JOURNAL OF COMPUTATIONAL PHYSICS 122, 266–279 (1995)

An Automatic Continuation Strategy for the Solution of Singularly Perturbed Linear Two-Point Boundary Value Problems

J. R. CASH, G. MOORE, AND R. W. WRIGHT

Department of Mathematics, Imperial College, South Kensington, London SW7 2BZ, England

Received October 26, 1994; revised March 28, 1995

An automatic continuation algorithm for the solution of linear singular perturbation problems is developed and incorporated into two codes which implement global methods for solving two-point boundary value problems. Specifically, the algorithm will be based upon error estimates formulated in the collocation code COLSYS and the deferred correction code HAGRON. The benefits of using continuation are clearly demonstrated for both codes for a large class of problems.

1. INTRODUCTION

Singular perturbations of the form

$$\varepsilon \mathbf{u}' = \mathbf{f}(x, \mathbf{u}), \quad a \le x \le b, \quad \mathbf{g}(\mathbf{u}(a), \mathbf{u}(b)) = \mathbf{0} \tag{1}$$

arise in many practical applications. For example, in [3] applications are given from the fields of seismology, semiconductor theory, fluid dynamics, and the theory of shells of revolution, and many more examples occur in the literature. There is, therefore, a real need to be able to solve such problems efficiently, although they often present a formidable challenge to existing numerical methods. For example, the presence of the parameter ε in problems of the form (1) often causes the problem to become increasingly stiff or, in the nonlinear case, have a solution that requires a very good initial guess in order for the Newton iteration to converge. In such cases it is very difficult to approximate the solution of (1) directly. Problems of this nature usually require initial knowledge of the solution and mesh in order to solve them. Such knowledge can be acquired by solving a chain of problems in which the parameter varies and this process is called continuation (see, e.g., [3]).

Essentially, there are two types of continuation; the first of these is implemented when solving nonlinear problems which have multiple solutions for certain values of the continuation parameter. With such problems, the aim is to proceed carefully through the parameter range in order to identify all solution branches, turning points, and bifurcation points while minimising the number of Newton iterations by supplying a good initial guess to the solution at each step. This type of continuation is not considered here, as we shall only be concerned with linear problems. The second type of continuation is the so-called homotopy continuation in which the aim is to solve one very difficult problem by first solving a chain of easier problems. Here, the mesh and solution (if nonlinear) are passed on from problem to problem until the last (difficult) problem is reached with a suitable initial approximation to the solution and a suitable initial mesh. With this second type of continuation it is preferable to solve as few intermediate problems as possible, while minimising the number of Newton iterations and mesh adaptations at each step.

Since we shall be solving only linear problems, the prime consideration in the homotopic continuation process will be to ensure that for each problem a suitable initial mesh is provided so that little, or no, mesh adaptation has to be performed at each continuation step. The main aim is then to take as large a step in the continuation parameter as possible while ensuring that this consideration is satisfied. For example, consider the general linear singular perturbation problem,

$$\varepsilon u^{(m)} = \sum_{i=0}^{m-1} c_i(x) u^{(i)} + q(x), \quad a \le x \le b. \tag{2}$$

As ε becomes increasingly smaller for this problem, the solution generally becomes increasingly harder to approximate directly. Indeed, for very small values of the parameter ε, direct approximation of problem (2) may not be possible, hence the need for continuation. For the above problem (2) a schematic continuation algorithm might take the following form:

SCHEMATIC CONTINUATION ALGORITHM.

1. Select an initial problem ε_0, a desired final problem ε_f and an error tolerance *tol*.
2. Initially solve problem ε_0 on a mesh π_0. Set $j = 0$.
3. REPEAT
   - 3.1. Select ε_{j+1} such that we believe π_j to be a "good" initial mesh for this problem.
   - 3.2. Install π_j as initial mesh for problem ε_{j+1}.
   - 3.3. Ultimately solve problem ε_{j+1} on the final mesh π_{j+1}, where π_{j+1} is obtained from π_j by using the automatic mesh refinement algorithm of the underlying boundary value problem solver.

0021-9991/95 $12.00

3.4. $j = j + 1$
UNTIL $\{\varepsilon_j = \varepsilon_f\}$.

The above algorithm will, essentially, form the basis of the continuation strategy to be explained in this paper. Clearly, several points need to be expanded upon in order to have a clear understanding of how the algorithm will work. At this stage we develop our approach in a general way in order to elucidate the aim of our strategy, leaving the precise details for later sections.

The first point to note in the above algorithm is that the selection of ε_0 is not automatic; rather, it is supplied by the user. We note that it would be desirable to develop an efficient strategy in which the user need only supply the value of ε_f, although this has so far proved elusive. In practice ε_0 is usually chosen in the range [0.1, 1] in order to give an initial problem that is only mildly stiff. Having selected ε_0 and possibly supplied a uniform initial mesh, the boundary value problem code is then used to construct a solution for problem ε_0 which satisfies or approximately satisfies the tolerance. It is not considered vital that the solutions to the intermediate problems satisfy the tolerance exactly since, in the case of a homotopic strategy, it is only the solution to the final problem that is of interest.

It is important to clarify what is meant by specifying that π_j should be a "good" initial mesh for the problem ε_{j+1}. For the purpose of our algorithm a "good" initial mesh π_j shall be one that has the following two properties:

(i) at most one adaptation of the mesh π_j will be needed in forming a mesh π_{j+1} on which problem ε_{j+1} is appropriately resolved
(ii) the mesh formed from adapting π_j once shall be "close to optimal" in the sense that it shall not have many more mesh points than are actually required.

It will also be important to explain how we select ε_{j+1} such that we believe the mesh π_j will be a "good" initial mesh for this problem. In order to appreciate how parameter selection is performed in our strategy we will need to understand how the functions which control mesh selection in the two codes vary along the chain of selected problems. In particular, we will be aiming to develop a strategy which monitors and predicts the maximum values of these functions from problem to problem.

In the next section we explain the mesh selection or "monitor" functions used in both the collocation code COLSYS [1, 2] and the deferred correction code HAGRON [6–8]. In addition, we consider how these monitor functions are utilised in the mesh selection process. In Section 3 we discuss the choice of an appropriate formulation for the continuation parameter and give some notational details, while in Section 4 the full details of our continuation strategy are revealed. In Section 5 we list our test set and display the results for both codes with and without use of continuation. In Section 6 we draw conclusions and suggest further work of interest in this area. An important point to note about our strategy is that we are ultimately interested in continuation for nonlinear problems and we develop our approach with this in mind.

2. MESH SELECTION

In this section we explain the monitor functions used in the mesh selection algorithms of COLSYS and HAGRON. The behaviour of these functions from problem to problem will prove central to our continuation strategy. We do not provide the full details of how mesh selection is performed for both codes here; however, a full discussion of monitor functions and mesh selection can be found in [3].

2.1. *COLSYS Monitor Function*

COLSYS is a package of subroutines developed in 1980 by Ascher *et al.* [1] for solving mixed-order systems of ODEs subject to separated, multipoint boundary conditions. The method used to approximate the solution is collocation at Gaussian points.

It has been demonstrated [11] that the mesh selection algorithm of COLSYS exhibits certain "instabilities" when attempting to solve extremely stiff problems in a continuation framework. In a recent paper [11] the present authors developed certain modifications for eradicating these instabilities, a process which resulted in a modified version of the code, entitled COLMOD. Throughout this paper we shall use COLMOD instead of COLSYS whenever we are working in a continuation framework.

