad t_0 \le t \le T, \quad g(y(t_0), y(T)) = 0, \\ &y : [t_0, T] \to R^m, \\ &f : R \times R^m \to R^m, \qquad g : R^m \times R^m \to R^m. \end{aligned} \tag{1}$$

One of the most interesting aspects associated with the numerical solution of BVPs is to find a step size variation strategy that allows the solution of difficult

* work supported by G.N.C.S. INdAM

problems with the minimum effort. Usually the performance of codes for the numerical solution of BVPs depends critically on the determination of a suitable adapted mesh. Many attempts have been proposed, each of which has improved single facets of the problem. For example, by improving the technique of approximation of the error (or the residual) [8,9] or by adopting new mesh selection strategies [11], or using the codes in a homotopic continuation framework [5]. Our opinion is that a substantial improvement will be made when the mesh is chosen by considering the conditioning of the problem. We assume the principle that both the discrete and the continuous problems should share the same order of magnitude of the conditioning parameters. However, usually we have no information about the conditioning of the discrete problem and its relation with the conditioning of the continuous one.

In [2,4] two quantities measuring the conditioning of the continuous problem, along with the corresponding ones for the discrete problem, have been defined. They have been used to define a monitor function for the step size variation strategy. The main objective of this paper is to investigate a modification of the step size variation described in [4] showing that it is particularly suited for stiff problems. We have inserted the refined strategy into a new MATLAB code, called TOM, based on a class of Boundary Value Methods [3] and on a quasi-linearization strategy. Here we discuss only the mesh selection strategy, while the study and the description of the nonlinear technique has been discussed in [10].

In sections 2 and 3 we recall the definitions of the conditioning of both continuous and discrete problems together with a definition of stiffness for boundary value problems. In section 4 we describe the error estimation strategy, in section 5 the mesh selection strategy and in section 6 the technique used for the solution of nonlinear problems. In section 7 we show the effectiveness of the mesh selection strategy by presenting some numerical results for stiff problems. We compare the mesh selection based on the conditioning parameter and on the error, presented in section 5, with a mesh selection based only on the error. In order to have a comparison with existing codes, based on standard mesh selection strategies, we also provide the corresponding results from BVP4c [9] included in the MATLAB 6.5, release 13 distribution, although we are aware, as pointed out by the authors, that it is not especially designed for stiff problems.

2. Conditioning of continuous problems

To define the conditioning parameter associated to the continuous problem we consider a linear BVP of the form

$$\begin{aligned} y' &= L(t)y \qquad y \in R^m, \\ B_0 y(t_0) + B_1 y(T) &= \eta. \end{aligned} \tag{2}$$

where $L(t) \in R^{m\times m}$. We assume that the BVP has a unique solution $y(t)$. The solution can be expressed in terms of the fundamental matrix $\Phi(t, t_0)$ as

$$y(t) = \Phi(t, t_0)Q^{-1}\eta$$

where $Q = B_0 + B_1\Phi(T, t_0)$ is assumed to be nonsingular. A perturbation $\delta\eta$ of the boundary conditions will cause a perturbation δy of the solution which is bounded by

$$\|\delta y(t)\| \le \|\Phi(t, t_0)Q^{-1}\|\|\delta\eta\|.$$

If we consider the following norms in $C([t_0, T])$

$$\|\delta y\|_\infty = \max_{t_0\le t\le T} \|\delta y(t)\|$$

and

$$\|\delta y\|_1 = \frac{1}{T - t_0}\int_{t_0}^{T} \|\delta y(t)\|dt$$

we obtain the two upper bounds

$$\|\delta y\|_\infty \le \kappa_c\|\delta\eta\|, \qquad \|\delta y\|_1 = \gamma_c\|\delta\eta\|$$

where, if $\phi(t) = \|\Phi(t, t_0)Q^{-1}\|$,

$$\kappa_c = \max_{t_0\le t\le T} \phi(t), \qquad \gamma_c = \frac{1}{T - t_0}\int_{t_0}^{T} \phi(t)dt.$$

These two parameters have been used to classify the conditioning of the continuous problem [2]. There are three possibilities. If both κ_c and γ_c have moderate sizes, then the continuous problem is well conditioned. If γ_c is of moderate size but $\kappa_c \gg \gamma_c$, then the problem is defined to be stiff; this means that the perturbation δy is large in subintervals which are small with respect to $T - t_0$.

The ratio $\sigma = \kappa_c/\gamma_c$ measures the stiffness of the problem. Finally, if both κ_c and γ_c are large then the problem is ill conditioned.

3. Conditioning of the discrete problem

To define the conditioning parameters for the discrete problem, we fix an initial mesh $\pi : t_0 < t_1 < \cdots < t_N$, with $h_i = t_i - t_{i-1}$, $i = 1, \ldots, N$, ($h = \max_i h_i$) on which the problem is to be approximated and we denote the vector of the numerical approximations by $\mathbf{y} = (y_0^T, y_1^T, \cdots, y_N^T)^T$, a block vector of size Nm. We use as the numerical method one of the symmetric differences schemes described in [3], even if the following considerations may apply to many other numerical methods. Then $\mathbf{y}$ satisfies the following discrete problem:

$$M\mathbf{y} = \mathbf{e}_1 \otimes \eta$$

where $\mathbf{e}_1 = (1\ 0\ \cdots\ 0)^T$ is of size $N+1$. The first row of the matrix M describes the boundary conditions while the others rows depend on the numerical method used.

