ARC-LENGTH CONTINUATION AND MULTI-GRID TECHNIQUES FOR NONLINEAR ELLIPTIC EIGENVALUE PROBLEMS

#197

Tony F.C. Chan[1] and H.B. Keller[2]

19 March 1981

ABSTRACT

We investigate multi-grid methods for solving linear systems arising from arc-length continuation techniques applied to nonlinear elliptic eigenvalue problems. We find that the usual multi-grid methods diverge in the neighborhood of singular points of the solution branches. As a result, the continuation method is unable to continue past a limit point in the Bratu problem. This divergence is analysed and a modified multi-grid algorithm has been devised based on this analysis. In principle, this new multi-grid algorithm converges for elliptic systems arbitrarily close to singularity and has been used successfully in conjunction with arc-length continuation procedures on the model problem. In the worst situation, both the storage and the computational work are only about a factor of two more than the unmodified multi-grid methods.

Abbreviated Title: Multi-Grid Continuation
Keywords: Multi-Grid, Arc-Length Continuation, Nonlinear Elliptic Eigenvalue Problems, Singular Systems.

[1]Computer Science Department, Yale University, Box 2158, Yale Station, New Haven, CT 06520. This author's work was supported by D.O.E. Contract EY-76-S-03-070 while at CalTech and by ONR Grant N00014-80-0076 under subcontract from Florida State University while at Yale.

[2]Applied Mathematics Department, CalTech, Pasadena, Ca 91125. This author's work was supported by the D.O.E. under Contract EY-76-S-03-070 and by the U.S.A.R.O. under Contract DAAG 29-78-C-0011.

Table of Contents

1. Introduction

Many problems of computational interest can be formulated as

$$G(u,\lambda) = 0, \tag{1.1}$$

where u represents the 'solution' (i.e. flow field, displacements, etc.) and λ is a real physical parameter (i.e. Reynold's number, load etc.) It is required to find the solution for some λ-intervals, that is a path of solutions, $[u(\lambda),\lambda]$. In this paper, we use a class of continuation based on parametrizing the solution branches by arc-length, say $[u(s),\lambda(s)]$. A main advantage of these arc-length continuation methods is that most singular points on the solution branches can be handled without much difficulty. Equations of the form (1.1) are called nonlinear elliptic eigenvalue problems if the operator G with λ fixed is an elliptic differential operator [2]. For nonlinear elliptic eigenvalue problems, a major portion of the computational work in the arc-length continuation methods is spent in solving large linear elliptic systems. In this paper, we investigate the use of multi-grid [4] methods for solving these linear systems. It turns out that a straight-forward implementation of the multi-grid methods fails in the neighborhood of the singular points and this usually prevents continuation past limit points. This failure is analyzed and a modified multi-grid method based on this analysis is devised. Even for very singular systems, the new multi-grid algorithm performes satisfactorily and never requires more than about twice the storage and computational work as the unmodified algorithm.

The arc-length continuation methods will be described in section 2 and

the multi-grid methods in section 3. In section 4, computational results for a model problem are presented, together with a description of the difficulties encountered by the multi-grid method near a limit point. The behaviour of the multi-grid method near singular points will be analyzed in section 5. The modified multi-grid algorithms designed to overcome these difficulties are described in section 6. The paper ends with a summary in section 7.

2. Newton's Method and Continuation Techniques

In this paper we are concerned with methods for computing a family or path of solutions of (1.1). The methods we employed will be based on some version of Newton's method.

2.1 Newton's Method

Given a value of λ and an initial guess u^0 for the solution $u(\lambda)$, we perform the following steps repeatedly until $||\delta u^i|| < \varepsilon$ is satisfied :

$$G_u^i \, \delta u^i = - G(u^i, \lambda) \tag{2.1}$$

$$u^{i+1} = u^i + \delta u^i. \tag{2.2}$$

In the above, subscripts denote partial derivatives and so G_u denotes the Jacobian of the operator G (with respect to u). This procedure will generally converge quadratically when it does converge. However, as is well known, in many instances it will fail to converge when the initial guess is not 'close' to the true solution.

2.2 Natural Continuation

A plausible procedure for overcoming this convergence difficulty and also for determining the dependence of u on λ is to start at a known solution (u_0, λ_0) on the solution curve and use it as initial guess for a Newton-type iteration to find the solution for a neighboring point on the solution curve with λ close to λ_0. The procedure is then repeated. We can improve on this by computing the derivative, u_λ, at a known solution and use it to get a better initial guess for the next value of λ in a predictor-corrector fashion.

We call this a natural continuation procedure because it corresponds to parametrizing the solution curve by λ, the naturally occurring parameter. A specific form of this is the more or less well-known:

Euler-Newton Continuation Procedure:

Given a known solution (u_0,λ_0), we compute the solutions at nearby values of λ as follows:

1. First compute the derivative u_λ at (u_0,λ_0) from

$$G_u\, u_\lambda = -\, G_\lambda. \tag{2.3}$$

2. Perform an Euler predictor step:

$$u^0 = u_0 + u_\lambda\, (\lambda - \lambda_0). \tag{2.4}$$

3. Use u^0 as initial guess in Newton's method :

$$G_u^i\, (u^{i+1} - u^i) = -\, G(u^i,\lambda) \tag{2.5}$$

until convergence.

4. Use $(u(\lambda),\lambda)$ as the new (u_0,λ_0) and go back to Step 1.

Note that the computation of the derivative u_λ does not cause much computational overhead because we usually have the factorization of the Jacobian G_u computed already in the Newton step. Using such a predictor-corrector method will often allow us to take a much bigger step in λ and thus reduce the overall cost of determining the dependence of u on λ.

Unfortunately, this procedure needs some modification in order to handle

general nonlinear systems because of the possibility of existence of nonunique solutions. The nonuniqueness usually manifests itself in the form of existence of 'singular' points where the Jacobian G_u is singular (see Figure 2-1). Points such as point A in Figure 2-1 are called limit points (or turning points) and points such as point B are called bifurcation points. These singular points are further characterized by the conditions that $G_\lambda \notin$ Range(G_u) at a limit point and that $G_\lambda \in$ Range(G_u) at a bifurcation point [12].

The difficulties that a natural continuation procedure will encounter at singular points are three-fold. First of all, since G_u is singular at these points, Newton's method will at best be linearly convergent, making it much more costly to compute the solution. Moreover, near a limit point, there may not exist a solution for a given value of λ (see Figure 2-2) and hence the iterations must fail to converge. Lastly, we need some mechanism for switching branches at a bifurcation point.

2.3 Arc-length Continuation

In the pseudo arc-length continuation approach [12], these difficulties are overcome by not parametrizing the solution u by λ. Instead, we parametrize the solution branches using an arc-length parameter s, and specify how far along the current solution branch we want to march.

To be more specific, we let s be the arc-length parameter, and treat u(s) and λ(s) as functions of s. We can compute the 'tangent' $[\dot{u}(s_0), \dot{\lambda}(s_0)]$ (where