A useful feature of COLSYS is that it is possible to impose different error tolerances on different components of the collocation solution $\mathbf{v}(x)$. For example, let us suppose that (2) has been converted to first-order form and solved for a certain value of ε on a mesh π

$$\pi : a = x_1 < x_2 < \cdots < x_N < x_{N+1} = b,$$
$$h_i = x_{i+1} - x_i, \quad h = \max_{1 \le i \le N} h_i.$$

Given a set of *ntol* tolerances tol_j and pointers to which solution components c_j they apply, the code attempts to satisfy the following conditions

$$\|u_{c_j} - v_{c_j}\|_i \le tol_j(1 + \|v_{c_j}\|_i), \quad i = 1, ..., N; j = 1, ..., ntol, \quad (3)$$

where we define

$$\|u_m - v_m\|_i \equiv \max_{x_i \le x < x_{i+1}} |u_m(x) - v_m(x)|, \quad i = 1, ..., N. \quad (4)$$

The objective of the COLSYS mesh selection algorithm is to form a mesh π^* that yields a collocation solution which satisfies the above conditions (3). In order to show how this objective

is achieved we consider the first term of the global error of the collocation solution $\mathbf{v}(x)$. This can be shown to have the localised form

$$\mathbf{u}(x) - \mathbf{v}(x) \approx C\mathbf{u}^{(l+1)}(x)h_i^{l+1}, \quad x \in [x_i, x_{i+1}], i = 1, ..., N, \quad (5)$$

where C is a computable constant and l is the number of collocation points per subinterval. An $O(h)$ piecewise constant approximation $\mathbf{s}(x)$ to $\mathbf{u}^{(l+1)}(x)$ can be formed from the computed solution $\mathbf{v}(x)$, thus leaving us with an a posteriori error estimate

$$C\mathbf{s}(x)h_i^{l+1} \approx C\mathbf{u}^{(l+1)}(x)h_i^{l+1}, \quad x \in [x_i, x_{i+1}], i = 1, ..., N. \quad (6)$$

One the error estimate in (6) has been formed, the aim is then to establish whether the following conditions hold:

$$C|s_{c_j}(x_i)|h_i^{l+1} \leq tol_j(1 + \|v_{c_j}\|_i), \quad i = 1, ..., N, j = 1, ..., ntol. \quad (7)$$

In order to motivate the choice of COLSYS monitor function, let us suppose that on some subinterval h_I the above condition (7) does not hold for solution component v_{c_j}. In this situation extra mesh points would be added to subinterval h_I when forming a new mesh. The number of mesh points that would need to be added is determined by calculating the subinterval size h_I^* that would make (7) an equality, that is,

$$h_I^* = \left[\frac{tol_j(1 + \|v_{c_j}\|_I)}{C|s_{c_j}(x_I)|}\right]^{1/(l+1)}. \quad (8)$$

Therefore, the number of points that need to be added to the old subinterval h_I when forming a new mesh is calculated in the following manner

$$\frac{h_I}{h_I^*} = h_I\left[\frac{C|s_{c_j}(x_I)|}{tol_j(1 + \|v_{c_j}\|_I)}\right]^{1/(l+1)}. \quad (9)$$

It is this calculation that is essentially used when performing mesh selection in COLSYS. In general, however, the COLSYS monitor function is a piecewise constant function of the following form:

$$\phi(x) = \max_{1 \leq j \leq ntol}\left[\frac{C|s_{c_j}(x_i)|}{tol_j(1 + \|v_{c_j}\|_i)}\right]^{1/(l+1)}, \quad x \in [x_i, x_{i+1}], i = 1, ..., N. \quad (10)$$

It is the behaviour of the function $\phi(x)$ from problem to problem that will prove to be central to our continuation strategy.

We finish our discussion of the COLSYS monitor function by noting that a new mesh π^* is formed in COLSYS by considering the equidistribution of the function (10) over each of the subintervals of π^*, that is,

$$\int_{x_i^*}^{x_{i+1}^*} \phi(x)dx = \frac{1}{N^*}\int_a^b \phi(x)dx, \quad 1 \leq i \leq N^*. \quad (11)$$

The number of mesh points N^* in the new mesh π^* is determined by the calculation

$$N^* = \sum_{i=1}^{N} \phi(x_i)h_i. \quad (12)$$

2.2. *HAGRON Monitor Function*

HAGRON is a package of subroutines developed by Cash and Wright [6–8] for solving first-order systems of ODEs subject to separated, two-point boundary conditions. The method used to approximate the solution is based on symmetric Runge–Kutta formulae with deferred corrections.

In order to describe the monitor function used for mesh selection in HAGRON, we briefly describe how this code calculates its numerical solution. For a given problem, HAGRON computes a first solution η using a fourth-order Runge–Kutta formula ξ_4 so that

$$\xi_4(\eta) = 0, \quad (13)$$

where η is a fourth-order solution vector on the given mesh. The fourth-order solution is then substituted into a higher order formula to construct an appropriate deferred correction. A higher order solution $\overline{\eta}$ is then computed from

$$\xi_4(\overline{\eta}) = -\xi_6(\eta), \quad (14)$$

where ξ_6 is a sixth-order Runge–Kutta formula. Once again a higher order deferred correction can be formed and thus an eighth-order solution $\overline{\overline{\eta}}$ is computed from

$$\xi_4(\overline{\overline{\eta}}) = -\xi_6(\eta) - \xi_8(\overline{\eta}). \quad (15)$$

The final deferred correction $\xi_8(\overline{\eta})$ is the value that HAGRON uses in its monitor function for adapting the mesh. This sixth-order correction is a piecewise constant over each subinterval of the mesh and can be shown to be an $O(h)$ approximation to an expression of the form

$$\xi_{8,i}(\overline{\eta}) \approx \mathbf{f}(x)h_i^6, \quad x \in [x_i, x_{i+1}], i = 1, ..., N, \quad (16)$$

where $\mathbf{f}(x)$ is a continuous function appearing in the exact form of the error in the sixth-order solution. We take $\xi_{8,i}(\overline{\eta})$ to mean the value of the deferred correction on the ith mesh subinterval. Thus, a piecewise constant $O(h)$ approximation to $\mathbf{f}(x)$ is easily calculated

$$\mathbf{s}(x) = \frac{\xi_{8,i}(\bar{\eta})}{h_i^6} \approx \mathbf{f}(x), \quad x \in [x_i, x_{i+1}], i = 1, ..., N. \quad (17)$$

As with COLSYS we now have an a posteriori error estimate, and we aim to establish whether the conditions

$$|s_{c_j}(x_i)|h_i^6 \leq tol_j * \max(1, |\eta_i^{c_j}|), \quad i = 1, ..., N, j = 1, ..., ntol, \quad (18)$$

hold, where $\eta_i^{c_j}$ is the value of the c_jth component of the solution at the ith mesh point. The choice of a monitor function in HAGRON is therefore motivated in a fashion similar to that of COLSYS; in short, it is a piecewise constant function of the form

$$\phi(x) = \max_{1 \leq j \leq ntol} \left[\frac{|s_{c_j}(x_i)|}{tol_j * \max(1, |\eta_i^{c_j}|)} \right]^{1/6}, \quad x \in [x_i, x_{i+1}], \quad i = 1, ..., N. \quad (19)$$

As with COLSYS, it is the behaviour of this function along the chain of continuation problems that will be central to our continuation strategy.