Following [4], we define the block elements of the matrix $G = M^{-1}$ as G_{ij}, $i, j = 0, \ldots, N$, and the matrix Ω with elements $\Omega_{ij} = \|G_{ij}\|$ of size $N+1$. Then a perturbation $\delta\eta$ of the boundary condition produces a perturbation $\delta\mathbf{y}$ in the solution bounded by

$$\|\delta\mathbf{y}\| \leq \Omega_{*0}\|\delta\eta\|$$

where $\|\delta\mathbf{y}\| = (\|y_0\|, \|y_1\|, \cdots, \|y_N\|)^T$ and Ω_{*0} is the first column of the matrix Ω. We can therefore use the following parameters

$$\kappa_d(\pi) = \max_i \Omega_{i0}, \quad \text{and} \quad \gamma_d(\pi) = \frac{1}{T - t_0} \sum_{i=1}^{N} h_i \max(\Omega_{(i-1)0}, \Omega_{i0})$$

to define a bound for the error computed using the following two norms

$$e_\infty(\pi) = \|\delta\mathbf{y}\|_\infty \leq \kappa_d\|\delta\eta\|$$

and

$$e_1(\pi) = \frac{1}{T - t_0} \sum_{i=1}^{n} h_i \max(\|\delta y_{i-1}\|, \|\delta y_i\|) \leq \gamma_d\|\delta\eta\|.$$

Now $k_d(\pi)$ and $\gamma_d(\pi)$ correspond to k_c and γ_c for the discrete problem which can be classified similarly. However, unlike the k_c and γ_c, they depend on the mesh π and do not necessarily share the same order of magnitude.

This suggests the following definition:

Definition 1. A mesh π of size N is said *optimal WRC (with respect to conditioning)* if $k_c \simeq k_d(\pi)$ and $\gamma_c \simeq \gamma_d(\pi)$.

Our problem is then to find an optimal WRC mesh. In a following section we shall refine the obtained mesh by also considering the minimization of the global error. In this case, we shall obtain a mesh *optimal WRCE* (with respect both the conditioning and the error).

4. Error estimation

The strategy to approximate the global error and the local truncation error is similar to the one described in [3] for initial value problems.

Let

$$F_p(\mathbf{y}) = 0$$

be the discrete problem associated to the following general non linear BVP

$$\begin{aligned} y' &= f(t, y) \\ g(y(t_0), y(T)) &= \eta \end{aligned}$$

and let $\hat{\mathbf{y}} = (y(t_0)^T, y(t_1)^T, \ldots, y(t_N)^T)^T$ be the continuous solution evaluated on the mesh. We have that

$$F_p(\hat{\mathbf{y}}) = \tau_p, \qquad \tau_p = (0, \tau_{p,1}, \ldots, \tau_{p,N})^T, \qquad \tau_{p,i} = O(h_i^{p+1}).$$

The vector τ of the local truncation errors has 0 as first entry because the first row of the discrete nonlinear problem concerns the boundary conditions, which for simplicity are supposed to be exact. If we denote by $\hat{M}_p$ the Jacobian matrix of F_p computed in $\hat{\mathbf{y}}$, we have that

$$F_p(\hat{\mathbf{y}}) - F_p(\mathbf{y}) = \tau_p \quad \text{and} \quad \hat{\mathbf{y}} - \mathbf{y} \approx \hat{M}_p^{-1}\tau_p. \tag{3}$$

Now, let us consider a method of order $q > p$; the exact solution satisfies

$$F_q(\hat{\mathbf{y}}) = \tau_q$$

and

$$F_q(\mathbf{y}) \approx F_q(\hat{\mathbf{y}} - \hat{M}_p^{-1}\tau_p) \approx F_q(\hat{\mathbf{y}}) - \hat{M}_q\hat{M}_p^{-1}\tau_p. \tag{4}$$

In [3], Theorem 10.7.1 shows that

$$\hat{M}_q\hat{M}_p^{-1}\tau_p = \tau_p + O(h^{(p+2)}). \tag{5}$$

By inserting the relation (5) into (4) we obtain that $-F_q(\mathbf{y})$ is an approximation of τ_p of order $p+2$ and we derive the following relation for the absolute global error

$$\hat{\mathbf{y}} - \mathbf{y} \approx -M_p^{-1}F_q(\mathbf{y}) = \mathbf{e}$$

where now M_p is the Jacobian matrix evaluated at $\mathbf{y}$, the numerical solution.

We accept the computed solution if the following criterion

$$\max_{0\le i\le N}\left(\max_{1\le j\le m}\frac{|\mathbf{e}_{ij}|}{\max(Abstol_j, Reltol|\mathbf{y}_{ij}|)}\right) = \max_{0\le i\le N}\zeta_i \le 1 \tag{6}$$

is satisfied; *Reltol* and $Abstol_j$, $1 \le j \le m$ denote input tolerances, and ζ_i is the normalized relative error.

5. Mesh selection

As already said, our principal aim is to find an *optimal WRCE* mesh. Among the two quantities, $\gamma_d(\pi)$ is more suitable to be used in a step size variation strategy since it is defined by an integral (and then its behavior is smoother than that of $\kappa_d(\pi)$), and in fact $\gamma_d(\pi)$ is used in [4] in order to find the *optimal WRC* mesh. We have modified and improved this strategy by also using the information derived from the approximation of the global error defined in the previous section, in order to minimize the error in the numerical approximation with a given number of steps.

We will use the following result (see [4]):

Theorem 2. The first block column of G is an approximation of $\Phi(t, t_0)Q^{-1}$ of order p, if p is the order of the method used.

Therefore we can consider

$$\gamma_d(\pi) = \frac{1}{T-t_0}\sum_{i=1}^{N} h_i \max(\Omega_{i-1,0}, \Omega_{i0}) =$$

$$\frac{1}{T-t_0}\sum_{i=1}^{N} h_i \max(\phi(t_{i-1}), \phi(t_i)) + O(h^p);$$

this means that $\gamma_d(\pi)$ is related to the quadrature formula used to approximate γ_c. In order to construct an *optimal WRC* mesh we choose the mesh that minimizes the error between the quadrature formula and γ_c, that is

$$E_\gamma = \left| \frac{1}{T-t_0}\sum_{i=1}^{N} h_i \max(\phi(t_{i-1}), \phi(t_i)) - \frac{1}{T-t_0}\int_{t_0}^{T} \phi(t)dt \right|.$$

The error can be bounded by

$$E_\gamma \leq \frac{1}{T-t_0}\sum_{i=1}^{N} h_i(h_i|\phi'(\xi_i)| + O(h_i^2)),$$

where

$$\xi_i = \begin{cases} t_i, & \text{if } \phi(t_i) > \phi(t_{i-1}), \\ t_{i-1}, & \text{otherwise.} \end{cases}$$

