

## Solving Boundary-Value Problems with a Spline-Collocation Code\*

URI ASCHER

*Department of Computer Science, The University of British Columbia,  
Vancouver, British Columbia, V6T 1W5 Canada*

Received December 11, 1978; revised March 5, 1979

Whereas initial-value ODEs have been routinely solved to date using robust general-purpose packages, for boundary-value problems many researchers currently write their own special-purpose codes, often based on special-purpose methods. This painful, time-consuming, and error-prone process can and should be avoided in all but the toughest and most peculiar cases. In this paper we use a general-purpose code, developed with our colleagues, to solve reliably, accurately, and efficiently five problems in science and engineering which display a variety of numerical difficulties. The code is based on spline-collocation at Gaussian points and is capable of handling nonlinear mixed-order systems of multipoint boundary-value ODEs. It produces reliable error estimates and has been found particularly useful for difficult problems. Here we demonstrate its power and applicability.

### 1. INTRODUCTION

Many researchers currently seem to feel the need to write their own special-purpose codes, often based on special-purpose numerical methods, to solve particular boundary-value differential equations (ODEs) encountered in their work. This is in contrast to the situation with initial-value problems in ODEs, where available general-purpose software is in common use. A good special-purpose implementation is usually a painful, time-consuming, and error-prone process, and if possible should be avoided. The advance of robust boundary-value software has made it possible to solve all but the toughest and most peculiar problems using a general tool.

Such a general purpose code, COLSYS [1, 2], is recommended and used here. The package is based on spline collocation at Gaussian points and is capable of handling nonlinear mixed-order systems of multipoint boundary-value ODEs. Reliable error estimates are produced and adaptive mesh refinement is performed. The piecewise polynomial approximate solution is given in terms of a  $B$ -spline basis [3, 4], and the resulting sparse linear systems are solved efficiently [5]. For nonlinear problems, the damped Newton method is used for the first mesh, and modified Newton iterations with a fixed Jacobian are performed for subsequent refined meshes.

In preliminary comparisons, COLSYS has been found to be competitive with other

\* This work was supported in part by NRC Grant No. A4306.

robust software for boundary-value ODEs [1, 2]. It is particularly useful for difficult problems, including those containing thin layers and singular coefficients. Singular Jacobians, however, cannot be currently handled by the code. Also, a very large order of the differential system may cause excessive storage requirements.

In this paper we use the code to solve a variety of problems in science and engineering, demonstrating its power and applicability. Following a short description of the numerical techniques used we present and discuss the solution of problems in semiconductor devices, vertical channel flow, shell buckling, optimal control of exhaustible resources, and elastic shell deformation. These problems give rise to a variety of numerical difficulties which are resolved here; e.g., steep interior and boundary layers in the solution, singularities, termination of a solution branch, and infinite intervals with slowly decaying solutions.

We conclude that a robust, general-purpose code like COLSYS can be used successfully for solving accurately, reliably, and efficiently the difficult boundary-value problems encountered in practice.

## 2. PROBLEM DEFINITION AND NUMERICAL TECHNIQUES

Consider a system of  $d$  nonlinear differential equations of orders  $1 \leq m_1 \leq \dots \leq m_d$ ,

$$u_n^{(m_n)}(x) = F_n(x; \mathbf{z}(\mathbf{u})) \quad a < x < b, \quad n = 1, \dots, d, \quad (1)$$

where the sought solution  $\mathbf{u} = (u_1, \dots, u_d)$  is an isolated solution vector and  $\mathbf{z}(\mathbf{u}) = (u_1, u_1', \dots, u_1^{(m_1-1)}, u_2, \dots, u_d, \dots, u_d^{(m_d-1)})$  is the vector of unknowns that would result from converting (1) to a first-order system. The system is subject to  $m^* = \sum_{n=1}^d m_n$  nonlinear multipoint separated boundary conditions

$$g_j(\zeta_j; \mathbf{z}(\mathbf{u})) = 0 \quad j = 1, \dots, m^*, \quad (2)$$

where  $\zeta_j$  is the location of the  $j$ th boundary (or side) condition,  $a \leq \zeta_1 \leq \zeta_2 \leq \dots \leq \zeta_{m^*} \leq b$ .