Figure 2-1: A Typical Bifurcation Diagram

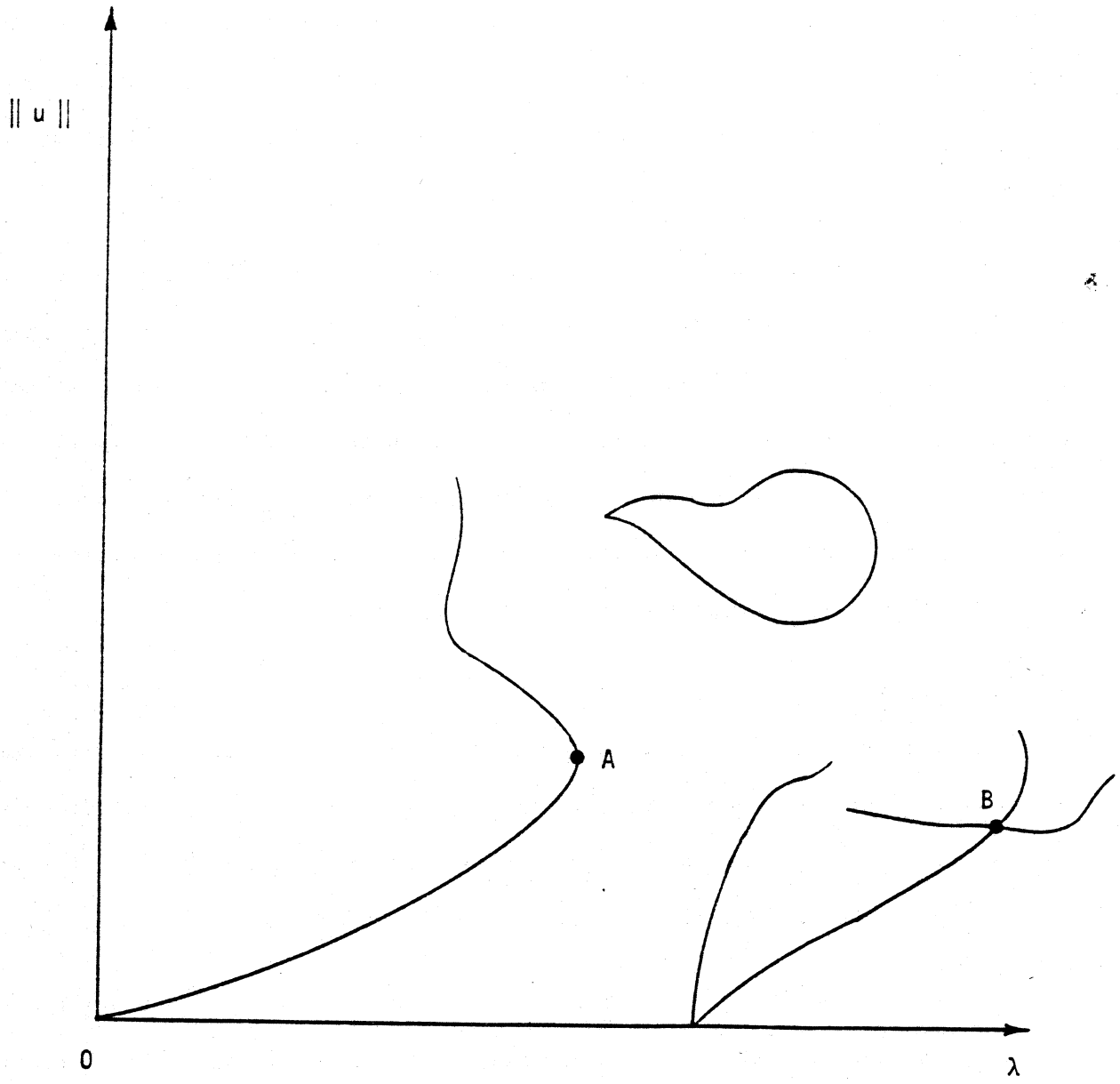
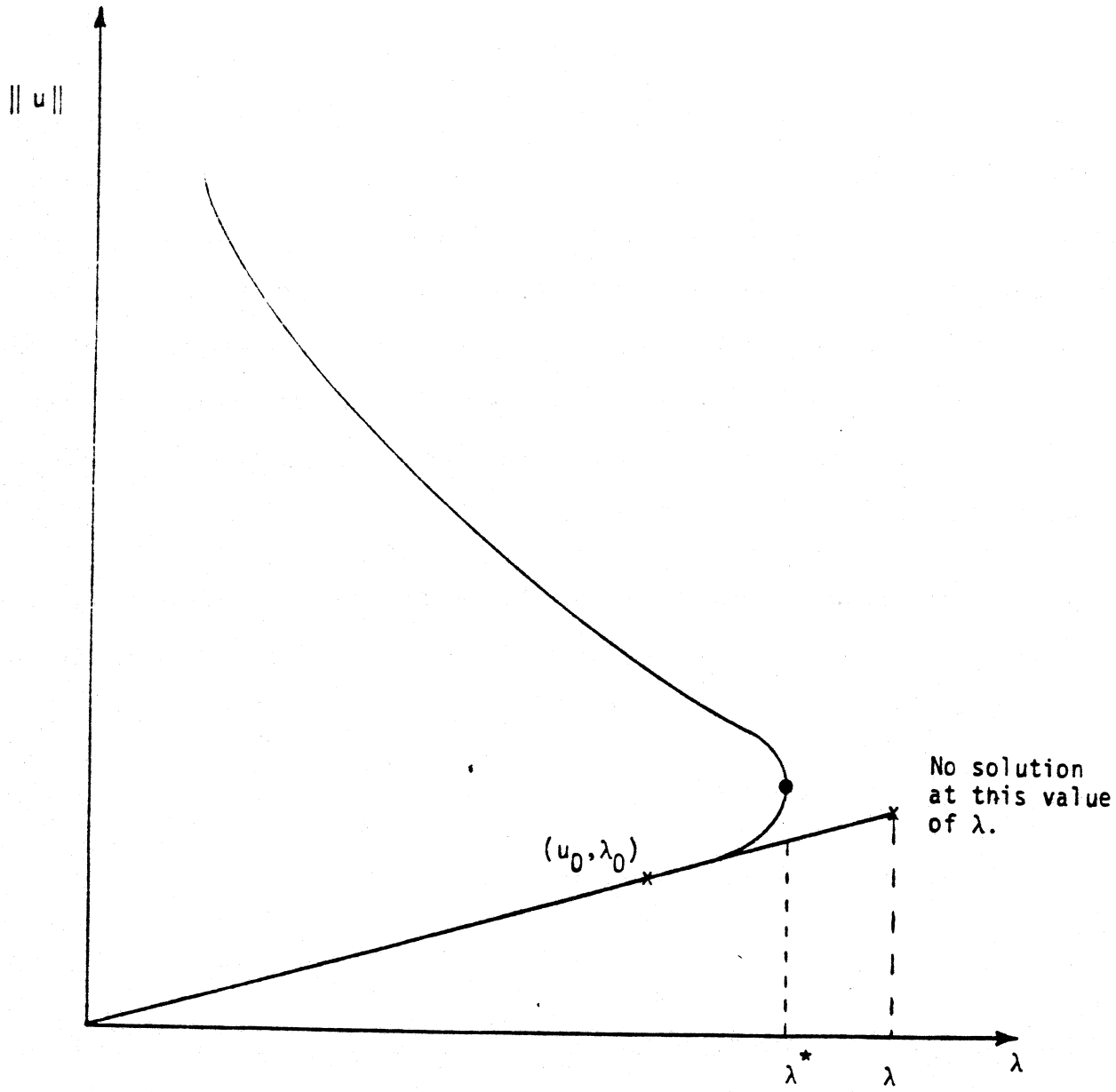


Figure 2-2: Failure of Natural Continuation Near Limit Points



the dots denote differentiation with respect to s) of a known solution at $s=s_0$ from the following two equations:

$$G_u \dot{u}_0 + \dot{\lambda}_0 G_\lambda = 0, \tag{2.6}$$

$$||\dot{u}_0||^2 + |\dot{\lambda}_0|^2 - 1 = 0. \tag{2.7}$$

Equation (2.6) is obtained from differentiating $G(u,\lambda) = 0$ with respect to s and (2.7) imposes the arc-length condition. We could theoretically generate the solution curve by integrating the initial value problem obtained by solving (2.6), (2.7) for $\dot{u}(s)$ and $\dot{\lambda}(s)$. Although this process is subject to the usual instabilities inherent in solving initial value problems approximately, it can be an extremely effective procedure. Indeed our pseudo arc-length continuation procedure can be viewed as a method for stabilizing Euler integration of (2.6), (2.7).

In the pseudo arc-length continuation procedure, we advance from s_0 to s along the tangent to the solution branch and require the new solution u(s) and $\lambda(s)$ to satisfy

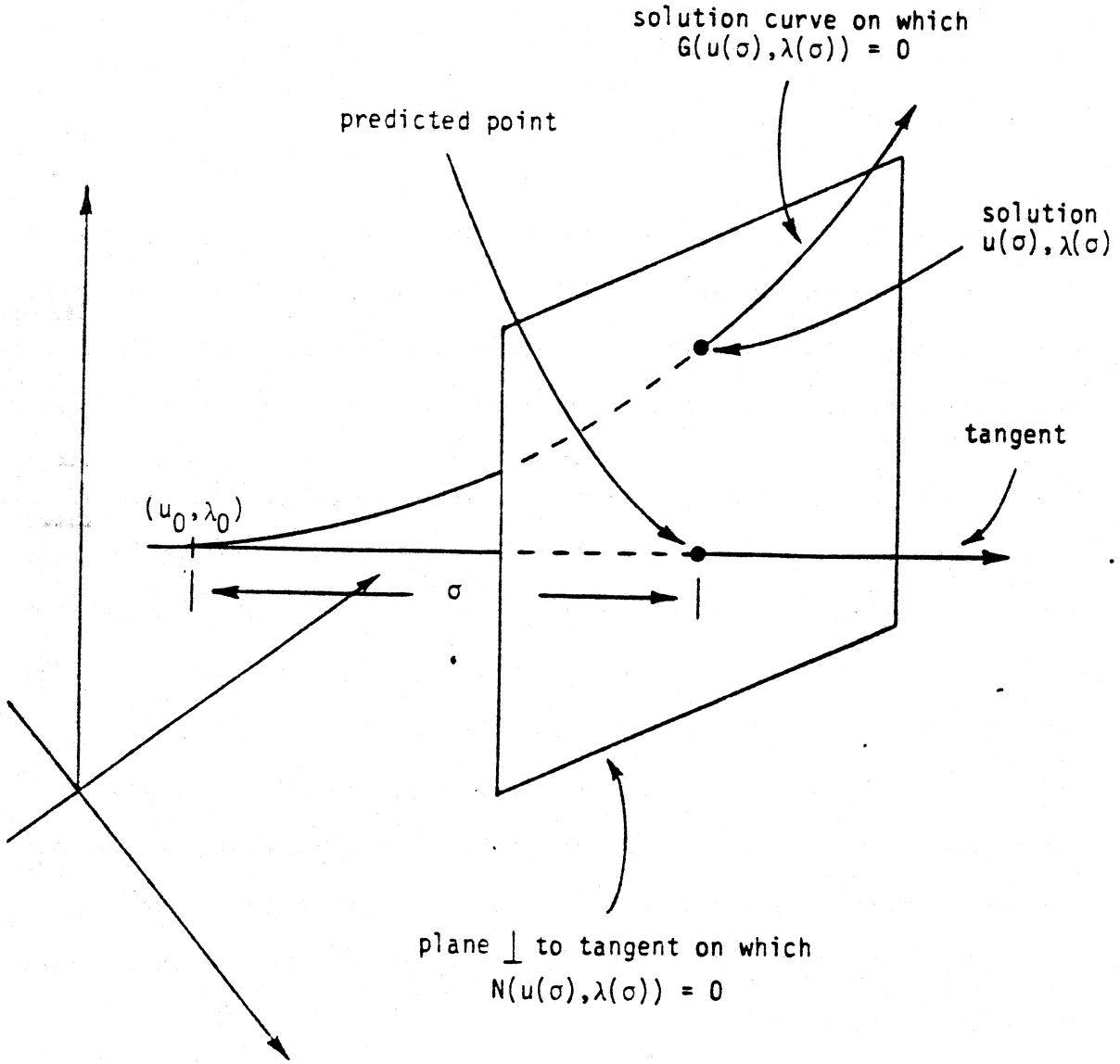
$$N(u(s),\lambda(s)) \equiv \dot{u}_0^T(u(s) - u(s_0)) + \dot{\lambda}_0(\lambda(s) - \lambda(s_0)) - (s - s_0) = 0. \tag{2.8}$$

In addition we require, of course:

$$G(u(s),\lambda(s)) = 0. \tag{2.9}$$

Equation (2.8) is the linearization of (2.7) and as indicated forces the new solution to lie on a hyperplane perpendicular to the tangent vector to the solution curve at s_0 and at a distance $(s-s_0)$ from it. Equation (2.9)

Figure 2-3: Pseudo Arc-length Continuation



requires $u(s)$ and $\lambda(s)$ to lie on the true solution curve (Figure 2-3). We now solve the coupled system (2.8) and (2.9) for $u(s)$ and $\lambda(s)$, given the step size $(s-s_0)$ (efficient strategies for choosing the step size are discussed in [23]). We use Newton's method, in which case we have to solve the following linear system at each iteration:

$$A \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix} = \begin{bmatrix} G_u & G_\lambda \\ N_u^T & N_\lambda \end{bmatrix} \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix} = - \begin{bmatrix} G \\ N \end{bmatrix} \tag{2.10}$$

It can be shown that at limit points, where G_u is singular and $G_\lambda \notin$ Range(G_u), the linear system in (2.10) is nonsingular (see [12]) and therefore Newton's method for the coupled system (2.8) and (2.9) is well-defined. Hence limit points present no problem and even quadratic convergence is achievable.