Since we only know the discrete approximation of the function $\phi(t)$, we may approximate the error using the following bound

$$E_\gamma \leq \frac{1}{T-t_0}\sum_{i=1}^{N} h_i(|\Omega_{i0} - \Omega_{i-1,0}| + O(h_i^2))$$

and therefore, to minimize E_γ, we solve the minimax problem

$$\min \max_i h_i|\Omega_{i0} - \Omega_{i-1,0}|$$

defining the normalized monitor function

$$\psi_\gamma(t) = \alpha|\Omega_{i0} - \Omega_{i-1,0}|, \quad t \in (t_{i-1}, t_i)$$

where $\alpha = 1/\psi_\gamma^{max} = \max_i |\Omega_{i0} - \Omega_{i-1,0}|$. This problem can be solved using the usual technique of equidistribution [1].

Starting from a mesh π_0, after the equidistribution we obtain a new mesh π_1 with the same size. The equidistribution process could be repeated on π_1 obtaining a new mesh π_2, usually during this process we obtain $\gamma_d(\pi_0) \leq \gamma_d(\pi_1) \leq \gamma_d(\pi_2)$ because smaller stepsizes are used where the entries of ϕ_γ are larger, this

means that the value of E_γ decreases and also the value of γ_d decreases, since the quadrature formula approximates the integral from above.

In [4] the technique used to find the optimal mesh was to continue the equidistribution process until the value of γ_d decreases. This strategy is not efficient if we start the process with a coarse initial mesh, i.e. when the size of the mesh is not sufficient to provide a reasonable approximation of γ_d. So we decide to add and/or remove points if the new mesh π_{i+1} does not change enough with respect to the mesh π_i after the equidistribution process. The empirical technique for adding or removing points is based on the following quantities associated with the monitor function ψ_γ ([1] pag. 370):

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

and

$$r_2 = \sum_{i=1,\ldots,N}(\psi_\gamma(t_i)h_i)/N,$$

where h_i refers to the equidistributed mesh. If the mesh is well distributed then the ratio $r_1/r_2 \approx 1$ and usually the mesh is halved. Following the strategy used in [9], we decide to add two additional mesh points when $\psi_\gamma(t_i)h_i$ is greater than $\max(0.65r_1, r_2)$; we remove points, replacing two consecutive mesh intervals by one, when $\psi_\gamma(t_i)h_i$ is less than $10^{-3}r_2$. The action of removing points is more conservative since we need to not destroy the information already acquired concerning κ_d and γ_d.

Finally, we require the mesh to be locally quasi-uniform. This property is important since we deal with difference schemes and it is known that the stability properties of the associated difference equations are studied for constant stepsizes [3]. A smooth stepsize variation is then necessary to preserve the stability properties of the method. Locally quasi-uniformity is achieved by fixing two quantities r_π and c_π and by imposing the conditions

$$\begin{aligned} 1/r_\pi \leq h_i/h_{i-1} \leq r_\pi, \\ 1/r_\pi \leq h_{i+1}/h_i \leq r_\pi. \end{aligned} \tag{7}$$

If these restrictions do not hold for a certain h_i then we add a new mesh point, halving the step; moreover we impose the condition that the stepsize must remain constant c_π times. The sizes of r_π and c_π are chosen empirically.

The overall process allows us to obtain a discrete problem, whose conditioning parameters are almost insensitive to the addition of new points. This means that such parameters are good approximation of the continuous ones and an *optimal WRC* mesh has been found. Moreover on this mesh the discrete approximation is already a good approximation of the solution. This means that the approximation of the global error $\mathbf{e}$ is reliable and we can use it in the step size variation strategy.

Since we want to find the mesh that minimizes the error, we define the following monitor function

$$\psi_\zeta(t) = \max(|\zeta_i|, |\zeta_{i-1}|)^{1/p}, \quad t \in (t_{i-1}, t_i)$$

and the final monitor function is defined as a linear combination of ψ_ζ and ψ_γ. Numerical experiments show that the following monitor function performs well:

$$\psi_{tot}(t) = \alpha_\gamma \psi_\gamma(t) + \psi_\zeta(t), \quad t \in (t_{i-1}, t_i)$$

where $\alpha_\gamma = 0.05 \max_i \phi_\zeta(t_i)$. The empirical technique for adding and removing points and for achieving locally quasi-uniformity is the same that was used before for ψ_γ, but is now associated to ψ_ζ, using

$$r_1 = \max_{i=1,N} (\psi_\zeta(t_i) h_i) \qquad \text{and} \qquad r_2 = \sum_{i=1,N} (\psi_\zeta(t_i) h_i)/N.$$

A possible strategy to reduce the overall computational cost is to combine a lower order method with a higher order one. The lower order method is used to compute the mesh with the same conditioning parameters of the continuous problem, then we switch to the higher order method to reach the desired tolerance. The decision to change order or to introduce the error in the equidistribution strategy is taken by considering not only whether the mesh is *optimal WRC* but also if the problem is stiff. If the problem is not stiff or moderately stiff, no matter if it is well conditioned or ill conditioned, we use the higher order method.

The main computational cost in the mesh selection strategy is the computation of ψ_γ, which requires the solution of m linear systems, and this is expensive for problems of large size. Shortcuts in evaluating κ_d and γ_d could be defined in the case of large values of m, but we prefer at the present to skip this question.

6. Nonlinear problems

The mesh selection strategy described in the previous section is based on the conditioning parameters associated to a linear boundary value problem. To extend this strategy to nonlinear problems we apply the Newton scheme to the original continuous problem [10]. In this way we have to solve a sequence of linear problems, and for each problem we calculate the conditioning parameters and we define the monitor function for the mesh selection strategy. This is particularly useful for stiff problems, and the numerical experiments show the efficiency of this technique.