The method of spline collocation at Gaussian points, or orthogonal collocation [6, 7], has been implemented in COLSYS to solve (1), (2). The problem is solved on a sequence of discrete meshes, until user-specified error tolerances are satisfied. For a specific mesh  $\pi : a = x_1 < x_2 < \dots < x_{N+1} = b$ , with  $h_i = x_{i+1} - x_i$ ,  $h = \max_{1 \leq i \leq N} h_i$ , and an integer  $k > m_d$ , the collocation solution  $\mathbf{v}(x) = (v_1, \dots, v_d)$  is a piecewise polynomial vector function: each  $v_n \in C^{m_n-1}[a, b]$  is a polynomial of degree less than  $k + m_n$  on each subinterval (or element)  $(x_i, x_{i+1})$ ,  $n = 1, \dots, d$ ,  $i = 1, \dots, N$ . This solution is determined by requiring that it satisfy the differential equations (1) at the images of the  $k$  zeros of the appropriate Legendre polynomial in each element.

We now briefly describe the techniques for controlling the error and the nonlinear iteration in COLSYS. More complete descriptions can be found in [1, 2].

Provided that the problem (1), (2) is smooth enough, the error in  $\mathbf{v}$  for  $x \in [x_i, x_{i+1})$  is given by

$$u_n^{(l)}(x) - v_n^{(l)}(x) = c_{n,l} u^{(k+m_n)}(x_i) h_i^{k+m_n-l} + O(h^{k+m_n-l+1}),$$

$$l = 0, 1, \dots, m_n - 1, \quad n = 1, \dots, d, \quad (3)$$

where  $c_{n,l}$  are known bounded functions of  $x$ . This expression is used both for estimating the error accurately via mesh halving to check against user-given tolerances and for mesh refinement. By approximating  $u_n^{(k+m_n)}(x_i)$  using  $v_n(x)$ , a rough estimate for the error in each subinterval is obtained. If considered worthwhile, a redistribution of the mesh points is performed to equidistribute the error (i.e., have approximately the same magnitude of error in each element) and  $v(x)$  is recomputed. If not, each element is halved, a new solution  $v^*(x)$  is computed, and the error in  $v^*(x)$  is estimated using  $v(x)$ ,  $v^*(x)$  and (3). The decision whether to redistribute the mesh points or not, as well as an estimate of the desired mesh size, are made automatically by COLSYS.

For nonlinear problems, two modifications of Newton's method are used. Using quasi-linearization, at each iteration a linearized problem is solved by collocation as described above. At the first mesh, for which frequently no adequate initial approximation is available, the damped Newton method is used with the damping or relaxation factor being controlled by a modified scheme originally due to Deuffhard [8]. For further meshes (provided previous convergence has been obtained) the converged solution on the previous mesh serves as an excellent initial approximation, and fast Newton iterations with the Jacobian held fixed are performed as long as the residual monotonically decreases at a sufficiently rapid rate.

To use COLSYS, the user must specify a set of tolerances  $tol_j$  and pointers  $lto_j$ ,  $j = 1, \dots, ntol$ . The (successful) stopping criterion is that

$$\|z_i(\mathbf{u}) - z_i(\mathbf{v})\|_{(i)} \leq tol_j + \|z_i(v)\|_{(i)} \cdot tol_j, \quad l = lto_j, \quad j = 1, \dots, ntol, \quad i = 1, \dots, N, \quad (4)$$

where, for any appropriate function  $\psi$ ,

$$\|\psi\|_{(i)} := \max_{x \in [x_i, x_{i+1})} |\psi(x)| \quad (5)$$

is being evaluated approximately. The reliability of the error-estimating procedure being used for the stopping criterion has been verified elsewhere [1, 2].

### 3. PROBLEMS AND SOLUTIONS

The following notation is used in the presentation here.

$tol(u_i^{(j)})$  = Combined error tolerance for the component  $u_i^{(j)}(x)$  of the solution. (COLSYS allows the user to specify different tolerances for different components, and the mesh selection algorithm considers only these components for which tolerances are specified.)

time = Actual solution time in seconds.