The main difference between the COLSYS and HAGRON mesh selection strategies arises due to the manner in which their respective monitor functions are used to calculate a new mesh. Essentially, COLSYS adapts the mesh in a dynamic way and attempts to equidistribute its monitor function, as in (11). HAGRON, however, utilises its monitor function in a static way, in the sense that it tends to either remove a particular mesh point or to keep it in the mesh rather than to move it to a different location. The mesh selection algorithm in HAGRON adds extra mesh points to a particular subinterval whenever $\phi(x_i)h_i > 1$. It adds the new points at equal subintervals of the old subinterval, and the number of points to be added is determined by the value of $\phi(x_i)h_i$.

3. PARAMETER SELECTION AND NOTATION

In this section we give some notational details that will be used throughout this paper. Additionally, we discuss the method that we use for parameter selection in our continuation strategy. At this stage, we do not motivate the basis for our particular selection strategy, instead we delay a discussion of this matter until the next section.

3.1. *Notation*

In the Introduction we noted that our continuation strategy would involve the solution of a sequence of problems in which the continuation parameter gradually becomes smaller. That is, given an initial problem ε_0 and a final problem ε_f, we solve a sequence of problems in which the parameter varies as

$$\varepsilon_0 > \varepsilon_1 > \varepsilon_2 > \cdots > \varepsilon_j > \cdots > \varepsilon_f. \quad (20)$$

For each of these problems we denote $\pi_{j,k}$ as the kth mesh formed in solving problem ε_j, and on any mesh $\pi_{j,k}$ with N points we define

$$\Phi_{j,k} = \max_{i=1,N} \phi(x_i), \quad H_{j,k} = \{h_i : \phi(x_i) = \Phi_{j,k}\}, \quad P_{j,k} = \sum_{i=1}^{N} \phi(x_i)h_i. \quad (21)$$

$P_{j,k}$ may be considered as a prediction of the number of points that will be required in the $(k + 1)$th mesh for problem ε_j.

Finally, if we consider the exact counterparts of the monitor functions (10) and (19), then we may define $\Phi_{j,t}$ as the best available approximation to the exact monitor function for problem ε_j. By "exact counterpart" we mean the functions that result from replacing the approximation $\mathbf{s}(x)$ in (10) with $\mathbf{u}^{(l+1)}(x)$ and the approximation $\mathbf{s}(x)$ in (19) with $\mathbf{f}(x)$. We will expand upon how $\Phi_{j,t}$ is determined in Section 4.

3.2. *Parameter Selection by Extrapolation*

In the previous section we noted that our continuation strategy would be based upon the behaviour of the monitor functions (10) and (19) along the chain of continuation problems (20). To be more specific, the strategy relies upon monitoring and predicting the maximum value of these functions from problem to problem. At each continuation step, our aim will be to predict, by reference to information assembled from previous continuation steps, the value of the continuation parameter that we believe will correspond with the maximum value of the monitor function taking a desired value.

Let us suppose that we have solved problems ε_{j-1} and ε_j and that we now want to select the parameter ε_{j+1} such that $\Phi_{j+1,t} \approx \Phi_{j+1,d}$, where $\Phi_{j+1,d}$ is the desired value of $\Phi_{j+1,t}$ (we detail how $\Phi_{j+1,d}$ is chosen in Section 4). It is possible to linearly extrapolate to find the appropriate value of ε_{j+1} as

$$\varepsilon_{j+1} = \varepsilon_{j-1} + (\Phi_{j+1,d} - \Phi_{j-1,t}) \left[\frac{\varepsilon_j - \varepsilon_{j-1}}{\Phi_{j,t} - \Phi_{j-1,t}} \right]. \quad (22)$$

Alternatively, if problem ε_{j-2} has also been solved it would be possible to quadratically extrapolate. In general, quadratic extrapolation is the highest order extrapolation that we use in our strategy.

Naturally, the accuracy of both linear and quadratic extrapolation will depend greatly upon the smoothness properties of the functions that we are trying to extrapolate. It is with this in mind that we consider the following example.

EXAMPLE 1. Consider a singular perturbation problem with a boundary layer at the right end of the interval

$$\varepsilon u'' - u' = 0, \quad u(0) = 1, \quad u(1) = 0. \tag{23}$$

The exact solution for this problem is

$$u(x) = \frac{1 - e^{(x-1)/\varepsilon}}{1 - e^{-1/\varepsilon}}. \tag{24}$$

In order to demonstrate the potential hazards of the extrapolation strategy (22), let us consider the exact counterpart of the COLSYS monitor function (10) for this problem for the case $l = 4$, $ntol = 1$, and $c_1 = 1$. The exact version of the monitor function is

$$\phi(x) = \left[\frac{Ce^{(x-1)/\varepsilon}}{\varepsilon^5(1 - e^{-1/\varepsilon})tol_1(1 + |u(x)|)}\right]^{1/5}. \tag{25}$$

We are interested in the behaviour of the maximum value of this function as ε varies. For this problem the maximum value of (25) occurs at $x = 1$ for all values of ε. We denote this maximum value for each ε by $\Phi(\varepsilon)$, and after grouping constants, we find its value to be of the form

$$\Phi(\varepsilon) = B/\varepsilon + O(e^{-1/\varepsilon}), \tag{26}$$

where B is a constant. The derivatives with respect to ε of the function (26) are particularly large for small ε, which makes it unsuitable for linear or quadratic extrapolation. However, if we simply consider expressing (26) in terms of

$$E = 1/\varepsilon \tag{27}$$

then we form a function which is altogether more easy to deal with, namely

$$\Phi(E) = BE + O(e^{-E}) \tag{28}$$

and, clearly, since this is essentially a linear function, linear extrapolation should work very well here.

This example indicates what we have generally found to be true with singular perturbation problems, namely, that greater accuracy is achieved in our parameter selection strategy if we consider extrapolation of the function $\Phi(E)$ instead of $\Phi(\varepsilon)$, where $E = 1/\varepsilon$.

As a footnote to this section, we outline the method for parameter selection when only the first problem ε_0 has been solved. For this special case, we simply assume that $\Phi(E)$ is a linear function of E with gradient m. Thus we can predict m from

$$\Phi_{0,t} = mE_0 \tag{29}$$

and hence we select E_1 (and consequently ε_1) as

$$E_1 = \Phi_{1,d}/m, \tag{30}$$

where $\Phi_{1,d}$ is the desired value of $\Phi_{1,t}$.

4. OVERALL CONTINUATION STRATEGY

In this section we aim to draw together and expand upon the details of the first three sections in order to establish our overall continuation strategy.

4.1. *Derivation of Best Approximation* $\phi_{j,t}$

In the previous section we defined the variable $\Phi_{j,t}$ to be the best available approximation to the maximum value of the exact monitor function for problem ε_j. We now concern ourselves with how this variable is derived at each continuation step.

Let us suppose that we have solved problem ε_j on an initial mesh $\pi_{j,1}$ and established the variables defined in (21). The initial mesh $\pi_{j,1}$ at each continuation step does not normally yield a numerical solution that appropriately satisfies the error tolerances, therefore, a second mesh $\pi_{j,2}$ is usually formed on which a new numerical solution is calculated, along with the variables in (21). We note at this stage, that one of the aims of our continuation strategy is to select ε_j such that only one adaptation of the initial mesh $\pi_{j,1}$ will be needed in forming a mesh on which the problem ε_j is appropriately resolved. In general then, the mesh $\pi_{j,2}$ will be the final mesh required for problem ε_j.