This strategy, called quasi-linearization [7], is usually not considered in codes for the solution of BVPs, the most used technique being the solution of the nonlinear discrete problems by means of a modified or a damped Newton method. However, if the problem is very stiff, and not enough information to fix a correct initial mesh is available, the convergence of the damped Newton scheme is not assured, so one is obliged to halve the mesh and try again.

We note that if the grid is not changed during the iterative procedure the two techniques generate the same sequence of discrete problems, but if the mesh is changed the behavior is completely different.

7. Numerical experiments

The MATLAB code TOM, that has been used for the numerical experiments, solves nonlinear BVPs by using as the discrete method the Top Order Methods [3] that are symmetric schemes described by the following formulas:

$$\sum_{i=0}^{\nu-1} \alpha_i(\mathbf{y}_{n+i} - \mathbf{y}_{n+k-i}) = h_n \sum_{i=0}^{\nu-1} \beta_i(\mathbf{f}_{n+i} + \mathbf{f}_{n+k-i}),$$

where $p = 2k$, k odd, $\nu = (k+1)/2$.

The code uses the order 6 method together with the method of order 2 in the same class, which corresponds to the trapezoidal rule. To estimate the error we use methods in the ETR class [3]. For nonlinear problems a quasilinearization procedure is used, as described in [10]. The value of r_π in (7) is set to 4 for the order 2 method and to 3 for the order 6 method, the value of c_π is set to 5. A problem is considered stiff if the stiffness ratio is higher than 10^3.

The numerical experiments were run on a PC with an Pentium III 1.2GHz processor and 256MBytes of RAM. The version of MATLAB is 6.5, Release 13; they try to show the effectiveness of the mesh selection strategy presented in section 5 to solve singularly perturbed BVPs.

We have compared the mesh selection described in section 4, with a mesh selection based only on the approximation of the error. In this latter case we do not compute the conditioning parameter,we set the order to 6, and the monitor function is always ψ_ζ. All the other empirical parameters are the same. This mesh selection is called optimal with respect to the error (TOM(WRE)).

In order to have a comparison with standard solvers we execute the numerical tests also with the MATLAB code BVP4c. BVP4c is a finite difference code that implements the three-stage Lobatto IIIa formula. The mesh selection and the error control are based on the residual of the continuous solution provided by the collocation polynomial [9]. Is not possible to make a fair comparison because BVP4c controls the residual and not the error; we choose this code because it is the only code inserted in the MATLAB PSE and uses a standard mesh selection strategy.

For our numerical experiments we took the tolerances $Abstol_j$ and $Reltol$ equal to tol for all the components. We give the analytical Jacobians as optional input parameters. When we know the exact solution we calculate the maximum relative error in the numerical solution. If $\pi : t_i, 0 \le i \le N$ is the final mesh, then we compute

$$\max_{0\le i\le N} \left(\frac{|y(t_i) - y_\pi(t_i)|}{\max(1, |y(t_i)|)} \right),$$

where y_π represents the numerical solution and y is the true solution.

Problem 1

The first test problem is the linear singularly perturbed problem presented in [1]. It has been chosen because, for $0 < \epsilon \ll 1$, the solution has a rapid transition layer at $t = 0$:

$$\epsilon y'' + ty' = 0 - \epsilon\pi^2 \cos(\pi t) - \pi t \sin(\pi t), \qquad y(-1) = -2, y(1) = 0.$$

The exact solution is $\cos(\pi t) + \mathrm{erf}(t/\sqrt{2\epsilon})/\mathrm{erf}(1/\sqrt{2\epsilon})$.

It has been solved for different values of the parameter ϵ and $tol = 10^{-3}$ and in Figure 1 we report the solution obtained with the WRCE mesh selection for $\epsilon = 10^{-8}$. The solution is reached after 8 steps and the final mesh has 401 points. Really different is the behavior of the code when the conditioning parameter are

			TOM(WRE)			TOM(WRCE)			BVP4c		
ϵ	κ_d	γ_d	Time	Error	N+1	Time	Error	N+1	Time	Error	N+1
10^{-1}	1.8	1.4	0.06	6.1e-5	16	0.06	6.1e-5	16	0.05	2.8e-5	17
10^{-2}	4.5	1.5	0.16	6.2e-6	61	0.16	7.9e-7	61	0.16	9.3e-6	37
10^{-3}	13	1.5	0.33	4.0e-5	121	0.33	3.5e-7	111	0.28	1.5e-4	57
10^{-4}	40	1.6	1.10	1.9e-7	241	0.33	7.3e-6	101	1.05	5.8e-4	95
10^{-5}	1.3e+2	1.5	2.41	2.9e-5	441	0.66	8.3e-8	211	2.41	8.8e-5	233
10^{-6}	4.0e+2	1.6	6.48	1.5e-5	751	1.26	1.0e-6	261	6.59	5.2e-4	530
10^{-7}	1.3e+3	1.5	*	*	*	1.49	1.6e-7	291	*	*	*
10^{-8}	4.0e+3	1.5	*	*	*	2.09	5.7e-9	401	*	*	*
10^{-9}	1.3e+4	1.5	*	*	*	3.02	3.8e-6	371	*	*	*
10^{-10}	4.0e+4	1.5	*	*	*	1.93	9.0e-7	361	*	*	*
10^{-11}	1.3e+5	1.5	*	*	*	5.55	4.5e-10	761	*	*	*
10^{-12}	4.0e+5	1.5	*	*	*	7.42	4.3e-8	731	*	*	*