At bifurcation points, where G_u is singular and $G_\lambda \in$ Range(G_u), things are more complicated. In the simplest case of only one branch bifurcating from the main branch (simple bifurcation), an additional higher order condition involving G_{uu}, $G_{u\lambda}$ and $G_{\lambda\lambda}$ has to be satisfied. It can be shown [12] that this condition, together with (2.6) and (2.7) and the left and right null vectors of G_u, enable two solutions for $(\dot{u}_0, \dot{\lambda}_0)$ to be computed at a simple bifurcation point, with one solution corresponding to each branch. Using the appropriate pair of $(\dot{u}_0, \dot{\lambda}_0)$ in (2.8) allows branch switching. In [7] a more detailed study of the singular behaviour and branch switching at bifurcation is given.

In order to solve the linear system in (2.10) by direct methods, several approaches are possible. One way is to perform Gaussian Elimination on the inflated matrix A, with some form of pivoting to insure stability. But this approach completely ignores the sparse structure which is usually found in G_u's arising from nonlinear elliptic eigenvalue problems. In order to take advantage of the structure in G_u, Keller [12] suggested the following block-elimination procedure:

Algorithm BE: (block-elimination)

Solve $$G_u\, y = G_\lambda \tag{2.11}$$

and $$G_u\, z = -\, G. \tag{2.12}$$

Set $$\delta\lambda = (-N_u^T z - N)/(N_\lambda - N_u^T y) \tag{2.13}$$

and $$\delta u = z - \delta\lambda\, y. \tag{2.14}$$

Note that only systems with the coefficient matrix G_u have to be solved, so structures in G_u can be exploited. Moreover, only one factorization of G_u is needed. It has been shown [27] that even when G_u is becoming singular, Algorithm BE produces iterates that converge quadratically at limit points.

Continuation methods of various forms and levels of sophistication have been widely used in the engineering literature. For a recent survey of numerical methods for bifurcation problems, see for example [18]. The approach taken here is due to Keller [12], and has recently been applied to other problems in fluid mechanics ([5], [6], [15] , [16], [25], [27]). A related approach suggested by Abbott [1] corresponds (in a loose way) to

applying Algorithm BE to the matrix A with the last column permuted into the first n columns so that the corresponding coefficient matrix in Equations (2.11) and (2.12) becomes nonsingular even at limit points. However, as has already been pointed out, any structure or symmetry in G_u is lost in the process, and hence that approach seems unsuitable for large elliptic systems in two or three dimensions.

3. Multi-Grid Methods

3.1 Introduction

The class of multi-grid (MG) methods that we use here is based on work by Bakhvalov [3], Brandt [4], Federenko [8], Hackbush [10], and Nicolaides [19]. We shall only briefly describe here the particular MG algorithms that we have used for linear elliptic problems that arise in our treatment of nonlinear elliptic eigenvalue problems.

The particular way in which we use the MG idea is to use a hierarchy of grids, rather than a single one, in order to speed up the convergence rate of the solution process. The MG process has some very desirable theoretical properties: for certain elliptic operators on an n by n grid, it computes the approximate solution to truncation error accuracy in $O(n^2)$ arithmetic operations and $O(n^2)$ storage. It seems natural to consider the use of MG methods for solving nonlinear eigenvalue problems. MG methods have been applied to solution of linear eigenvalue problems by Hackbush [11] and McCormick [17].

3.2 The 'Cycle C' MG Algorithm

The particular MG algorithm that has been used in this study is based on the 'Cycle C' algorithm described in Brandt [4]. This is an algorithm for iteratively solving the discrete equations approximating a linear elliptic problem on a given grid, through interaction with a hierarchy of coarser grids, taking advantage of the fact that the different discretizations on the

different grids are all approximations to the same continuous problem. We note that there are other MG algorithms [4] proposed for implementing continuation procedures outside of the context of the pseudo arc-length framework. Some potential problems with these related algorithms are discussed in section 3.4. We do not know how well such MG algorithms perform and we hope to carry out our own investigation on such related methods in the future. In this paper, MG algorithms are used to solve the fine grid discrete equations that arise in the pseudo arc-length continuation procedure.

Consider a hierarchy of grids $(G^0, G^1, \ldots, G^M)$, with G^M being the finest one, defined on a domain Ω with corresponding mesh sizes $(h_0 > h_1 > \ldots > h_M)$, and all approximating the same linear elliptic problem :

$$\begin{aligned} L\,U &= F \quad \text{on } \Omega \\ U &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{3.1}$$

The discrete equation on a grid G^k is written as:

$$\begin{aligned} L^k\,U^k &= F^k \quad \text{on } G^k \\ U^k &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{3.2}$$

We are primarily interested in obtaining the approximating solution U^M on the finest grid, and we shall start with an initial guess on G^M and apply a standard relaxation procedure such as the Gauss-Seidel procedure. It is well known that the error is reduced rapidly in the first few iterations but then the reduction rate becomes very slow. By a frequency analysis, it can be shown that the fast reduction occurs when the residual (or the error) in the current iterate has large harmonics on the scale of the grid, the so-called

high-frequencies. Now at a stage in the iterative process where the error reduction rate slows down, let the current iterate be u^M. Define the error v^M in the iterate as $v^M = U^M - u^M$. Then the error v^M satisfies the following equation:

$$L^M v^M = F^M - L^M u^M \equiv R^M \quad \text{on } G^M , \qquad (3.3)$$
$$v^M = 0 \quad \text{on } \partial G^M .$$

The residual R^M is computable and hence the original problem of solving for U^M can be reduced to an equivalent one of solving (3.3) for v^M. There seems to be no obvious advantage in using (3.3) rather than continuing with the original relaxation procedure with u^M. However, if the error v^M and the residual R^M are <u>smooth</u> relative to G^M, that is, if their high frequency components have been smoothed out by the relaxation procedure, then we can <u>approximate</u> the solution of (3.3) on a <u>coarser</u> grid, say G^{M-1}, by solving :

$$L^{M-1} v^{M-1} = F^{M-1} \equiv I_M^{M-1} R^M \quad \text{on } G^{M-1}, \qquad (3.4)$$
$$v^{M-1} = 0 \quad \text{on } \partial G^{M-1} ,$$

After this problem is solved we can interpolate the solution v^{M-1} onto G^M to get:

$$\text{new } u^M = \text{old } u^M + w_{M-1} I_{M-1}^M v^{M-1} , \qquad (3.5)$$

where w_{M-1} is an interpolation factor, normally taking the value unity and I_i^j stands for some interpolation operator from G^i to G^j. The solution process for Equation (3.4) on G^{M-1} usually costs considerably less than the cost of solving Equation (3.3) on G^M. If v^M is indeed smooth (relative to G^M), then G^{M-1} should provide adequate resolution for v^M and hence $I_{M-1}^M v^{M-1}$ should be a good approximation for v^M. This principle of transferring to a coarser grid

when convergence slows down can be applied recursively. Thus for example, we can start with a zero initial guess for v^{M-1} in Equation (3.4) and apply the Gauss-Seidel relaxation procedure to the iterates on G^{M-1}. When the convergence slows down, we can again transfer to the next coarser grid G^{M-2}, and so on. One can view the whole process as each grid smoothing just those frequencies in the error that are high relative to its own mesh size, each doing its job efficiently because these high frequencies are precisely those that are efficiently smoothed out by relaxation procedures.

The control of when to transfer between grids can follow a fixed strategy or an adaptive one. A fixed strategy could be of the following kind (see Nicolaides [19]) : perform p relaxation sweeps on each grid G^k before transferring to a coaser grid G^{k-1}, and perform q relaxation sweeps before interpolating back to a finer grid G^{k+1}. An adaptive strategy could be as follow (see Brandt [4]) : transfer to a coaser grid when the ratio of the residual norm of current iterate to the residual norm a sweep earlier is greater than some tolerance η, and transfer to a finer grid when the ratio of the residual norm of current iterate to the residual norm on the next finer grid is less than another tolerance δ. For simple problems like Poisson's equation on a square, the overall MG efficiency is very insensitive to which particular strategy is used and what values are used for (p,q) or (η,δ). We shall refer to the above particular fixed strategy the (p,q) strategy and the adaptive strategy the (η,δ) strategy.

3.3 Indefinite Problems

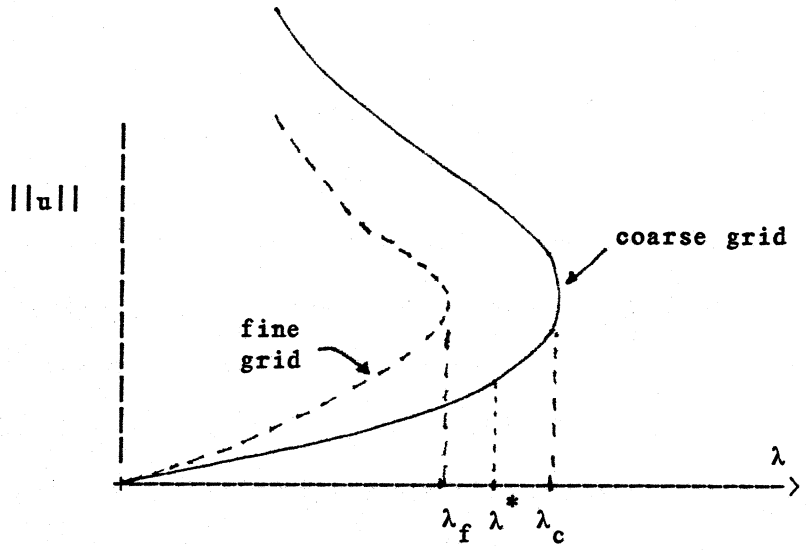
In the 'Cycle C' algorithm just described, convergence on the lowest (coarsest) grid G^0 is obtained by repeated relaxation sweeps. For positive definite matrices, convergence on G^0 can be guaranteed. For indefinite problems, however, convergence on G^0 cannot be obtained by repeated relaxation sweeps, because the components of the error that correspond to eigenfunctions with negative eigenvalues will grow as a result of relaxation sweeps (see the analysis in section 5). Therefore, for indefinite problems, a direct solution (e.g. Gaussian Elimination) must be employed on the coarsest grid. If this coarsest grid is fine enough, it will also provide corrections to those growing components of the iterates on all finer grids. However, too fine a grid for G^0 will increase the cost of the direct solution procedure. Hence a little care must be taken regarding the size of the coarsest grid for indefinite problems. Fortunately, for 'not too indefinite' problems, G^0 can be chosen coarse enough so that the direct solution on G^0 will not affect the overall efficiency of the MG procedure seriously. Since indefinite problems occur frequently in nonlinear elliptic eigenvalue problems and, in particular, in our model problem, we shall use such a direct solution on G^0 whenever necessary.