In Section 2, we noted that the monitor functions (10) and (19) contain first-order approximations, denoted by $s(x)$, to certain functions that appear in the first term of the error for both methods. We now consider the concept of an exact monitor function, which we shall denote by $\phi_t(x)$, and which is formed by replacing the approximations $s(x)$ in (10) and (19) with their exact counterparts. Thus in general, for either of the monitor functions (10) and (19), we have

$$\phi(x) = \phi_t(x) + cg(x)h_i + \text{h.o.t.}, \quad x \in [x_i, x_{i+1}], \; i = 1, \ldots, N, \tag{31}$$

where c is a constant and $g(x)$ denotes a general function found in the first term of the error that results from approximating $\phi_t(x)$ by either (10) or (19). Now, if we denote by $\phi_{j,t}(x)$ the function $\phi_t(x)$ for problem ε_j and consider the variables (21) on the meshes $\pi_{j,1}$ and $\pi_{j,2}$, we have

$$\begin{aligned} \Phi_{j,1} &= \Phi_{j,t} + \bar{c}H_{j,1} \\ \Phi_{j,2} &= \Phi_{j,t} + \bar{c}H_{j,2}, \end{aligned} \tag{32}$$

where

$$\Phi_{j,t} \approx \max_{x \in [a,b]} \phi_{j,t}(x), \quad \bar{c} = cg_j(x_m), \quad x_m = \{x : \phi_{j,t}(x) = \Phi_{j,t}\}.$$

An important point to note about the equations (32) is that while the variables $\Phi_{j,2}$ and $H_{j,2}$ are calculated in the manner given in (21), the variables $\Phi_{j,1}$ and $H_{j,1}$ are calculated as

$$\Phi_{j,1} = \max_{x\in[X_1,X_2)} \phi_{j,1}(x), \quad H_{j,1} = \{h_i : \phi_{j,1}(x_i) = \Phi_{j,1}\}, \tag{33}$$

where

$$H_{j,2} = X_2 - X_1, \quad \phi_{j,2}(x) = \Phi_{j,2} \quad \forall x \in [X_1, X_2].$$

We note that, in most cases, the evaluation of $\Phi_{j,1}$ and $H_{j,1}$ by means of (33) will be equivalent to their evaluation by (21). Now, since the two equations (32) contain only two unknowns, it is possible to calculate $\Phi_{j,t}$ which we deem to be the most accurate available approximation to the maximum value of the exact monitor function for problem ε_j.

In general, the calculation of $\Phi_{j,t}$ relies upon forming the two equations (32). However, it occasionally arises that the initial mesh $\pi_{j,1}$ is apt for the appropriate solution of a given continuation problem ε_j, in which case, only the first of the two equations (32) is available. In this case, we simply choose $\Phi_{j,t} = \Phi_{j,1}$.

4.2. *Choice of Desired Maximum Monitor Function Value* $\Phi_{j+1,d}$

In Section 1 we discussed the properties that constituted a "good" initial mesh, we now formalise these properties in order to gain an understanding of how $\Phi_{j+1,d}$ is selected at each continuation step. Let us suppose that we have solved problem ε_j on initial mesh $\pi_{j,1}$ and final mesh $\pi_{j,2}$, we desire that the following two properties hold:

$$\begin{aligned} &\text{(i) } \Phi_{j,t} H_{j,2} \le 2, \\ &\text{(ii) } P_{j,1} \le 1.5 * P_{j,2}. \end{aligned} \tag{34}$$

It is worth noting that these properties are not only required for the intermediate continuation problems, but also for the final problem ε_f. However, for the final problem, it is usually the case that more than two meshes are formed in order to satisfy the tolerances exactly. In particular, if the mesh $\pi_{f,2}$ does not yield a solution which satisfies the tolerances we mesh select once more to form a third mesh $\pi_{f,3}$ and, if this mesh does not yield an appropriate solution then the mesh is repeatedly doubled until either the tolerances are satisfied or there is no more storage space.

If we consider that, for either of the codes under consideration here, the calculation $\phi(x_i)h_i$ gives the number of mesh points that need to be added to the ith subinterval in order for the tolerance to be satisfied there, then condition (i) tells us that, on our final mesh $\pi_{j,2}$, one mesh point at the most will need to be added to subinterval $H_{j,2}$ in order to resolve the problem to the requested tolerance. The choice of one here is somewhat heuristic and it acts as a relaxation of the tolerance constraints for each intermediate problem in the chain of continuation problems. The reason for relaxing the tolerance constraints is to allow the selection of larger continuation step lengths.

Condition (ii) is concerned with ensuring that we do not form meshes that are "far from optimal." Let us consider the situation where a mesh $\pi_{j,1}$ is particularly unsuitable for the solution of problem ε_j (for example, a boundary layer may not be resolved). In this case the effect of the poor resolution of the boundary layer would be to contaminate the error estimates not only in the boundary layer but also in the smooth part of the solution; the overall effect would be that a large mesh $\pi_{j,2}$ would be predicted. However, on the mesh $\pi_{j,2}$, the boundary layer would be more appropriately resolved and thus the error estimates in the smooth part of the solution less contaminated; hence we would expect the prediction of a smaller mesh $\pi_{j,3}$. This is a situation we wish to avoid, since numerical solution on the large mesh $\pi_{j,2}$ is computationally expensive and theoretically unnecessary.

So how do we select $\Phi_{j+1,d}$ in order to satisfy the above conditions? Let us consider condition (i) first. For a given continuation problem, ε_j, let us suppose that the monitor function $\phi_{j,1}(x)$ has been formed on a mesh $\pi_{j,1}$. Given the function $\phi_{j,1}(x)$, both COLMOD and HAGRON select a new mesh, $\pi_{j,2}$, such that

$$\phi_{j,1}(x_i)h_i \le 1, \quad 1 \le i \le N_{j,2}, \tag{35}$$

where $N_{j,2}$ is the number of mesh points in $\pi_{j,2}$. In particular, we have

$$\Phi_{j,1} H_{j,2} \le 1. \tag{36}$$

Now, if

$$\Phi_{j,1} = \Phi_{j,t}/2 \tag{37}$$

then we have condition (i). Thus, we aim to ensure that (37) is satisfied. If we refer back to (32), we can see

$$\Phi_{j,1} = \Phi_{j,t} + \tilde{c}H_{j,1}; \tag{38}$$

thus, in order to satisfy (37) we require

$$\tilde{c}H_{j,1} = -\Phi_{j,t}/2. \tag{39}$$

However, if (39) is not satisfied for a given problem ε_j, then we can calculate $\hat{H}_{j,1}$, which is a prediction of the value that $H_{j,1}$ should have taken in order for (39) to be satisfied,

$$\hat{H}_{j,1} = -\Phi_{j,t}/(2\tilde{c}). \tag{40}$$

Thus, when we consider the selection of the new parameter value ε_{j+1}, we will do so with the aim that

$$\Phi_{j+1,l}H_{j+1,1} = \Phi_{j,l}\hat{H}_{j,1}. \tag{41}$$

We already know the initial mesh for ε_{j+1} and if we assume that the maximum value of the monitor function remains on the same subinterval as for problem ε_j; that is, if we assume

$$H_{j+1,1} = H_{j,2}, \tag{42}$$

then we can calculate the choice of desired maximum monitor function value as

$$\Phi_{j+1,d} = \Phi_{j,l}\hat{H}_{j,1}/H_{j+1,1}. \tag{43}$$

The assumption that the maximum value of the monitor function will remain in the same location from one continuation problem to the next rarely poses a practical restriction for linear problems. However, if the situation arises that

$$H_{j+1,1} \neq H_{j,2}, \tag{44}$$

then when selecting the value of $\Phi_{j+2,d}$ we would do so in the following manner:

$$\Phi_{j+2,d} = \left(\frac{H_{j,2}}{H_{j+1,1}}\right) * \Phi_{j+1,l}\hat{H}_{j+1,1}/H_{j+2,1}. \tag{45}$$