Table 1

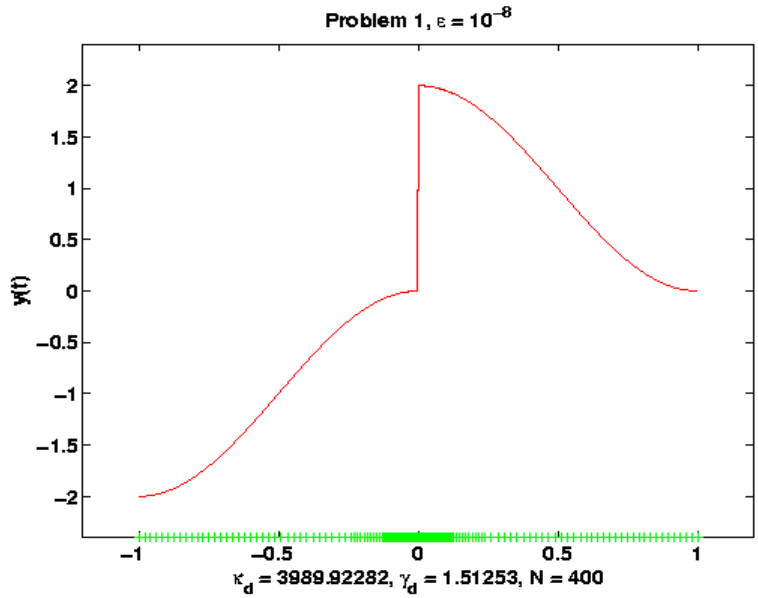
Conditioning parameters, execution time, error and final mesh for Problem 1

ϵ	Mesh sequence TOM(WRE)	Mesh sequence TOM(WRCE)
10^{-1}	16	16
10^{-2}	16,61	16,61
10^{-3}	16,61,121	16,61,111
10^{-4}	16,61,121,211,241	16,61,101
10^{-5}	16,61,121,211,351,441	16,61,111,211
10^{-6}	16,61,121,241,431,451,511,621, 731,751	16,61,111,111,191,261
10^{-7}	16,61,121,251,451,496,601,836, 1411,2161,2171,2201 *	16,61,111,211,211,291
10^{-8}	16,61,121,251,451,491,731,1211, 1441,2266 *	16,61,111,151,211,241,271,401
10^{-9}	16,61,121,251,451,491,781,1151, 1321,1356,1576,2121,2236,2486 *	16,61,101,131,201,241,331,371, 371,371
10^{-10}	16,61,121,251,451,491,771,1121, 1326,1606,1746,1836 *	16,61,101,141,211,251,361,361
10^{-11}	16,61,121,251,451,491,771,1121, 1326,1606,1746,1806 *	16,61,101,141,211,251,341,351, 381,761
10^{-12}	16,61,121,251,451,491,771,1121, 1326,1606,1746,1806 *	16,61,101,141,211,241,311,321, 381,381,571,731

Table 2

Mesh profile for Problem 1

Figure 1. Solution for Problem 1



not taken in consideration. In fact for $\epsilon = 10^{-8}$ it fails to give a solution using less than 2500 mesh points. When ϵ is small, TOM(WRE) requires a lot of points in order to have some information on the solution profile. This does not happen for TOM(WRCE): in this case the information about the behavior of the linear operator is given by the monitor function ϕ_γ.

To show the difference we report, in Table 1 the conditioning parameters, the behavior of the code TOM, using the two mesh selection strategies, and of BVP4c. In Table 2 we report the mesh profile for the code TOM using the two mesh selection strategies. The column headed "Time" in the table report the execution time, the column headed "N+1" report the number of mesh points in the final mesh. The asterisk signifies that more than the maximum number of mesh points (set to 2500) was required.

We see that the behaviors of TOM(WRE) and BVP4c are similar: both work very well when the values of ϵ are higher than 10^{-3}; for ϵ less than 10^{-6} both fail to give the solution without using more than 2500 mesh points. This is a common behavior of solvers for which the mesh selection is based only on the error or on the residual. Usually this problem is solved by using a continuation strategy. TOM(WRCE) is able to compute the solution for very small values of

the parameter ϵ, without using continuation. Moreover, the use of a continuation strategy further improves the performance of TOM(WRCE).

The conditioning parameters κ_d and γ_d reported in Table 1 show that the problem becomes stiff as ϵ decreases.

Problem 2

The second test problem has been chosen because it has a boundary layer of width $O(\epsilon)$ at $t = 0$ [6]:

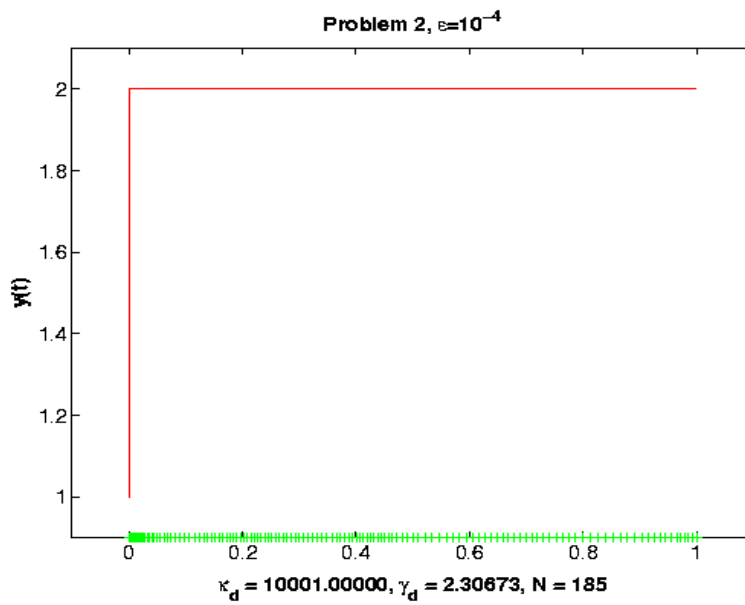
$$\epsilon y'' + y' = 0, \qquad y(0) = 1, y(1) = 2.$$

This problem has the exact solution

$$y(t) = \frac{2 - e^{-1/\epsilon} - e^{-t/\epsilon}}{1 - e^{-1/\epsilon}}.$$

The solution computed by TOM(WRCE) for $\epsilon = 10^{-4}$ is reported in Figure 2.