3.4 Continuation Methods

Brandt [4] suggested using continuation methods in conjunction with the MG procedure. His main idea is to use coarse grids for continuation, with little work and crude accuracy, and only use the finer grids at the final

continuation step to achieve higher accuracy. We have not pursued this idea here. We believe that it will work as long as we stay away from singular points. Around a limit point, however, the solution branches corresponding to different grids may look like the situation in Figure 3-1. If we continue on the coarse grid to λ^* and try to refine using the finer grid, while keeping λ^* fixed, we cannot hope to obtain a fine grid solution because λ^* is larger than the fine grid limit point λ_f (i.e. no fine grid solution exists for $\lambda > \lambda_f$). In the opposite case, there is no coarse grid solution at λ^* so we cannot get started on that grid. Hence, in general, we have to be extremely careful in using MG methods and continuation around singular points.

Figure 3-1: Limit Points for Different Grids



4. Application to the Bratu Problem

4.1 Bratu's Problem

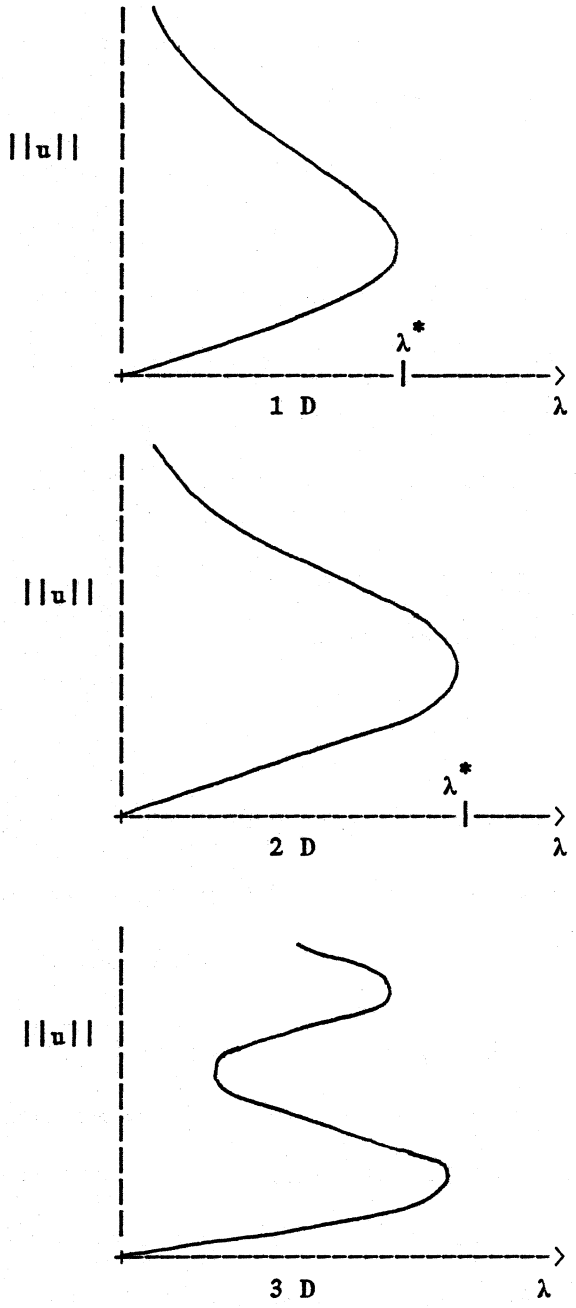
As a typical example of an nonlinear elliptic eigenvalue problem, we consider the Bratu problem :

$$G(u,\lambda) = \Delta u + \lambda e^{u} = 0 \qquad \text{on } \Omega, \tag{4.1}$$

$$u = 0 \qquad \text{on } \partial\Omega.$$

Equation (4.1) arises in many physical problems, for example, in chemical reactor theory, radiative heat transfer, and in modelling the expansion of the universe. The domain Ω is the unit interval $[0,1]$ in R^1, or the unit square $[0,1]x[0,1]$ in R^2, or the unit cube $[0,1]x[0,1]x[0,1]$ in R^3. There are no bifurcation points in this problem, all the singular points are limit points. The behaviour of the solution near the singular points has been studied numerically [1, 26] and theoretically [14, 20, 21, 24]. Typical solution diagrams are shown in Figure 4-1. For both the one and two dimensional cases, the problem has exactly one limit point, whereas the three dimensional case has infinitely many limit points (if Ω is a sphere). From now on we only consider the two dimensional case, with Ω the unit square. For this case, the value of λ^* and the corresponding $||u||_\infty$ at the limit point are given by : $\lambda^* \cong 6.81$ and $||u||_\infty = u(0.5,0.5) \cong 1.39$. For $\lambda > \lambda^*$, Equation (4.1) has no solution, and for $\lambda < \lambda^*$, it has exactly two solutions.

Figure 4-1: Solution for the Bratu Problem



4.2 Arc-length Continuation with Direct Methods

We first apply the arc-length continuation method of Section 2 to Equation (4.1) using direct methods. For this problem, a trivial solution is $(u = 0, \lambda = 0)$. We can thus start at this trivial solution on the lower branch and march along the solution branch, past the limit point, and continue on to the upper solution branch. Since the only singular point in this problem is a limit point, this in principle presents no problem to the arc-length continuation procedure, although the step size might have to be reduced and controlled appropriately near the limit point. If desired, the limit point can be accurately determined by other related techniques [1, 13].

The derivatives of the operator G in Equation (4.1) that are needed for the arc-length continuation technique are :

$$G_u = \Delta + \lambda e^u , \tag{4.2}$$

$$G_\lambda = e^u . \tag{4.3}$$

Now if we approximate the Laplacian operator by the standard five-point stencil on a uniform grid, the operator G_u will be approximated by the usual block tridiagonal matrix and the operator G_λ by a column vector.

In the application of the arc-length continuation technique, we will have to repeatedly solve linear systems of equations with the matrix given by G_u. The solution of these linear systems is the central part of the arc-length continuation method. Hence, an efficient linear system solver is crucial to the overall performance of the continuation technique. In this section, we present some computational results for Bratu's Problem using a direct method

(Gaussian Elimination) of solution of the linearized difference equations. For large problems, this would be prohibitively expensive. However, the results here are intended to demonstrate the performance of the continuation procedure independent of the linear algebra method employed. In the next section, we shall investigate the use of multi-grid methods for solving the linear equations. It should be pointed out that G_u is generally not separable, and therefore we cannot use fast Poisson solvers directly even on rectangular domains. Moreover, this matrix is indefinite on the upper branch, and hence iterative methods like Successive-Over-Relaxation cannot be used directly.

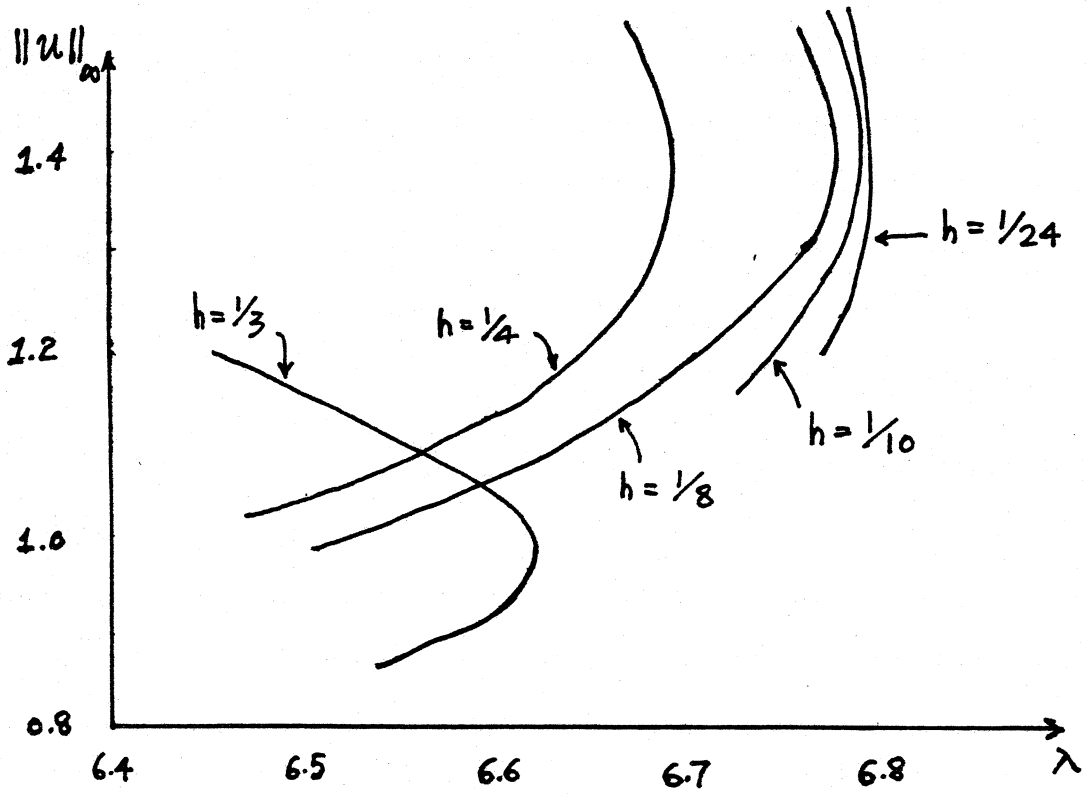
We present some of our computed results in Figure 4-2. Only the behaviour of the solution branch near the limit point for a few relatively coarse discretizations are presented. These are to be compared with the values : $\lambda^* = 6.80811698$ and $u(.5,.5) = 1.3916603$ for a grid with $h = 1/24$ with the nine-point finite difference operator as computed by Abbott [1] and to the easily obtainable exact solution ($\lambda^* = 18/e \tilde{=} 6.62183$, $u^* = 1$) for the case $h = 1/3$. As expected, the step size $\partial s = s - s_0$ had to be suitably controlled near the limit point, but otherwise we encountered no difficulty in continuing past the limit point.