$a \pm b = a \cdot 10^{\pm b}$ .

$k$  = Number of collocation points per subinterval.

$N(I)$  = Successive mesh sizes, i.e., numbers  $N$  of elements (subintervals) required, optionally followed in parenthesis by the number of full Newton iterations performed on each mesh for nonlinear problems.

The initial meshes are uniform unless otherwise stated. In the case of continuation, i.e., using a formerly obtained solution for the initial approximation of another problem, the new mesh is twice as coarse as the mesh on which this initial approximation is defined.

All the computations reported here were carried out on the Amdahl V/6-II computer at the University of British Columbia, using the IBM Fortran H compiler with double precision (14 hexadecimal digits).

**PROBLEM 1.** The following singularly perturbed system with singularities in the boundary conditions describes the contact of two semiconductors with different types of conductance [9]:

$$\epsilon y' = x_1 - x_2 + K, \quad (6)$$

$$x_1' = x_1 y - c, \quad (7)$$

$$x_2' = -x_2 y + c, \quad (8)$$

$$x_1(0) = x_2(0), \quad x_1(1) = 0, \quad x_2(1) = K. \quad (9)$$

$y$ -Electric field strength;  $x_1$ -hole density;  $x_2$ -electron density;  $K, c$  are given constants (we take  $K = c = 1.0$ ). The parameter  $\epsilon$  is a small positive constant:  $\epsilon \sim (l/L)^2$ , where  $L$  is a characteristic dimension of the system and  $l$  is a characteristic dimension of the space charge region. See [9] and references therein for asymptotic analysis, physical interpretation, and the approximate solution computed for  $\epsilon = 10^{-3}$ . The solution components have a boundary layer at 0 with  $y(0) \sim -\epsilon^{-1/2}$ .

We use this as a model to examine the performance of the code on a mildly difficult boundary layer problem. We have computed the solution for  $\epsilon = 10^{-p}$ ,  $p = 3, 4, \dots, 10$ , with  $k = 5$  and  $tol_j = 10^{-6}$ ,  $j = 1, 2, 3$ . For  $p = 3, 4$ , and 5 the initial solution was  $y \equiv x_1 \equiv 0$ ,  $x_2 \equiv K$  with a uniform initial mesh of 10 subintervals. For  $p \geq 6$  simple continuation was used with the initial solution for  $p$  being the final one for  $p - 1$ . Values of the solution at 0 are tabulated below together with the mesh sequences. In the column entitled  $N$ -layer we list the number of elements inside the interval  $(0, 10\epsilon^{1/2})$  for the last mesh encountered for each  $\epsilon$ .

**PROBLEM 2.** Consider the problem of fluid injection through one side of a long vertical channel [10]. The Navier-Stokes and the heat transfer equations can be reduced to the following system

TABLE I  
Problem 1

$\epsilon$	$y(0)$	$x_1(0)$	Time	$N$	$N$ -Layer
$10^{-3}$	$-0.21308 + 2$	0.98833	1.6	10, 9, 18, 36	24
$10^{-4}$	$-0.69191 + 2$	0.99644	2.7	10, 10, 10, 20, 11, 22, 44	28
$10^{-5}$	$-0.22060 + 3$	0.99889	3.6	10, 20, 20, 16, 32, 16, 32	20
$10^{-6}$	$-0.69938 + 3$	0.99965	5.1	16, 16, 16, 32, 22, 44, 22, 44, 22, 44	26
$10^{-7}$	$-0.22134 + 4$	0.99989	6.4	22, 22, 22, 44, 27, 54, 27, 54, 27, 54	32
$10^{-8}$	$-0.70012 + 4$	0.99996	3.7	27, 26, 52, 26, 52	20
$10^{-9}$	$-0.22142 + 5$	0.99999	4.6	26, 26, 26, 52, 36, 72	20
$10^{-10}$	$-0.70020 + 5$	1.00000	5.2	36, 35, 70, 42, 84	22