Figure 2. Solution for Problem 2



The comparison is reported in Tables 3 and 4: the execution times are similar for all the solvers when ϵ is larger than 10^{-3}; when the problem becomes more difficult TOM(WRCE) is able to compute the solution, whereas TOM(WRE)

			TOM(WRE)			TOM(WRCE)			BVP4c		
ϵ	κ_d	γ_d	Time	Error	N+1	Time	Error	N+1	Time	Error	N+1
10^{-1}	11	2.3	0.06	7.3e-5	16	0.06	7.3e-5	16	0.05	6.5e-5	16
10^{-2}	1e+2	2.4	0.27	2.1e-8	86	0.11	6.4e-5	46	0.11	3.0e-5	28
10^{-3}	1e+3	2	1.48	8.1e-5	181	0.50	7.4e-11	106	0.44	7.4e-4	74
10^{-4}	1e+4	2.3	4.45	3.3e-5	581	0.93	3.0e-5	186	1.64	1.7e-3	135
10^{-5}	1e+5	2	*			1.76	3.2e-10	271	6.49	0.0013	213
10^{-6}	1e+6	2	*			2.63	3.4e-11	256	S.J.		
10^{-7}	1e+7	2	*			5.00	2.2e-11	586	S.J.		
10^{-8}	1e+8	2	*			14.45	7.8e-12	1226	S.J.		

Table 3
Conditioning parameters, execution time, error and final mesh for Problem 2

ϵ	Mesh sequence TOM(WRE)	Mesh sequence TOM(WRCE)
10^{-1}	16	16
10^{-2}	16,61,86	16,46
10^{-3}	16,61,101,111,131,151,171,181	16,61,91,76,106
10^{-4}	16,61,121,206,311,446,461,561,581	16,61,116,166,186
10^{-5}	16,61,121,241,331,371,391,416,831, 871,911,1376,1426,1426 *	16,61,111,166,221,221,226,236,271
10^{-6}	16,61,121,221,261,511,826,836,846, 1196,1216,1226,2451,2471,2496 *	16,61,111,151,221,221,221,221,221, 231,236,246,256
10^{-7}	16,61,121,191,351,561,611,921,1191, 1221,1401,1426,1656,1656,1656 *	16,61,111,211,301,371,541,576,576, 586,586,586
10^{-8}	16,61,121,191,331,501,731,1066,1086, 1116,2231,2241,2311,2341 *	16,61,111,191,351,581,771,1001, 1111, 1181,1216,1216,1216,1226,1226,1226

Table 4
Mesh profile for Problem 2

and BVP4c fail. BVP4c fails because of a singular Jacobian (S.J. in the Table). The values of κ_d and γ_d show that the problem becomes very stiff as ϵ decreases.

Problem 3

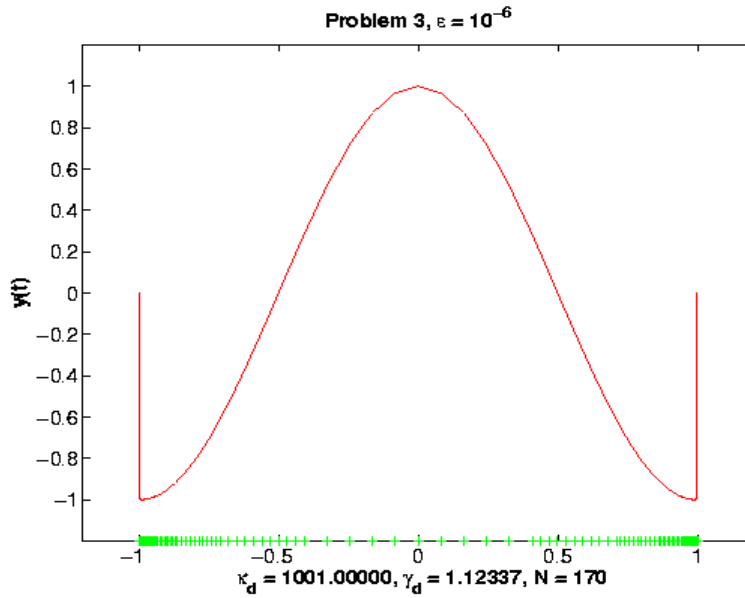
The third problem is a linear problem with two boundary layers [6]:

$$\epsilon y'' - y = -(\epsilon\pi^2 + 1)\cos(\pi t) \quad y(-1) = y(1) = 0.$$

The exact solution is

$$y(t) = \cos(\pi t) + \exp((x-1)/\sqrt{\epsilon} + \exp -(t+1)/\sqrt{\epsilon}.$$

Figure 3. Solution for Problem 3



			TOM(WRE)			TOM(WRCE)			BVP4c		
ϵ	κ_d	γ_d	Time	Error	N+1	Time	Error	N+1	Time	Error	N+1
10^{-1}	4.2	1.5	0.05	1.8e-3	16	0.05	1.8e-3	16	0.05	1.8e-3	16
10^{-2}	11	1.2	0.17	9.4e-7	51	0.16	2.1e-7	61	0.11	9.7e-5	24
10^{-3}	33	1.5	0.17	6.2e-4	61	0.17	1.7e-4	61	0.11	1.2e-4	38
10^{-4}	1.0e+2	1.1	0.55	4.5e-9	151	0.33	4.8e-7	111	0.27	1.5e-4	55
10^{-5}	3.2e+2	1.5	0.99	8.2e-9	231	0.33	3.6e-4	111	0.49	1.9e-4	90
10^{-6}	1.0e+3	1	3.07	7.1e-7	361	0.71	2.2e-7	171	0.88	1.3e-3	148
10^{-7}	3.2e+3	1	6.37	5.9e-5	1041	1.70	1.2e-9	291	3.46	1.6e-3	207
10^{-8}	1.0e+4	1	15.81	8.1e-4	1781	2.04	9.3e-9	351	16.59	3.0e-3	347
10^{-9}	3.2e+4	1	*	*	*	2.47	1.6e-7	341			
10^{-10}	1.0e+5	1	*	*	*	7.53	1.7e-8	921			

Table 5
Conditioning parameters, execution time, error and final mesh for Problem 3

It has been solved for different values of the parameter ϵ and $tol = 10^{-3}$ and in Figure 3 we report the solution computed by TOM(WRCE) for $\epsilon = 10^{-6}$; the solution is reached after 4 steps and the final mesh has 171 points.