4.3 Arc-length Continuation with Multi-Grid Methods

In this section we discuss the use of MG methods, rather than direct methods, for solving the linear equations that arise in the continuation procedure. The MG method that we use was described in Section 3 and

Figure 4-2: Computed Results for Bratu's Problem Near Limit Point

h	λ	u	$\dot{\lambda}$	
1/3	6.000000	0.619061	0.9841	
	6.485170	0.809435	0.9165	
	6.572858	0.883052	0.7948	
	6.621830	0.999899	2.8889E-4	← limit point
	6.614022	1.04937	-0.4207	
1/24	6.500000	1.00456	0.9632	
	6.689007	1.14350	0.9041	
	6.802681	1.34995	0.2965	
	6.805499	1.39043	-1.1732E-4	← limit point
	6.805485	1.39368	-0.0125	



Gauss-Seidel is the smoothing relaxation process. Since the Jacobian matrix G_u becomes indefinite on the upper branch, we use a direct method on the coarsest grid in the neighborhood of the the limit point and on the upper branch.

We started the continuation procedure with the trivial solution ($u = 0$, $\lambda = 0$), with $h = 1/4$ on the coarsest grid, and a total of four levels of grids, making the finest grid with $h = 1/32$. As expected, the MG method worked fine and we were able to continue up to very close to the limit point, at $\lambda \tilde{=} 6.804$ on the lower branch. However, we noticed that the convergence of the MG method deteriorates as we move in towards the limit point. For example, the number of equivalent relaxation sweeps on the finest grid required to reduce the residual norm by an order of magnitude, which is a convenient way of measuring the efficiency of MG methods, went from about 5 at $\lambda = 0$ to about 20 at $\lambda = 6.803$ and to <u>divergence</u> at $\lambda = 6.805$. The divergence occurred in the MG method and not in the Newton iteration. It is <u>not</u> due to the possible indefiniteness of the Jacobian matrix on the finest grid. This can occur near the limit point after a large Euler-predictor step. But we performed other tests starting on the upper branch, away from the limit point, where the Jacobian matrix is indefinite, and the MG method performed as efficiently as on the lower branch. From our experience, this divergence is strictly a phenomenon associated with the limit point, and to the best of our knowledge, has never been discussed or analysed in the literature. We study this effect in section 5.

The exact value of λ at which this divergence first occurs varies slightly with the size of the coarsest grid h_0, but is quite independent of the other parameters of the Cycle C algorithm (e.g. η and δ). In all the cases we have run, this divergence made it impossible to continue past the limit point. Therefore, a remedy is needed. Before we can find one, we must understand the reason for the divergence.

5. Analysis of Multi-Grid Methods for Near-singular Systems

For the present analysis, we assume that the linear operator L is self-adjoint and has the complete set of orthonromal eigenfunctions $\{\xi_1, \xi_2, \ldots\}$ with corresponding <u>real</u> eigenvalues $\{\mu_1 \leq \mu_2, \ldots\}$. The operator G_u in the Bratu problem clearly satisfies the above hypothesis. Thus the solution U to L U = F can be written as:

$$U = \sum_{i=1}^{\infty} a_i \xi_i; \quad a_j = \langle \xi_j, F \rangle, \; j=1,2,\ldots \tag{5.1}$$

We assume that the discrete approximations L^k to the continuous L are <u>symmetric</u>. Thus they have <u>real</u> eigenvalues $\{\mu_1^k \leq \mu_2^k \leq \ldots \leq \mu_{N_k}^k\}$ and a complete set of orthonormal eigenvectors $\{\xi_1^k, \xi_2^k, \ldots, \xi_{N_k}^k\}$. Here N_k is the dimension of the matrix representing L^k. For most reasonable approximations, and certainly for the five point formula used for the Bratu Problem on a rectangle this is true.

Assume that after iterating (relaxing) on the grid G^k, convergence has slowed down and a transfer to the next coarser grid is desired. Let the current iterate be u^k, and the corresponding 'correction' be v^k so that $U^k = u^k + v^k$ where U^k satisfies $L^k U^k = F^k$. The correction problem is given (as in section 3) by:

$$L^k v^k = R^k = F^k - L^k u^k, \text{ in } G^k; \; v^k = 0 \text{ on } \partial G^k. \tag{5.2}$$

This is approximated on G^{k-1} by

$$L^{k-1} v^{k-1} = I_{k-1}^k R^k, \text{ in } G^k; \; v^{k-1} = 0 \text{ on } \partial G^{k-1}. \tag{5.3}$$

Using the eigenvector expansion of v^k in (5.2) we get:

$$v^k = \sum_{i=1}^{N_k} a_i^k \xi_i^k, \tag{5.4}$$

where

$$a_i^k = \langle R^k, \xi_i^k \rangle \; / \; \mu_i^k; \quad i=1,\ldots,N_k. \tag{5.5}$$

Suppose now that (5.3) is solved <u>exactly</u> (by either direct solution or Cycle C or any other means) on G^{k-1}. The solution v^{k-1} is then:

$$v^{k-1} = \sum_{i=1}^{N_{k-1}} a_i^{k-1} \; \xi_i^{k-1} \; , \tag{5.6}$$

where

$$a_i^{k-1} = \langle I_k^{k-1} R^k, \xi_i^{k-1} \rangle \; / \; \mu_i^{k-1}. \tag{5.7}$$

The key idea in the MG method is that if v^k and R^k are smooth enough, they can be well approximated on G^{k-1}. Thus it is important for efficiency considerations that[3]

$$I_{k-1}^k \; v^{k-1} \tilde{=} v^k. \tag{5.8}$$

Using (5.4) and (5.6), this is equivalent to:

$$\sum_{i=1}^{N_{k-1}} a_i^{k-1} \; I_{k-1}^k \; \xi_i^{k-1} \tilde{=} \sum_{i=1}^{N_k} a_i^k \; \xi_i^k. \tag{5.9}$$

This will be the case if

$$\text{(a)} \quad I_{k-1}^k \; \xi_i^{k-1} \tilde{=} \xi_i^k, \; 1 \leq i \leq N_{k-1} \; , \tag{5.10}$$

$$\text{(b)} \quad a_i^{k-1} \tilde{=} a_i^k, \qquad 1 \leq i \leq N_{k-1}, \tag{5.11}$$

---

[3] We shall use the '$\tilde{=}$' symbol to mean rather loosely 'approximately equal to'. The meaning should be clear by context. Also, we shall assume that the interpolation factor w_{k-1} in Equation (3.5) is equal to one unless stated otherwise.

$$(c)\quad a_i^k \cong 0, \quad i > N_{k-1}. \tag{5.12}$$

Conditions (5.10) and (5.11) ensure that the coarse grid correction v^{k-1} improves the lower modes of the iterate u^k. Condition (5.12) is essentially the smoothness required of v^k on G^k (i.e. negligible higher modes).

Now condition (5.10) is satisfied for the low frequency eigenfunctions of the continuous operator L if the grids G^k and G^{k-1} are both fine enough to resolve these eigenfunctions. This holds in many cases since the lower eigenfunctions of most second order elliptic operators over smooth domains are very smooth. For the Bratu problem, the eigenfunctions are very close to products of sines and cosines (the eigenfunctions of the Laplacian operator) and so the lower modes are easily resolved by very coarse grids. Condition (5.11), on the other hand, turns out to be violated if the operator L^k is near singular. This is what caused the divergence of the Cycle C algorithm in the arc-length continuation procedure as we approach the limit point (see section 4.3). We shall analyse this case next.

From (5.5) and (5.7), condition (5.11) becomes:

$$\langle I_k^{k-1} R^k, \xi_i^{k-1} \rangle \;/\; \mu_i^{k-1} \cong \langle R^k, \xi_i^k \rangle \;/\; \mu_i^k,$$
$$i \leq i \leq N_{k-1}. \tag{5.13}$$

We claim that if condition (5.10) is satisfied, and if the transfer from G^k to G^{k-1} is done only after the residual R^k has been smoothed, then the numerators in (5.13) will have approximately the same value. To show this, we expand R^k as

$$R^k = \sum_{i=1}^{N_k} r_i \, \xi_i^k, \tag{5.14}$$

where

$$r_i = \langle R^k, \xi_i^k \rangle. \tag{5.15}$$

Thus the numerator on the right hand side of (5.13) is precisely r_i. To estimate the numerator on the left hand side of (5.13), we proceed as follows:

$$I_k^{k-1} \, R^k = \sum_{i=1}^{N_k} r_i \, I_k^{k-1} \, \xi_i^k \, ,$$

$$= \sum_{i=1}^{N_{k-1}} r_i \, I_k^{k-1} \, \xi_i^k + \sum_{i=N_{k-1}+1}^{N_k} r_i \, I_k^{k-1} \, \xi_i^k. \tag{5.16}$$

Now if condition (5.10) holds, its converse:

$$I_k^{k-1} \, \xi_i^k \cong \xi_i^{k-1}, \; 1 \leq i \leq N_{k-1}, \tag{5.17}$$

also holds. Also, if R^k has been smoothed on G^k, then r_i [for $N_{k-1} < i \leq N_k$] must be small compared with r_i [for $1 \leq i \leq N_{k-1}$]). Alternatively (5.12) assumes $a_i^k = r_i/\mu_i^k \cong 0$ for $i > N_{k-1}$. Therefore, we can approximate in (5.16) by dropping the second sum on the right hand side to get

$$I_k^{k-1} \, R^k \cong \sum_{i=1}^{N_{k-1}} r_i \, \xi_i^{k-1}. \tag{5.18}$$

Hence

$$\langle I_k^{k-1} \, R^k, \, \xi_i^{k-1} \rangle \cong \; r_i, \; 1 \leq i \leq N_{k-1}. \tag{5.19}$$

Therefore, from (5.15) and (5.19), we have, as claimed earlier,

$$\langle I_k^{k-1} R^k, \xi_i^{k-1} \rangle \cong \langle R^k, \xi_i^k \rangle \text{ for } 1 \leq i \leq N_{k-1}. \tag{5.20}$$