Before discussing the importance of the second condition in (34), we note that for each continuation problem ε_j we expect the a posteriori error estimate on $\pi_{j,2}$ to be more accurate than that on $\pi_{j,1}$. Now, since mesh selection is computationally inexpensive and since the error estimate on $\pi_{j,2}$ is freely available, we generally form the initial mesh $\pi_{j+1,1}$ for problem ε_{j+1} by adapting the mesh $\pi_{j,2}$ in the usual manner.

With regard to the second of the conditions outlined in (34), our choice of $\Phi_{j+1,d}$ is only influenced when this condition fails. Indeed, our strategy for selecting $\phi_{j+1,d}$ at each step is rather less straightforward than simply selecting $\Phi_{j+1,d}$ as in (43), and involves various safeguards in order to ensure that our parameter selection strategy works smoothly. Perhaps, the most important of these safeguards is implemented when condition (ii) fails for some problem ε_p; in this case we insist that for all future problems

$$\Phi_{p+j,d}H_{p+j,1} < \Phi_{p,l}H_{p,1}, \quad j = 1,2,\ldots. \tag{46}$$

This restriction is imposed on the basis that the degree of (lack of) resolution of the transition layer for problem ε_p leads to the construction of a poor error estimate, and hence the failure of condition (ii). In general, in our algorithm we select $\phi_{j+1,d}$ as in (43) and then impose the following safeguards

$$\begin{aligned}
&\text{if } \Phi_{j+1,d}H_{j+1,1} \geq N_j,\\
&\quad \phi_{j+1,d} = \min\left(\Phi_{j+1,d}, \max\left(\tfrac{3}{2} * N_j, (N_j + 3)\right)/H_{j+1,1}\right),\\
&\text{if } \Phi_{j+1,d}H_{j+1,1} < N_j,\\
&\quad \Phi_{j+1,d} = \max\left(\Phi_{j+1,d}, \max\left(\tfrac{2}{3} * N_j, (N_{j-3})\right)/H_{j+1,1}\right)
\end{aligned} \tag{47}$$

where $N_j = \Phi_{j,l}H_{j,1}$. These safeguards are experimentally, rather than theoretically, based and, while it is probable that the factors $\frac{3}{2}$, $\frac{2}{3}$, and 3 could be selected differently without degrading the performance of our algorithm, we have found them to be reliable for a large class of problems.

5. TEST PROBLEMS AND NUMERICAL RESULTS

In this section we test our continuation strategy on four linear singular perturbation problems. Results for solution with and without use of continuation are displayed, and these results not only reveal the benefits of continuation but also serve as a useful guide to the relative merits of COLSYS and HAGRON. For the COLSYS non-continuation results we use the most recent version of COLSYS, called COLNEW [4], while for the continuation results a modified version of COLNEW, called COLMOD [11], is utilised. The codes COLMOD and HAGRON, which is a modification of TWPBVP.f appearing in the ode directory of NETLIB, are available from the third author on request.

5.1. *Test Problems*

Our test set consists of four linear second-order two-point boundary value problems, and are drawn from the challenging test set in [9]. The solutions to these problems variously exhibit boundary layer, interior layer, corner layer, and smooth types of behaviour. In our numerical tests the problems are converted to first-order form.

Problem 1. $\varepsilon y'' - y = 0$, $y(0) = 1$, $y(1) = 0$. Exact solution is

$$y(x) = (\exp(-x/\sqrt{\varepsilon}) - \exp((x-2)/\sqrt{\varepsilon}))/(1 - \exp(-2/\sqrt{\varepsilon})).$$

The solution has a boundary layer of width $O(\sqrt{\varepsilon})$ at $x = 0$.

Problem 2. $\varepsilon y'' + xy' = -\varepsilon\pi^2 \cos(\pi x) - \pi x \sin(\pi x)$, $y(-1) = -2$, $y(1) = 0$. Exact solution is $y(x) = \cos(\pi x) + \operatorname{erf}(x/\sqrt{2\varepsilon})/\operatorname{erf}(1/\sqrt{2\varepsilon})$. The solution has a shock layer in the turning point region near $x = 0$.

Problem 3. $\varepsilon y'' + xy' = 0$, $y(-1) = 0$, $y(1) = 2$. Exact solution is $y(x) = 1 + \operatorname{erf}(x/\sqrt{2\varepsilon})/(1/\sqrt{2\varepsilon})$. The solution has a turning point of width $O(\sqrt{\varepsilon})$ at $x = 0$.

Problem 4. $\varepsilon y'' - y = -(\varepsilon\pi^2 + 1)\cos(\pi x)$. The boundary conditions are $y(-1) = y(1) = 0$. Exact solution is $y(x) = \cos(\pi x) + \exp((x-1)/\sqrt{\varepsilon}) + \exp(-(x+1)/\sqrt{\varepsilon})$. The solution has boundary layers near $x = -1$ and $x = 1$.

TABLE I

Problem 1, COLSYS Results

	COLNEW, no continuation			COLMOD, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	80	0.50d-09	0.07	34	0.73d-09	0.09	3
0.10d-03	40	0.77d-08	0.05	48	0.82d-09	0.14	4
0.10d-05	42	0.62d-07	0.07	52	0.81d-09	0.22	6
0.10d-07	140	0.37d-09	0.27	52	0.85d-09	0.23	6
0.10d-09	320	0.61d-11	0.82	56	0.84d-09	0.26	7
0.10d-11	1280	0.16d-08	2.47	56	0.81d-09	0.31	8
0.10d-13	3288	0.13d-12	8.61	56	0.82d-09	0.35	9
0.10d-15				55	0.92d-09	0.35	9

TABLE II

Problem 1, HAGRON Results

	HAGRON, no continuation			HAGRON, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	24	0.21d-09	0.01	17	0.68d-08	0.02	3
0.10d-03	59	0.16d-09	0.02	32	0.21d-08	0.03	4
0.10d-05	64	0.18d-10	0.03	33	0.46d-08	0.04	5
0.10d-07	140	0.27d-08	0.06	37	0.24d-08	0.06	6
0.10d-09	167	0.18d-10	0.08	52	0.92d-09	0.08	7
0.10d-11	227	0.21d-10	0.10	52	0.96d-09	0.10	8
0.10d-13	814	0.59d-12	0.46	52	0.14d-08	0.11	9
0.10d-15	10387	0.29d-12	5.08	52	0.20d-08	0.12	10

TABLE III

Problem 2, COLSYS Results

	COLNEW, no continuation			COLMOD, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	80	0.29d-08	0.08	80	0.17d-08	0.21	3
0.10d-03	160	0.12d-08	0.16	110	0.16d-08	0.34	4
0.10d-05	92	0.18d-06	0.18	124	0.16d-08	0.45	5
0.10d-07	160	0.19d-08	0.33	126	0.17d-08	0.55	6
0.10d-09	160	0.15d-07	0.43	139	0.16d-08	0.66	7
0.10d-11	320	0.35d-10	0.74	139	0.23d-08	0.68	7
0.10d-13	640	0.91d-12	1.46	202	0.19d-08	0.82	8
0.10d-15	640	0.94d-12	1.51	202	0.16d-08	0.95	9

TABLE IV

Problem 2, HAGRON Results

	HAGRON, no continuation			HAGRON, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	68	0.13d-07	0.02	54	0.22d-07	0.04	3
0.10d-03	238	0.21d-10	0.13	101	0.79d-08	0.11	4
0.10d-05	1720	0.82d-08	0.67	131	0.62d-08	0.16	5
0.10d-07	20	0.93d-06	0.01[a]	428	0.68d-10	0.50	7
0.10d-09	1696	0.22d-09	1.08	9580	0.17d-12	5.25	8
0.10d-10	5296	0.34d-10	3.13				

[a] See text.