In Table 5 we report the comparison. The conditioning parameters show that the problem becomes stiff for ϵ less than 10^{-6} and in fact TOM(WRCE)

ϵ	Mesh sequence TOM(WRE)	Mesh sequence TOM(WRCE)
10^{-1}	16	16
10^{-2}	16,51	16,61
10^{-3}	16,61	16,61
10^{-4}	16,61,101,151	16,61,111
10^{-5}	16,61,121,181,231	16,61,111
10^{-6}	16,61,121,201,221,261,301,321, 361	16,61,121,171
10^{-7}	16,61,121,231,431,451,491,981, 1041	16,61,131,201,196,241,291
10^{-8}	16,61,121,241,431,551,781,821, 861,1721,1761,1781	16,61,121,121,181,201,241,281, 351
10^{-9}	16,61,121,241,431,471,551,771, 811,871,1741,1741,1741,1741 *	16,61,121,121,211,261,271,281, 341,341
10^{-10}	16,61,121,241,431,821,901,981, 1961,2061,2201,2291 *	16,61,121,121,201,341,401,451, 471,491,601,631,821,921

Table 6
Mesh profile for Problem 3

becomes more efficient with respect to TOM(WRE) and BVP4c for values of ϵ less than 10^{-6}. In Table 6 we report the mesh sequence for TOM(WRE) and TOM(WRCE).

Problem 4

This problem describes the fluid injection through one side of a long vertical channel, considered in Example 1.4 of [1]: The differential equations

$$\begin{aligned} &f''' - R((f')^2 - ff''] + RA = 0, \\ &h'' + Rfh' + 1 = 0, \\ &\theta'' + Pf\theta' = 0, \end{aligned}$$

are to be solved subject to boundary conditions

$$\begin{aligned} &f(0) = f'(0) = 0, f(1) = 1, f'(1) = 0, \\ &h(0) = h(1) = 0, \\ &\theta(0) = 0, \theta(1) = 1. \end{aligned}$$

Here R and P are known constants, but A is determined by the boundary conditions.

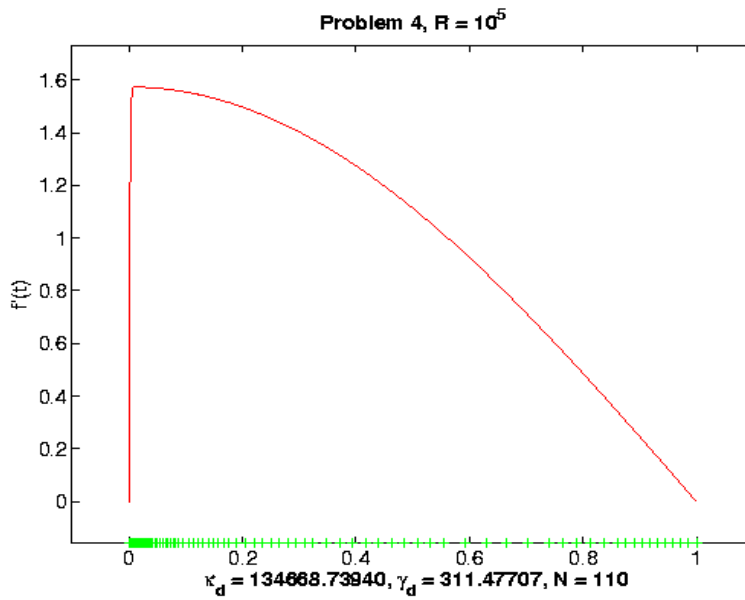
For a Reynolds number $R = 100$, this problem can be solved with crude guesses, but as R increases, it becomes much more difficult because of a boundary

			TOM(WRE)				TOM(WRCE)			
R	κ_d	γ_d	Time	N+1	h_{min}	h_{max}	Time	N+1	h_{min}	h_{max}
10^1	16	13	0.11	16	6.7e-2	6.7e-2	0.22	16	6.7e-2	6.7e-2
10^2	1.1e+2	17	0.39	61	1.3e-2	2.0e-2	0.33	46	6.4e-3	7.4e-2
10^3	1.3e+3	40	0.55	61	1.6e-2	1.8e-2	0.49	46	6.6e-3	7.3e-2
10^4	1.3e+4	1.0e+2	1.26	106	2.6e-3	2.2e-2	1.16	81	6.3e-4	4.9e-2
10^5	1.3e+5	3.1e+2	3.90	221	2.3e-4	1.5e-2	1.87	111	3.8e-4	3.7e-2
10^6	1.4e+6	1.0e+3	8.40	406	2.7e-4	5.2e-3	2.80	146	2.2e-4	2.0e-2
10^7	1.4e+7	3.1e+3	31.47	1336	7.9e-5	1.6e-3	4.72	211	4.6e-5	1.9e-2
10^8	1.4e+8	1.0e+4	42.08	1601	3.0e-5	2.1e-3	7.25	261	2.0e-5	7.8e-3
10^9	1.4e+9	3.5e+4	*	*	*	*	33.23	871	6.3e-6	2.5e-3

Table 7
Problem 4 - TOM

layer at t = 0. We consider A as unknown, adding the equation $A' = 0$. The code TOM in combination with the optimal WRCE mesh selection is able to compute the solution without continuation for very large values of the Reynolds number.