The relations in (5.20) imply that condition (5.13) will be true if

$$\mu_i^k \;/\; \mu_i^{k-1} \simeq 1, \quad 1 \leq i \leq N_{k-1}. \tag{5.21}$$

Actually, these conditions need to be <u>strengthened</u> in order to guarantee that the visit to G^{k-1} actually <u>improves</u> the accuracy of u^k. This can be seen as follows. The error in the iterate u^k <u>before</u> the transfer to G^{k-1} is given by

$$\text{old error} \equiv v^k = \sum_{i=1}^{N_k} a_i^k \, \xi_i^k. \tag{5.22}$$

From (3.5), the new error in u^k <u>after</u> coming back from a visit to G^{k-1} is given by

$$\text{new error} = v^k - w_{k-1} \, I_{k-1}^k \, v^{k-1}. \tag{5.23}$$

In view of (5.4) and (5.6), the above gives:

$$\begin{aligned} \text{new error} &\simeq \sum_{i=1}^{N_{k-1}} (a_i^k - w_{k-1} a_i^{k-1}) \xi_i^k + \text{higher modes} \\ &\simeq \sum_{i=1}^{N_{k-1}} (1 - w_{k-1} \, a_i^{k-1} \;/\; a_i^k) \, a_i^k \, \xi_i^k + \text{higher modes} \end{aligned} \tag{5.24}$$

From (5.5), (5.7) and (5.20), we have

$$a_i^{k-1} \;/\; a_i^k \simeq \mu_i^k \;/\; \mu_i^{k-1},$$

and therefore we can write the new error in (5.24) as:

$$\text{new error} \simeq \sum_{i=1}^{N_{k-1}} (1 - w_{k-1} \mu_i^k / \mu_i^{k-1}) a_i^k \xi_i^k + \text{ higher modes.} \tag{5.25}$$

For obvious efficiency and convergence considerations, the new error should preferably be less than the old error, at least for the lower modes. In other words, condition (5.21) should be strengthened to

$$|1 - w_{k-1} \mu_i^k / \mu_i^{k-1}| < 1, \tag{5.26}$$

i.e.

$$0 < w_{k-1} \, \mu_i^k / \mu_i^{k-1} < 2, \qquad \text{for } 1 \leq i \leq N_{k-1}. \tag{5.27}$$

Now if the ratios of eigenvalues in (5.21) are not close to unity, the interpolation factors, w_{k-1}, should be chosen so that condition (5.27) is satisfied. Otherwise the new error can be larger than the old error in some modes.

It should be pointed out that, in general, condition (5.27) is not necessary for the convergence of the Cycle C algorithm. This is the case, for instance, if L and the L^k's are all positive definite. Then Gauss-Seidel sweeps on any grid G^k will reduce the amplitude of every mode present in the error. In such cases, convergence on any grid can be achieved by merely doing enough relaxation sweeps. Then it is not necessary for the next coarser grid to provide any improvement on the current iterate, although it would obviously improve the efficiency of the overall algorithm if it does so. In fact, the MG method derives its efficiency from the very fact that the coarser grids do provide improvements in the current iterate u^k in the lower modes. These are precisely those modes that have poor convergence rates for the relaxation sweeps on G^k. Thus, even in the positive definite case, it is important (from an efficiency viewpoint) that conditions (5.27) hold, at least for small i's.

If the operator L and the L^k's are indefinite the situation is different because some modes will grow if we simply perform relaxation sweeps on a fixed grid. Such modes have to be corrected by going to coarser grids and using a direct method on the coarsest grid. Further the interpolation factors, w_{k-1}, should be chosen such that condition (5.27) is satisfied for these modes. Condition (5.27) has been suggested by Brandt [4] for indefinite problems.

However as we show later, most nonlinear eigenvalue problems with limit points and bifurcation points abound with indefinite operators but they do not cause difficulties in the sense of violating condition (5.27). Essentially only one mode causes problems on each G^k and it is the mode that correspond to the eigenvalue that is nearest zero as the singular point is approached. Merely including the interpolation factors so that condition (5.27) is satisfied turns out to be very inefficient. Further, it is not clear that such factors, w_{k-1}, can be found at all in this case.

Another source of difficulty is that the process of interpolating v^{k-1} into G^k introduces high frequency errors. That is, the exact relation corresponding to (5.10) is:

$$I_{k-1}^k \xi_i^{k-1} = \xi_i^k + \sum_{j=1}^{N_k} b_{ij}^k \xi_j^k , \quad i = 1,2,\ldots,N_{k-1};$$

$$\text{for } 1 \leq i \leq N_{k-1}, \tag{5.28}$$

and the coefficients b_{ij}^k may be large for $j > N_{k-1}$. This would result in a violation of (5.12). Fortunately, these high frequency errors are very efficiently smoothed out by the subsequent relaxation sweeps on G^k, and thus these errors are automatically corrected.

For elliptic operators which are 'far' from being singular and with a reasonable grid system $\{G^k\}$ condition (5.27) can be assured. For example, if L is the negative Laplacian, $-\Delta$, on a unit square with Dirichlet boundary conditions, then it is known (e.g. [9]) that the eigenvalues of L are given by

$$\mu_{m,n} = (m\pi)^2 + (n\pi)^2. \tag{5.29}$$

The corresponding eigenfunctions are:

$$\xi_{m,n} = \sin(m\pi x)\sin(n\pi y). \tag{5.30}$$

These eigenfunctions evaluated at the discrete interior grid points of a uniform mesh on the unit square, give the eigenfunctions of the discrete 5-point approximations, $L^k = -\Delta_h$. with h being the uniform mesh size. The eigenvalues of L^k are, with $\delta x = \delta y = h_k$:

$$\mu^k_{m,n} = 4[\sin^2(m\pi h_k/2) + \sin^2(n\pi h_k/2)] \,/\, h_k^2. \tag{5.31}$$

Some of these eigenvalues are tabulated in Table 5-1 for various mesh sizes, h_k. The ratios $\mu^k_{m,n}/\mu^{k-1}_{m,n}$ are given in Table 5-2. We see from Table 5-2 that condition (5.27) is satisfied, with $w_{k-1} \tilde{=} 1$, for all lower modes shown. These ratios are very close to unity, even for the case where the coarsest grid has only one interior point. We have seen from condition (5.11) that this closeness to unity is very desirable and this fact partly explains the well-documented success of MG methods for the Laplacian operator.

Near the limit point of the Bratu problem, the operator $L \equiv G_u = \Delta + \lambda e^u$ behaves very much like a shifted Laplacian operator. Clearly, if the factor e^u were replaced by a constant, α say, then G_u is replaced by the the Laplacian operator with a shift $\alpha\lambda$. Then the eigenvalue ratio $\mu^k_{1,1}/\mu^{k-1}_{1,1}$, valid for $\alpha\lambda = 0$, is replaced by:

$$(\mu^k_{1,1} - \alpha\lambda)/(\mu^{k-1}_{1,1} - \alpha\lambda). \tag{5.32}$$

Since $0 < u < 1.4$ the factor e^u does not vary much and we assume this approximation to be valid for some $\alpha > 0$. The situation is depicted

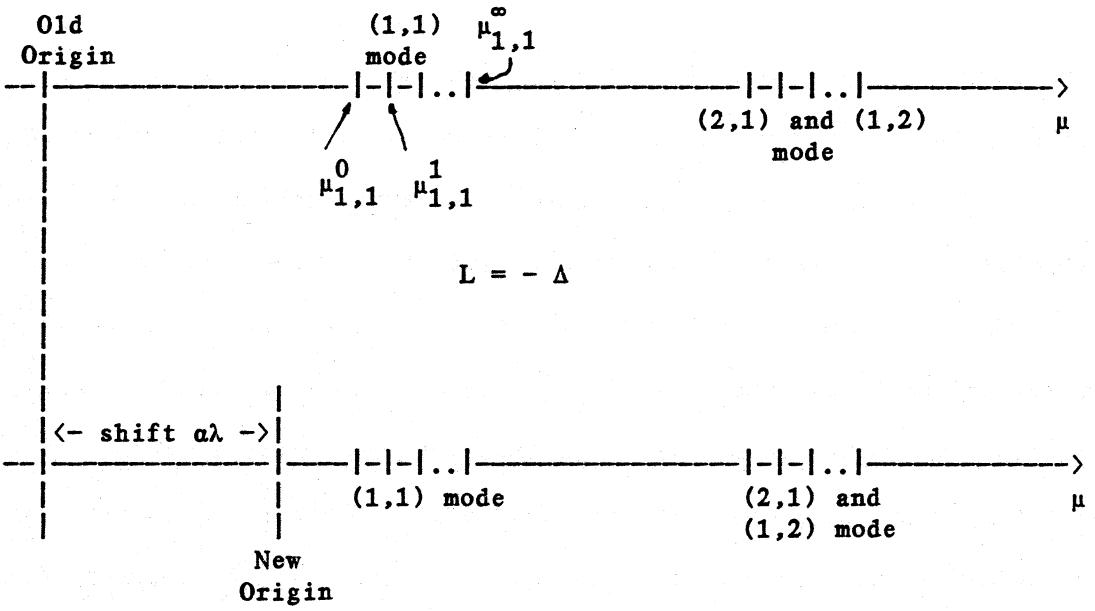
Table 5-1: $\mu^{k}_{m,n}$ for $-\Delta_{h_k}$

k =	0	1	2	3	∞
(m,n)	$h_0 = 1/2$	$h_1 = 1/4$	$h_2 = 1/8$	$h_3 = 1/16$	$h_\infty = 0$
1,1	16.0	18.745	19.487	19.676	19.739
2,1	NA	41.37258	47.238	48.812	49.348
1,2	NA	41.37258	47.238	48.812	49.348
2,2	NA	64.0	74.981	77.947	78.957
3,1	NA	NA	88.760	96.126	98.696
1,3	NA	NA	88.760	96.126	98.696
3,2	NA	NA	116.507	125.261	128.305
2,3	NA	NA	116.507	125.261	128.305
3,3	NA	NA	158.033	172.575	177.653

Table 5-2: Ratios $\mu^{k}_{m,n}/\mu^{k-1}_{m,n}$ for - Δ_{h_k}

(m,n)	h_k=1/4, h_{k-1}=1/2	h_k=1/8, h_{k-1}=1/4	h_k=1/16, h_{k-1}=1/8
1,1	1.17	1.04	1.01
2,1	NA	1.14	1.03
1,2	NA	1.14	1.03
2,2	NA	1.17	1.04
3,1	NA	NA	1.08
1,3	NA	NA	1.08
3,2	NA	NA	1.08
2,3	NA	NA	1.08
3,3	NA	NA	1.09