TABLE V

Problem 3, COLSYS Results

	COLNEW, no continuation			COLMOD, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	80	0.88d-08	0.07	70	0.14d-08	0.17	3
0.10d-03	56	0.24d-07	0.08	82	0.17d-08	0.25	4
0.10d-05	80	0.26d-08	0.15	83	0.16d-08	0.31	5
0.10d-07	160	0.58d-08	0.34	89	0.16d-08	0.38	6
0.10d-09	160	0.49d-09	0.40	98	0.15d-08	0.45	7
0.10d-11	320	0.22d-10	0.71	104	0.20d-08	0.48	7
0.10d-13	320	0.13d-09	0.82	137	0.19d-08	0.54	8
0.10d-15	640	0.14d-12	1.66	137	0.17d-08	0.62	9

TABLE VI

Problem 3, HAGRON Results

	HAGRON, no continuation			HAGRON, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	59	0.62d-09	0.02	50	0.74d-08	0.05	3
0.10d-03	207	0.17d-10	0.11	74	0.89d-07	0.10	5
0.10d-05	1769	0.14d-08	0.63	117	0.41d-07	0.14	6
0.10d-07	2030	0.37d-10	0.96	117	0.98d-09	0.17	7
0.10d-09	1691	0.22d-09	0.97	117	0.70d-07	0.22	9
0.10d-11				117	0.19d-08	0.30	11
0.10d-13				117	0.49d-07	0.35	13
0.10d-15				117	0.74d-07	0.41	15

TABLE VII

Problem 4, COLSYS Results

	COLNEW, no continuation			COLMOD, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	80	0.42d-08	0.09	69	0.28d-08	0.19	3
0.10d-03	160	0.10d-07	0.16	115	0.81d-09	0.33	4
0.10d-05	112	0.22d-07	0.18	128	0.42d-08	0.45	5
0.10d-07	284	0.20d-08	0.62	146	0.42d-07	0.60	6
0.10d-09	640	0.48d-10	1.79	172	0.82d-07	0.77	7
0.10d-11	2276	0.64d-09	5.03	312	0.61d-08	1.05	8
0.10d-13				364	0.62d-08	1.25	9
0.10d-15				388	0.68d-08	1.33	9

5.2. *Numerical Results*

In our numerical tests each problem is converted to first-order form. This is a necessary transformation for HAGRON. However, for COLNEW, the transformation does not take advantage of the flexibility of collocation methods and, indeed, COLNEW may well perform better when solving the test problems directly as second-order equations. Our motivation for solving all problems in first-order form lies in the fact that we believe a comparison of the performance of the two codes is more readily facilitated when solving problems in this form.

Both codes were allotted the same amount of storage space (500,000 words) which means that a maximum of 3288 mesh points is allowed in any mesh generated by COLNEW while the maximum mesh size for HAGRON is 12370. For each problem we place an error tolerance of 0.1d-07 on the solution. For COLNEW, the number of collocation points l is fixed at 4 for each problem. The initial mesh in both a non-continuation framework and a continuation framework is specified as a uniform mesh of 10 subintervals. The choice of a uniform initial mesh ignores the fact that an a priori analysis of linear singular perturbation problems can sometimes lead to knowledge of the location of the transition layers in the solution. Naturally, such knowledge would influence the choice of an initial mesh which, in cases of interest, would not be uniform. However, in our numerical tests we do not consider an a priori analysis of the problem, and this is for two reasons. First, we are interested in developing so-called "black box" methods where no analysis of the problem to be solved is required; instead, the method should efficiently and automatically determine where the transition layers lie. Second, an a priori analysis may be complicated and time consuming and, furthermore, may not always elicit information concerning the location of transition layers. In such situations we may be left with no alternative but to select a uniform initial mesh. However, the codes we discuss do have the option to allow a non-uniform initial mesh to be used and it will be of interest to see how such a facility can affect the performance of the codes in a non-continuation framework. We will consider the effect of supplying a non-uniform mesh in a non-continuation framework at the end of this section.

TABLE VIII

Problem 4, HAGRON Results

	HAGRON, no continuation			HAGRON, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	62	0.21d-08	0.03	46	0.13d-07	0.06	3
0.10d-03	161	0.19d-07	0.09	98	0.52d-06	0.11	4
0.10d-05	500	0.60d-08	0.19	124	0.32d-06	0.17	5
0.10d-07	1382	0.12d-08	0.51	150	0.10d-06	0.30	6
0.10d-09	1222	0.48d-09	0.57	200	0.12d-07	0.38	7
0.10d-11	1375	0.61d-09	0.93	263	0.29d-08	0.47	8
0.10d-13	5855	0.14d-09	4.37	263	0.12d-09	0.51	8
0.10d-15				368	0.37d-07	0.69	9

TABLE IX

Problem 5, COLNEW Errors on Shishkin Meshes

ε	$N = 10$	$N = 20$	$N = 40$	$N = 80$	$N = 160$	$N = 320$
0.10d-00	0.78d-05	0.44d-06	0.15d-07	0.50d-09	0.16d-10	0.51d-12
0.10d-01	0.13d-02	0.10d-03	0.35d-05	0.64d-07	0.75d-09	0.33d-10
0.10d-02	0.64d-02	0.16d-02	0.28d-03	0.31d-04	0.27d-05	0.13d-06
0.10d-03	0.12d-01	0.24d-02	0.45d-03	0.13d-03	0.28d-04	0.51d-05
0.10d-04	0.13d-01	0.31d-02	0.71d-03	0.13d-03	0.35d-04	0.92d-05
0.10d-05	0.13d-01	0.32d-02	0.80d-03	0.19d-03	0.42d-04	0.63d-05
0.10d-06	0.13d-01	0.32d-02	0.81d-03	0.20d-03	0.49d-04	0.12d-04
0.10d-07	0.13d-01	0.32d-02	0.81d-03	0.20d-03	0.50d-04	0.12d-04
0.10d-08	0.13d-01	0.32d-02	0.81d-03	0.20d-03	0.50d-04	0.13d-04

Since we know the exact solution for each of the test problems it is possible to calculate the maximum relative error for each problem. Thus, given a set of points, $x_1, x_2, \ldots, x_s$, we find

$$\text{Relative Error} = \max_{i=1,s} \left(\frac{|v_1(x_i) - u_1(x_i)|}{\max(1, |v_1(x_i)|)} \right). \tag{48}$$

The aim of the code HAGRON is to construct a discrete solution which satisfies the user-supplied error tolerances at the mesh points. Thus, for HAGRON, the relative error (48) is calculated using the set of mesh points in the final mesh for each problem. The aim of the code COLNEW is to form a continuous solution which satisfies the user-supplied error tolerances on the whole interval $[a, b]$. Therefore, for COLNEW, the relative error (48) is calculated using the set of mesh points in the final mesh for each problem, plus (arbitrarily) three equally spaced points in each mesh subinterval in the final mesh. In this manner, we are testing the degree of success of each code in fulfilling their objectives. The fact that HAGRON does not provide a continuous solution is an obvious disadvantage when compared with COLNEW.