Figure 4. Solution for Problem 4



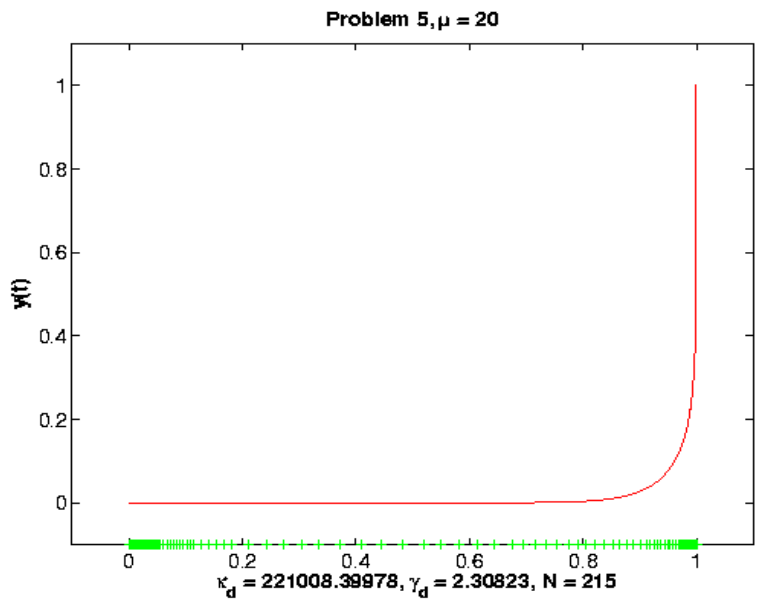
In Figure 4 we report the solution and the mesh profile computed by TOM(WRCE) for $R = 10^5$. The total number of Newton iterations, which corre-

	BVP4c			
R	Time	N+1	h_{min}	h_{max}
10^1	0.11	16	6.7e-2	6.7e-2
10^2	0.27	28	1.7e-2	5.0e-2
10^3	0.60	35	2.8e-3	7.5e-2
10^4	1.32	60	9.3e-4	2.5e-2
10^5 – 10^9	S.J.			

Table 8
Problem 4 - BVP4c

sponds to the number of linear BVPs that are solved, is 7, and only four different meshes with 16,46, 76 and 111 points are used. The first Newton iteration gives the approximated solution without changing the mesh, the second, the third and the forth iterations require two different meshes, and the other iterations converge using only one mesh. In Table 7 we report the results for the code TOM for different values of the Reynolds number, the initial guess is always zeros and $tol = 10^{-3}$. In Table 8 we report the results for BVP4c.

Figure 5. Solution for Problem 5



			TOM(WRE)				TOM(WRCE)			
μ	κ_d	γ_d	Time	N+1	h_{min}	h_{max}	Time	N+1	h_{min}	h_{max}
5	32	2.5	0.22	51	1.4e-2	4.0e-2	0.22	61	1.4e-02	2.0e-2
10	7.6e+02	2.7	1.21	196	3.6e-4	1.7e-2	0.54	71	9.1e-04	5.4e-2
15	1.4e+04	2.3	3.02	256	6.4e-5	1.9e-2	1.71	166	1.6e-05	4.1e-2
20	2.2e+05	2.2	23.29	1281	5.3e-6	3.9e-3	3.02	216	1.7e-06	3.7e-2
25	3.4e+06	2.2	27.90	1306	1.8e-7	7.9e-3	3.90	191	1.9e-08	5.0e-2
30	4.9e+07	2.2	*	*	*	*	5.11	301	2.0e-09	3.9e-2
35	7.0e+08	2.1	*	*	*	*	8.74	456	4.9e-11	5.8e-2
40	9.7e+09	2.3	*	*	*	*	12.96	411	2.7e-11	5.5e-2
45	1.3e+11	2.1	*	*	*	*	18.18	456	6.6e-13	4.7e-2
50	1.8e+12	2.2	*	*	*	*	26.04	551	8.7e-14	5.7e-2

Table 9
Problem 5 - TOM

	BVP4c			
μ	Time	N+1	h_{min}	h_{max}
5	0.22	21	8.3e-3	1.1e-1
10	1.43	53	6.2e-4	3.1e-2
15 – 50	S.J.			

Table 10
Problem 5 - BVP4c

Problem 5

The last nonlinear test, called Troesch's problem, is considered a difficult one [6]:

$$y'' = \mu \sinh(\mu y) \quad y(0) = 0, y(1) = 1.$$

the solution has a boundary layer near $x = 1$, see Figure 5. We use as initial guess 0.5 for y and zero for y' and $tol = 10^{-3}$. This problem is very difficult to solve without a good initial guess, especially for values of the parameter μ greater than 20. The mesh selection strategy based on the conditioning parameters, in combination with the quasilinearization strategy allows us to obtain the solution even with $\mu = 50$. The conditioning parameter reported in Table 9 show that the problem is very stiff. Also in this case the behaviors of TOM(WRE) and BVP4c are similar for $\mu = 5, 10$; for higher values of μ BVP4c finds a singular Jacobian (Table 10).

8. Conclusions

The mesh selection strategy based on the conditioning parameters, inserted in the code TOM, result to be much more efficient with respect to standard mesh selection strategies based on the error or on the residual, especially for singularly perturbed BVPs. Some preliminary results show that the conditioning parameters could be used also in combination with the residual, giving similar improvements. We also note that, for difficult problems, it is always possible to associate a continuation strategy, that could also take information from the conditioning parameters.

References

[1] U. Ascher, R. Mattheij, and R.D. Russell. *Numerical solution of boundary value problems for ODEs*. Prentice-Hall,Englewood CliffsNJ, 1988.

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

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

[4] Luigi Brugnano and Donato Trigiante. A new mesh selection strategy for ODEs. *Appl. Numer. Math.*, 24(1):1–21, 1997.

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

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

[7] E. J. Dean. An inexact Newton method for nonlinear two-point boundary value problems. *J. Optim. Theory Appl.*, 75(3):471–486, 1992.

[8] W. H. Enright and P. H. Muir. Runge-kutta software with defect control for boundary value odes. *SIAM J. Sci. Comput.*, 17:479–497, 1996.

[9] J. Kierzenka and L.F. Shampine. A BVP solver based on residual control and the MATLAB pse. *ACM Transaction of Mathematical Software*, 27(3):299–316, 2001.

[10] F. Mazzia and I. Sgura. Numerical approximation of nonlinear BVPs by means of BVMs. *Appl. Numer. Math.*, 42(1-3):337–352, 2002.

[11] R. Wright, J. Cash, and G. Moore. Mesh selection for stiff two-point boundary value problems. *Numerical Algorithms*, 7:205–224, 1994.