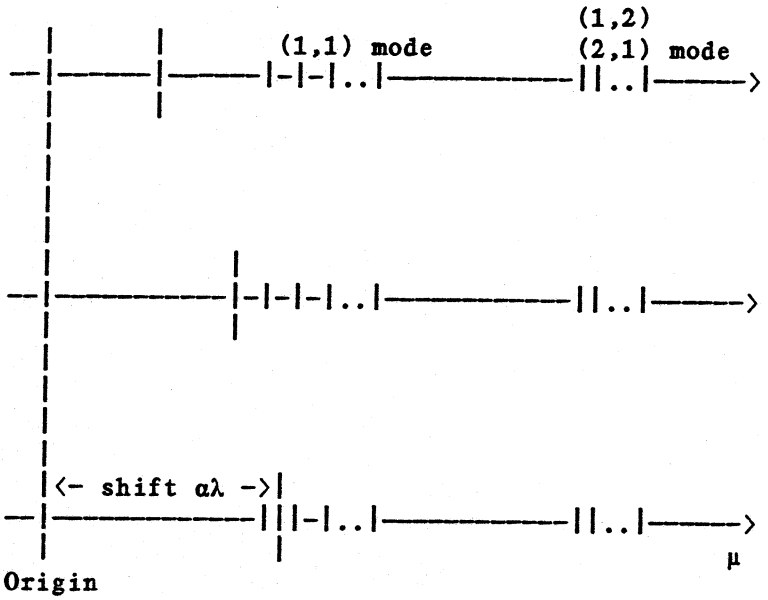
Figure 5-1: Spectrum of Shifted Laplacian



graphically in Figure 5-1 for the grid system that was used for Table 5-1. As the shift $\alpha\lambda$ approaches the group of eigenvalues corresponding to the (1,1) mode from below, the ratios in (5.31) increase. As $\alpha\lambda$ continues to increase the ratio of eigenvalues will become greater than 2, then increase towards $+\infty$, jump to $-\infty$ discontinuosly, and start increasing from $-\infty$ to 1. The situation is depicted in Figure 5-2.

We thus see, under the above assumptions, that condition (5.27) is first violated by the lowest mode (i.e. the (1,1) mode) on the two coarsest grids G^0 and G^1. In fact the lowest eigenvalues for the Bratu problem computed at the first point on the solution branch where Cycle C diverged, yields the ratio almost exactly 2! On the other hand, even at this point, condition (5.27) is satisfied by the (1,1) modes on the finer grids. In other words, the divergence of Cycle C is seen to be caused by <u>one</u> near-singular grid out of the whole hierachy of grids present. The mode that becomes singular at the limit point of the Bratu problem is the (1,1) mode, and this occurs first on the G^0 grid. As the limit point is approached, L^k on some of these grids may even become indefinite, while others (the finer grids) may still be positive definite. Essentially, the near-singular grid causes the (1,1) mode component of the correction v^{k-1}, when viewed as an approximation to v^k, to have the right <u>direction</u>, but the wrong <u>magnitude</u>. This phenomenon is not limitted to the Bratu problem. The only thing special about this problem is that it is the eigenvalue of the (1,1) mode that becomes zero at the limit point. For other problems, the eigenvalue of the operator L that becomes zero as the

Figure 5-2: Spectrum Near Singular Point



singular point is approached might correspond to other modes. Although the singular point in the Bratu problem is a limit point, we can expect the same behaviour at a bifurcation point.

Having now understood the cause of the divergence of the MG method, in the next section we shall discuss some modifications to the basic Cycle C algorithm that are designed to overcome such difficulties.

6. Remedies and New Algorithms

In this section we discuss approaches that have been devised to overcome the difficulties with the MG method near singular points. The first goal is to modify the basic Cycle C algorithm so that it will converge for values of λ close enough to the limit point so that the arc-length continuation procedure can take us past the limit point onto the upper solution branch. A more ambitious goal is to modify Cycle C further so that it will converge arbitrarily close to the singular point. Such an algorithm, when used in conjunction with the arc-length continuation technique for tracing solution branches, will make the overall algorithm much more robust. Moreover, such an algorithm may prove to be useful for locating singular points accurately, either using an arc-length continuation based procedure [13], or some other procedure that uses the operator G_u near the singular point [22]. We shall see that the first goal is relatively easy to achieve, whereas the second goal is much more difficult. However, we have devised a Cycle C based algorithm that has performed very well when applied very close to the limit point. The approaches that we have tried and that lead to the final algorithm will be discussed in this section. We shall describe them in the sequence that they were tried.

Before we proceed, however, we have to explain a few general strategies that were used. First of all, Gauss-Seidel and many other relaxation schemes are not very effective in smoothing the lower modes, especially modes with near zero eigenvalues. Hence, these modes must be eliminated by means other

than relaxation, even on the coarsest grid. Therefore, unless stated otherwise, we shall use a direct solution on the coarsest grid even though the operators L^k's may be positive definite. This does not affect the overall efficiency very much because the coarsest grid has so few points that direct solution is very fast and efficient.

Another strategy concerns the treatment of the mode that causes the divergence; that is the mode with a near zero eigenvalue, say ξ_1. In all the algorithms that are discussed, this mode is treated separately from the other modes. To do this, it is essential to have approximations to this mode and to its corresponding eigenvalues, say $\bar{\xi}_1^k$ and $\bar{\mu}_1^k$, respectively. Here we have to strike a balance between accuracy and efficiency. If we compute the ξ_1^k exactly, then we can completely eliminate the ξ_1^k error components on each grid. Thus, the problem on G^k can be reduced to one in which a_1^k is zero (see (5.25)). When this is done, we do not need to satisfy condition (5.27) for this mode. On the other hand, the work involved in computing accurate approximations to μ_1^k and ξ_1^k for each k would be at least as much as solving the original linear system. Our compromise has been to compute an approximation $\bar{\xi}_1^0$ to ξ_1 on the coarsest grid, G^0, by a few steps of inverse iteration with zero shift (since the eigenvalue we want is near zero). This is very inexpensive since G^0 is quite coarse and the LU factors of L^0 are already available. Then we interpolate $\bar{\xi}_1^0$ onto the finer grids. To eliminate the high frequency errors introduced in these interpolations, we do two things: 1) use higher order interpolation, e.g. cubic instead of linear, 2)

smooth the interpolated eigenfunctions by performing a few relaxation sweeps on $L^k \xi_1^k = 0$. Estimates of the eigenvalues, $\bar{\mu}_1^k$, are then computed using the Rayleigh Quotients: $\langle \bar{\xi}_1^k, L^k \bar{\xi}_1^k \rangle$. We view this as a preprocessing phase of the algorithm and the extra work is usually minimal compared to the overall work. Furthermore, since the eigenfunctions (not the eigenvalues) do not change very much in the neighborhood of the singular points, we can use the same approximation for different linearized operators L^k. The storage required to store these eigenfunctions is less than twice the size of the finest grid.

We use the (η,δ) adaptive version of the Cycle C algorithm, unless otherwise stated. The first modified algorithm is the following.

6.1 Under- and Over- Interpolation

The idea is to choose w_{k-1} in (3.5) for interpolation onto G^k, such that condition (5.27) is satisfied for ξ_1. Clearly the value:

$$w_{k-1} = \bar{\mu}_1^{k-1} / \bar{\mu}_1^k \ , \tag{6.1}$$

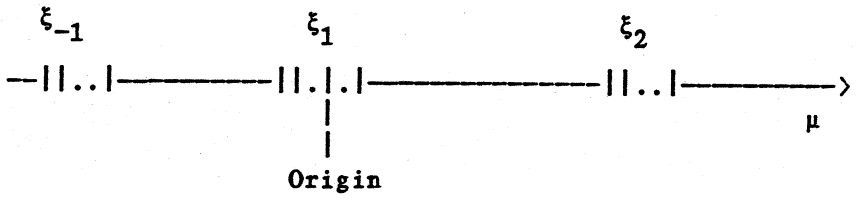
is in some sense optimal since it eliminates the ξ_1 term in (5.25). For the case discussed in Section 4.3, this modification allows the computation to continue past the point $\lambda = 6.804$, where divergence of Cycle C first occurred. In fact (with a little luck) we succeeded in continuing around the limit point onto the upper branch. Here the eigenfunction ξ_1 no longer presented difficulties for the MG algorithm. For some of these cases μ_1^0 is actually negative and therefore (6.1) yields a a negative value for w_1. In this case the transfer from G^0 to G^1 violates condition (5.27) for all modes <u>other</u> than ξ_1. The errors in these modes must be reduced by extra relaxation sweeps on

G^1. In other words G^0 only provides a proper correction on G^1 for the ξ_1 mode, all higher modes are treated incorrectly during the transfer. The efficiency of the algorithm thus suffers. This effect is especially pronounced if some factors w_k are either very large or negative or (worse) both. The algorithm is very sensitive to the parameters (η,δ) and thus is not robust. It can even diverge if the higher modes are not reduced fast enough on G^k after the transfer from G^{k-1}.

Even worse, the above algorithm will not work for indefinite problems in which some intermediate eigenvalue is near zero. For example, if the spectra of the L^k are similar to those in Figure 6-1, the interpolation factors w_k are controlled by the ξ_1^k belonging to eigenvalues μ_1^k near zero. On the other hand, the eigenfunctions ξ_{-1}^k require that condition (5.27) be satisfied because these modes cannot be liquidated by relaxation. Conflicts can occur when ξ_1^k requires w_k to be negative while ξ_{-1}^k requires w_k to be positive. Indefinite problems of this type occur frequently in nonlinear eigenvalue problems. Mere under- or over-interpolation must run into difficulties for such problems, near the singular points.

The above considerations make it clear that the eigenfunction with the near-zero eigenvalue must be isolated and treated different from the other eigenfunctions. We use the approximate eigenfunctions that are computed in the preprocessing phase for this purpose in the following procedure.