In Tables I–VIII the column "largest mesh" denotes the largest mesh used when solving for a particular value of ε. This value provides an indication of the amount of storage space needed when solving a particular problem. All calculations in the numerical tests are performed in double precision on an IBM RS6000.

The behaviour of HAGRON in a non-continuation framework in the above tables warrants some explanation. In general, one would expect that as ε becomes smaller the code would have increasing difficulty in solving a particular singular perturbation problem. However, our numerical results reveal that this is not always the case in practice. For example, if we consider Table IV, the non-continuation times initially increase as ε becomes smaller; however, HAGRON then appears to solve the problem with ε = 0.10d-07 extremely quickly. This behaviour is explained by the fact that the transition layers for this problem have not been resolved; that is, the solution has only been resolved in smooth regions. This is because HAGRON attempts to satisfy an error estimate only at mesh points. It is sometimes possible for this error estimate to be satisfied even when no mesh points are in the region of rapid variation. In such cases the code never "sees" the non-smooth part of the solution. This difficulty should not occur if continuation is used.

As discussed previously, the choice of a uniform initial mesh in a non-continuation framework is not always necessary since an a priori analysis of certain singular perturbation problems leads to information about the location of transition layers in the solution. We consider a singular perturbation problem for which this is the case.

Problem 5. $\varepsilon y'' - y' = 0$, $y(0) = 1$, $y(1) = 0$.
Exact solution is $y(x) = (1 - \exp((x-1)/\varepsilon))/(1 - \exp(-1/\varepsilon))$. The solution has a boundary layer of width $O(\varepsilon)$ at $x = 1$.

This problem is one of a class of singular perturbation prob-

TABLE X

Problem 5, COLSYS Results

	COLNEW, no continuation			COLMOD, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-00	40	0.56d-09	0.04	36	0.73d-09	0.04	1
0.10d-01	40	0.81d-07	0.05	48	0.82d-09	0.09	2
0.10d-02	40	0.46d-06	0.06	52	0.82d-09	0.13	3
0.10d-03	84	0.11d-07	0.18	50	0.74d-09	0.17	4
0.10d-04	320	0.14d-10	0.74	54	0.68d-09	0.22	5
0.10d-05	1082	0.11d-10	2.00	56	0.10d-08	0.22	5
0.10d-06	2560	0.38d-11	6.01	56	0.82d-09	0.27	6
0.10d-07				56	0.83d-09	0.34	7
0.10d-08				56	0.16d-08	0.30	7

TABLE XI

Problem 5, No Continuation, Shishkin Initial Mesh

	COLNEW, $N = 10$			COLNEW, $N = 80$		
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time
0.10d-00	28	0.26d-08	0.05	80	0.11d-10	0.10
0.10d-01	40	0.99d-07	0.07	80	0.71d-08	0.07
0.10d-02	40	0.45d-06	0.10	80	0.38d-07	0.08
0.10d-03	126	0.97d-10	0.24	104	0.33d-08	0.21
0.10d-04	320	0.79d-08	0.73	244	0.17d-09	0.49
0.10d-05	1090	0.48d-13	2.53	626	0.52d-13	1.76
0.10d-06	160	0.46d-09	0.48	1472	0.74d-14	4.21
0.10d-07	1134	0.24d-13	3.16	752	0.19d-12	1.69

lems which has been extensively examined in the literature. In particular, Shishkin [10] has proposed a special piecewise equidistant mesh for the solution of such problems. Given an even number of subintervals N, a Shishkin mesh (for Problem 5) is formed by dividing the interval [0, 1] into two subintervals

$$[0, 1 - \sigma], \quad [1 - \sigma, 1].$$

Equidistant meshes with $N/2$ points are then formed on each subinterval. The parameter σ is proportional to ε and depends on the size of N and the coefficient functions appearing in the differential equation and for Problem 5 is given by $\sigma = 2\varepsilon \ln N$. We consider the solution of Problem 5 by COLNEW on Shishkin meshes for various N in Table IX. The results in Table IX reveal that the Shishkin meshes lead to an error of $O(N^{-2})$ in the COLNEW solution. Furthermore, for sufficiently small ε, it can be seen that the errors in the solution on the Shishkin meshes are uniform in ε.

In order to gauge the benefits of supplying a Shishkin mesh as an initial mesh in a non-continuation framework we will first consider the solution of Problem 5 by the COLSYS codes in both a non-continuation framework and a continuation framework with a uniform initial mesh of 10 subintervals.

We then consider the solution of Problem 5 by COLNEW in a non-continuation framework, but now with a Shishkin initial mesh. Results are given in Table XI.

With reference to Tables X and XI, it appears that there is no significant advantage in supplying a Shishkin mesh rather than a uniform mesh as the initial mesh for the COLNEW experiments. However, this observation should be tempered by the following two facts. First we note that in [11] it was demonstrated that the COLSYS error estimate may become highly inaccurate if neighbouring subinterval sizes differ by many orders of magnitude—this is exactly the situation with a Shishkin mesh. Second, any general purpose code inevitably contains some heuristic factors and, for example, the choice between a mesh selection and a mesh halving at some stage in the solution process may crucially affect the overall results. This is particularly so for very small ε and explains the rather non-uniform behaviour of COLNEW in Table XI.

As a conclusion to this section we specify the ε-sequence and mesh sequence generated by COLMOD and HAGRON in

TABLE XII

Problem 4, $\varepsilon_f = 0.100$d-15, COLMOD Results

j	ε_j	$\Phi_{j,d}H_{j,1}$	$\Phi_{j,d}H_{j,1}$	$\hat{\Phi}_{j,d}H_{j,1}$	$\Phi_{j,1}/\Phi_{j,1}$	$P_{j,1}/P_{j,2}$	Mesh sequence
1	0.100d-00	4.58	—	—	1.01	0.95	10, 40
2	0.131d-01	3.18	8.43	182.38	1.08	1.02	42, 65
3	0.190d-02	2.69	5.55	18.41	1.07	1.00	64, 92
4	0.604d-04	5.69	5.69	20.36	1.15	1.03	92, 117
5	0.839d-06	8.68	8.69	21.61	1.23	1.02	114, 132
6	0.515d-08	12.98	13.02	22.89	1.35	1.04	129, 149
7	0.142d-10	19.39	19.47	24.79	1.54	1.08	141, 156
8	0.193d-13	27.55	27.71	27.71	1.77	1.11	146, 180
9	0.100d-15	14.33	14.19	31.56	1.39	1.13	158, 176, 97, 194, 388

TABLE XIII

Problem 4, ε_f = 0.100d-15, HAGRON Results

j	ε_j	$\Phi_{j,d}H_{j,1}$	$\Phi_{j,d}H_{j,1}$	$\hat{\Phi}_{j,d}H_{j,1}$	$\Phi_{j,d}/\Phi_{j,1}$	$P_{j,1}/P_{j,2}$	Mesh sequence
1	0.100d-00	1.84	—	—	0.86	1.17	10, 26
2	0.133d-01	1.63	4.84	—	0.79	1.10	26, 46
3	0.111d-02	1.93	4.84	—	0.80	1.09	46, 76
4	0.195d-04	5.15	4.93	—	0.81	1.06	76, 108
5	0.162d-06	8.33	8.15	—	0.88	1.11	108, 142
6	0.725d-09	12.88	12.49	—	1.00	1.26	142, 200
7	0.192d-11	19.84	19.32	718.54	1.22	1.44	176, 263
8	0.296d-14	30.91	29.76	53.31	1.60	1.77	202, 368
9	0.100d-15	8.74	27.91	41.09	0.87	1.09	230, 234

a continuation framework for a specific problem. The problem we consider is Problem 4 with ε_f = 0.10d-15. In Tables XII and XIII the symbol $\hat{\Phi}_{j,d}$ denotes the value of the desired maximum monitor function value given by (43) before the safeguards (47) have been implemented.

6. CONCLUSIONS AND FUTURE WORK

The overall aim in developing our continuation strategy was to solve extremely difficult singular perturbation problems more efficiently than when solving without continuation. In particular, we had three main aims:

- Solve more problems.
- Use less storage space when solving problems.
- Solve problems faster.

If we consider the results for the four problems presented in this paper it is possible to see how successful we have been in our aims.

With regard to the first of the above aims, the smallest value of ε that we were able to solve each problem for without use of continuation was usually surpassed (or at least equalled) by the smallest value of ε when using continuation. The only exception is for Problem 2 when using HAGRON (the advantage of using continuation for this problem is still demonstrated by the fact that without using continuation the problem ε = 0.1d-07 is not resolved correctly by HAGRON).

Our results reveal that for most of the problems the difference between the largest mesh used in solving without continuation and that used when solving with continuation generally becomes larger as ε becomes smaller. In the most extreme cases, such as with Problem 1, we see that for COLNEW approximately 60 times more storage space is required in solving for ε = 0.1d-15 without continuation than with continuation.

It is observed that the advantages of using continuation in terms of speed of solution generally increase as ε becomes smaller. For relatively large values of ε there is usually no advantage in using continuation.

The results for the four problems highlighted in this paper are representative of the results we have obtained generally. We have tested our continuation algorithm on the fifteen problems found in [5] and found only one problem (Problem 7 in [5]) which was generally solved faster without using continuation than with continuation.

It is our intention to expand our continuation strategy to include a mesh prediction strategy. At the moment we have a

TABLE XIV

Problem 6, COLSYS Results

	COLNEW, no continuation			COLMOD, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-00	26	0.91d-08	0.18	34	0.10d-08	0.28	3
0.10d-01				41	0.68d-09	0.38	4
0.10d-02				41	0.78d-09	0.56	5
0.10d-03				42	0.12d-08	0.62	6
0.10d-04				42	0.15d-07	0.76	6
0.10d-05				48	0.11d-06	1.75	8

TABLE XV

Problem 7, COLSYS Results

	COLNEW, no continuation			COLMOD, automatic continuation			
ε	Largest mesh	Relative error	Total time	Largest mesh	Relative error	Total time	No. of steps
0.10d-01	80	0.16d-08	0.05	34	0.72d-09	0.31	4
0.10d-03	40	0.76d-08	0.13	48	0.74d-09	0.53	5
0.10d-05				52	0.82d-09	0.70	6
0.10d-07				52	0.82d-09	0.94	7
0.10d-09				54	0.83d-09	1.10	8
0.10d-11				56	0.76d-09	1.20	8
0.10d-13				58	0.82d-09	1.40	9
0.10d-15				59	0.76d-09	1.63	10

so-called static rezoning regime, which means that the best mesh for problem ε_j is used as the initial mesh for problem ε_{j+1}. An alternative approach would be to predict an appropriate mesh for ε_{j+1} and then use this as the initial mesh for ε_{j+1}.

We also intend to apply our strategy to nonlinear problems where, naturally, there is the added consideration of convergence of Newton's method when selecting the continuation parameter. Our strategy as it stands does not preclude the solution of non-linear problems; indeed, early investigations on non-linear problems have yielded extremely encouraging results. For example, consider the following two non-linear problems.

Problem 6. This problem is given in Hemker [9]. $\varepsilon y'' + (y')^2 = 1$, $y(0) = 1 + \varepsilon \ln \cosh(-0.745/\varepsilon)$, $y(1) = 1 + \varepsilon \ln \cosh(0.255/\varepsilon)$.
Exact solution is $y(x) = 1 + \varepsilon \ln \cosh((x - 0.745)/\varepsilon)$.
As $\varepsilon \to 0$, $y(x) \to 1 + |x - 0.745|$, the solution has a corner layer at $x = 0.745$.

Problem 7. $\varepsilon y'' - y - y^2 = -\exp(-2x/\sqrt{\varepsilon})$, $y(0) = 1$, $y(1) = \exp(-1/\sqrt{\varepsilon})$.
Exact solution is $y(x) = \exp(-x/\sqrt{\varepsilon})$.
The solution has a boundary layer of width $O(\sqrt{\varepsilon})$ near $x = 0$.

We present the results for the COLSYS codes for these problems. The continuation strategy remains exactly the same as for linear problems. The initial guess for the solution of problem ε_{j+1} is provided by the final solution of problem ε_j.

These results are typical of the nonlinear problems we have tried and demonstrate the importance of the availability of a good initial guess for difficult nonlinear singular perturbation problems. It is our intention to develop our continuation strategy further to include convergence of Newton's method as a criterion for parameter selection.

ACKNOWLEDGMENT

The authors are grateful to the referees for their helpful comments and suggestions.

REFERENCES

1. U. Ascher, J. Christiansen, and R. D. Russell, *ACM Trans. Math Software* **7,** 209 (1981).
2. U. Ascher, J. Christiansen, and R. D. Russell, COLSYS—"A Collocation Code for Boundary-Value Problems," in *Codes for Boundary Value Problems in Ordinary Differential Equations,* Lecture Notes in Comput. Sci. Vol. **76** (Springer-Verlag, Berlin, 1979), p. 164.
3. U. Ascher, R. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations* (Prentice–Hall, Englewood Cliffs, New Jersey, 1988).
4. G. Bader and U. Ascher, *SIAM J. Sci. Stat. Comput.* **8,** 483 (1987).
5. J. R. Cash, *Appl. Math. Comput.* **31,** 449 (1989).
6. J. R. Cash, *Comput. Math. Appl.* **9,** 257 (1983).
7. J. R. Cash and A. Singhal, *BIT* **22,** 184 (1982).
8. J. R. Cash and M. H. Wright, *Computing Science Tech. Report* **146,** AT& T Bell Laboratories, Murray Hill, NJ, 1989 (unpublished).
9. P. W. Hemker, *A Numerical Study of Stiff Two Point Boundary Value Problems,* MC Tract, Vol. **80,** (Math. Centrum, Amsterdam, 1977).
10. G. I. Shishkin, *Sov. J. Numer. Anal. Math. Modelling* **3,** 393 (1988).
11. R. Wright, J. Cash, and G. Moore, *Numer. Algorithms.* **7,** 205 (1994).