Figure 6-1: Intermediate Eigenvalue near Zero



6.2 Under- and Over- Interpolate the Singular Eigenfunction Only

We use an interpolation different from that in (3.5). Specifically if

$$v^{k-1} = \sum_{i=1}^{N_{k-1}} a_i^{k-1} \xi_i^{k-1} \tag{6.2}$$

on G^{k-1}, we interpolate it onto G^k by

$$v^k = w_{k-1} a_1^{k-1} I_{k-1}^k \xi_1^{k-1} + I_{k-1}^k \sum_{i=2}^{N_{k-1}} a_i^{k-1} \xi_i^{k-1} \ . \tag{6.3}$$

Further w_{k-1} is chosen to satisfy (6.1). Since we only have an approximation to ξ_1^k, we use, instead of (6.3):

$$\begin{aligned} v^k = I_{k-1}^k \ & [v^{k-1} - \langle v^{k-1}, \bar{\xi}_1^{k-1} \rangle \bar{\xi}_1^{k-1}] \\ & + w_{k-1} \langle v^{k-1}, \bar{\xi}_1^{k-1} \rangle \ I_{k-1}^k \bar{\xi}_1^{k-1} . \end{aligned} \tag{6.4}$$

In practice, this performed much better than indiscriminate under- and over-interpolation described in section 6.1. It was the more efficient when both procedures worked. In many cases when (6.1) yields large and/or negative values for w_k, only the current scheme converges. In principle, it will also work for indefinite problems like that depicted in Figure 6-1. The efficiency in most cases was very respectable; in the range of 6-10 units per order of magnitude reduction in the residual. It is also quite insensitive to the parameters (η,δ). Thus, it can be used very efficiently and reliably with the arc-length continuation procedure for tracing out solution branches.

Unfortunately, this improved algorithm fails when the magnitude of w_k becomes too large. This occurs when L^k is very nearly singular, that is with μ_1^k very close to zero. Since we only have an approximation $\bar{\xi}_1^k$ to ξ_1^k, large factors w_k in (6.4) introduce very large errors in the other modes. Moreover,

the estimates $\bar{\mu}_1^k$ using Rayleigh-Quotients tend to be too large (relatively) when μ_1^k is very small. Then (6.1) gives a value of w_k that is too small. Both of the above result in lower efficiency and reliability. In extreme cases, this makes the algorithm impractical. To overcome this difficulty, we devise an algorithm that will work even if one of the operators L^k is very nearly singular. For this we employ the idea of skipping a grid.

6.3 Skipping the Singular Grid

The previous algorithm fails if the operator is very nearly singular on one of the grids, say G^k. The idea here is to simply delete this grid from the hierachy of grids used by the MG algorithm. If the remaining grids are not as singular as the deleted grid it would seem that the algorithm described in 6.2 should work. However, calculations show that skipping a grid can cause other problems. When G^k is skipped, the mesh changes more drastically from G^{k-1} to G^{k+1}, and hence the interpolation in (6.4) (now I_{k-1}^{k+1} instead of I_{k-1}^k) introduces larger errors into the higher modes on G^{k+1}. These high frequency errors can cause divergence of the MG process unless controlled properly by the parameters (η,δ). A large value of η, say between .8 and .9, makes the algorithm more robust but involves more work than for a smaller value of η, say .5. We encountered a case where, with all else the same, the new skipping algorithm converges for $\eta = .9$ but diverges for $\eta = .6$. Granted with $\eta = .9$ the algorithm may be very reliable, such sensitivity to one parameter is very undesirable. Therefore, we considered the following modification.

6.4 Skipping the Singular Grid for the Singular Eigenfunction Only

The idea is to skip the singular grid G^k for ξ_1 only, and to keep it for smoothing the other modes. In the actual implementation, we modify the algorithm described in section 6.2 to use

$$w_{k-1} = \bar{\mu}_1^{k-1} \; / \; \bar{\mu}_1^{k+1} \tag{6.5}$$

for ξ_1 and $w_{k-1} = 1$ for all other modes to transfer from G^{k-1} to G^k and, after a few smoothing sweeps on G^k, transfer to G^{k+1} with $w_k = 1$ for all modes. Note that we do not try to solve the G^k equations for v^k. Trying to do that would result in large magnification of the ξ_1^k component in v^k, since μ_1^k is near zero. This would in turn cause problems during the transfer to G^{k+1}.

In addition, we have experimented with using a mixture of the adaptive (η, δ) strategy with the non-adaptive (p,q) strategy (cf. section 3.2). We have found a (η,q) strategy that is as good as any other we have tried. In this strategy, we use η to control when we terminate relaxation on a certain grid and go on to a coarser grid, and use q to control how many sweeps to do on a grid after transfer from a coarser grid before interpolating onto a finer grid. A typical set of parameters that worked well is ($\eta = .6$, $q = 2$). The resulting algorithm is fairly insensitive to actual values of η and q and is quite robust. It is also quite efficient. It consistently achieved an efficiency of less than about 12 units per order of magnitude reduction in the residual for most problems that we have encountered. Some of these problems have very singular grids which presented difficulties for all of the previous algorithms.

7. Summary

In this paper, we study arc-length continuation techniques and multi-grid techniques for solving nonlinear elliptic eigenvalue problems. We have applied these techniques to solve a model nonlinear elliptic eigenvalue problem (the Bratu problem). We have found that as long as we stay away from singular points, the two techniques combined to give a very powerful and efficient procedure for tracing solution branches. Near singular points, however, the standard multi-grid method has difficulty converging on the linearized elliptic systems that arise in the continuation procedure. One consequence is that we cannot continue past the limit point in the model problem. This divergence is successfully analysed and several modified multi-grid algorithms have been designed based on this analysis. The best of these modified algorithms performs efficiently and reliably arbitrarily close to the singular points. This enables the continuation procedure to continue past the limit point with no difficulty. It seems reasonable that this modified multi-grid algorithm can be useful in more general situations where nearly singular elliptic systems arise, such as in inverse iteration [11, 17].

REFERENCES

[1]
J.P. Abbott, An Efficient Algorithm for the Determination of Certain Bifurcation Points, Journal of Computational and Applied Mathematics, 4 (1978), pp. 19 - 27.

[2]
H. Amann, Fixed Point Equations and Nonlinear Eigenvalue Problems in Ordered Banach Spaces, SIAM Review, 18 (1976), pp. 620-709.

[3]
N.S. Bakhvalov, Convergence of a Relaxation Method with Natural Constraints on an Elliptic Operator, Z. Vycisl. Mat. i Mat. Fiz., 6 (1966), pp. 861-885.

[4]
A. Brandt, Multi-level Adaptive Solution to Boundary Value Problems, Math. Comp., 31 (1977), pp. 333-390.

[5]
T.F. Chan, Numerical Computation of Large Amplitude Internal Solitary Waves, paper presented at SIAM conference at Denver, Nov., 1979.

[6]
B. Chen and P. Saffman, Numerical Evidence for the Existence of New Types of Gravity Waves of Permanent Form on Deep Water, Studies in Applied Math., 62 (1980), pp. 1-21.

[7]
D.W. Decker and H.B. Keller, Path Following Near Bifurcation, Comm. Pure and Appl. Math., 34 (1981), pp. .

[8]
R.P. Federenko, A Relaxation Method for Solving Elliptic Difference Equations, Z. Vycisl. Mat. i Mat. Fiz., 1 (1961), pp. 922-927.

[9]
G.E. Forsythe and W.R. Wasow, Finite Difference Methods for Partial Differential Equations, John Wiley and Sons, New York, 1960.

[10]
W. Hackbush, On the Multi-Grid Method Applied to Difference Equations, Computing, 20 (1978), pp. 291-306.

[11]
W. Hackbush, On the Computation of Approximate Eigenvalues and

Eigenfunctions of Elliptic Operators By Means of a Multi-Grid Method, SIAM J. Numer. Anal., 16 (1979), pp. 201-215.

[12]
H.B.Keller, Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems, Applications of Bifurcation Theory, P. Rabinowitz, ed., Academic Press, New York, 1977, pp. 359-384.

[13]
H.B. Keller, Global Homotopies and Newton Methods, Recent Advances in Numerical Analysis, Carl de Boor and Gene Golub, ed., Academic Press, New York, 1978, pp. 73-94.

[14]
H.B. Keller and D.S. Cohen, Some Positone Problems Suggested By Nonlinear Heat Generation, J. Math. and Mech., 16 (1967), pp. 1361-1376.

[15]
H.B. Keller and R. Schreiber, Accurate Solutions for the Driven Cavity, In preparation.

[16]
M. Lentini and H.B. Keller, The von Karman Swirling Flows, Siam J. Appl. Math., 38 (1980), pp. 52-64.

[17]
S.F. McCormick, A Mesh Refinement Method for $Ax = \lambda Bx$, Manuscript.

[18]
H.D. Mittelmann and H. Weber, Numerical Methods for Bifurcation Problems - A Survey and Classification, Bifurcation Problems and their Numerical Solution, Workshop on Bifurcation Problems and their Numerical Solution, January 15-17, Dortmund, 1980, pp. 1-45.

[19]
R.A. Nicolaides, On Multiple Grid and Related Techniques for Solving Discrete Elliptic Systems, J. Comp. Phys., 19 (1975), pp. 418-431.

[20]
S.V. Parter, Mildly Nonlinear Elliptic Partial Differential Equations and Their Numerical Solution. I , Numer. Math., 7 (1965), pp. 113-128.

[21]
S.V. Parter, Maximal Solutions of Mildly Nonlinear Elliptic Equations, Numerical Solution of Nonlinear Differential Equations, D. Greenspan,

ed., John Wiley, New York, 1966, pp. 213-238.

[22]

W.C. Rheinboldt, Numerical Methods for a Class of Finite Dimensional Bifurcation Problems, SIAM J. of Numer. Anal., 15 (1978), pp. 1-11.

[23]

W.C. Rheinboldt, Solution Fields of Nonlinear Equations and Continuation Methods, SIAM J. Numer. Anal., 17 (1980), pp. 221-237.

[24]

J.B. Rosen, Approximate Solution and Error Bounds for Quasilinear Elliptic Boundary Value Problems, SIAM J. Numer. Anal., 8 (1970), pp. 80-103.

[25]

S. Rosenblat and R. Szeto, Multiple Solutions of Nonlinear Boundary Value Problems, Studies in Applied Math., 63 (1980), pp. 99-117.

[26]

R.B. Simpson, A Method for the Numerical Determination of Bifurcation States of Nonlinear Systems of Equations, SIAM J. Numer. Anal., 12 (1975), pp. 439-451.

[27]

R.K.H. Szeto, The Flow Between Rotating Coaxial Disks, Ph.D. Thesis, California Institute of Technology, Pasadena, CA , 1978.