$$f''' - R[(f')^2 - ff''] + R \cdot A = 0, \tag{10a}$$

$$h'' + R \cdot fh' + 1 = 0, \tag{10b}$$

$$\theta'' + P \cdot f\theta' = 0, \tag{10c}$$

$$f(0) = f'(0) = 0, \quad f(1) = 1, \quad f'(1) = 0, \tag{11a}$$

$$h(0) = h(1) = 0, \tag{11b}$$

$$\theta(0) = 0, \quad \theta(1) = 1. \tag{11c}$$

Here  $f$  and  $h$  are two potential functions,  $\theta$  is a temperature distribution function, and  $A$  is an undetermined constant.  $V$ -Constant injection velocity,  $U$ -thickness of the channel,  $\nu$ -viscosity of the fluid,  $R = UV/\nu =$  crossflow Reynolds number,  $P$ -Peclet number (we take  $P = 0.7R$ ). See [10] for details.

In a typical manner, this problem is most efficiently solved by breaking it into three separate problems:

I. Differentiating (10a) we obtain the fourth-order equation

$$f'''' = R[f'f'' - ff'''] \tag{12}$$

subject to (11a).

II. The second-order *linear* problem (10b), (11b).

III. The second-order *linear* problem (10c), (11c).

The problems get numerically harder as Reynolds number  $R$  increases. In [10] solutions for values of up to  $R = 25$  were calculated. In Table II we summarize some computations for various values of  $R$ , performed without continuation. For larger  $R$ , continuation should be used. We used  $k = 5$  for problem I,  $k = 4$  for problems II and III, and  $tol = 10^{-6}$  on all components of  $\mathbf{z}$  of (1). The initial solutions for each  $R$  were  $f \equiv 0$  and the initial meshes were all uniform of five elements. In Figs. 1 and 2,

TABLE II  
Problem 2

Problem	$R$	Time	$N(I)$
I	25	0.4	5(4), 10(1)
II	25	0.2	5, 10, 20
III	25	0.2	5, 10, 20
I	100	0.8	5(5), 10(1), 7(1), 14(1)
II	100	0.3	5, 10, 8, 16
III	100	0.4	5, 10, 20, 11, 22
I	500	1.6	5(6), 10(1), 10(1), 8(1), 16(1), 32(1)
II	500	0.5	5, 10, 8, 16, 32
III	500	0.8	5, 10, 10, 20, 11, 22, 15, 30
I	2000	2.7	5(7), 10(2), 20(1), 17(1), 34(1), 17(1), 34(1)
II	2000	0.5	5, 10, 20, 10, 20
III	2000	0.6	5, 10, 10, 10, 20, 13, 26
I	10000 <sup>a</sup>	3.9	5(9), 10(2), 20(1), 40(1), 20(1), 40(1), 20(1), 40(1)
II	10000	0.8	5, 10, 20, 20, 10, 20, 10, 20
III	10000	1.1	5, 10, 20, 20, 20, 40, 20, 40

<sup>a</sup> For this value of  $R$ ,  $tol(f'')$  was relaxed to avoid roundoff error difficulties.

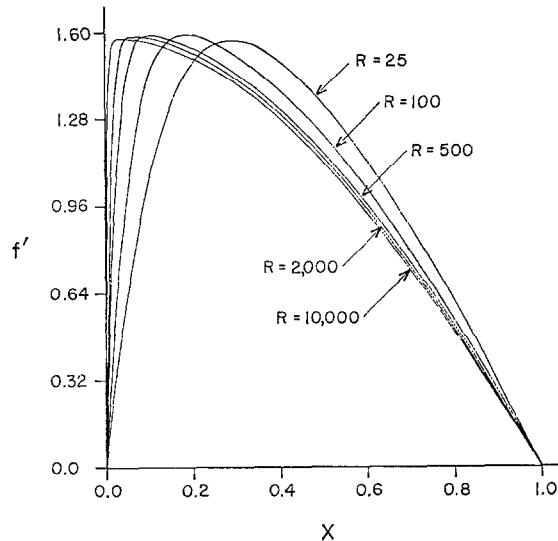


FIG. 1. Problem 2, tangential velocity.

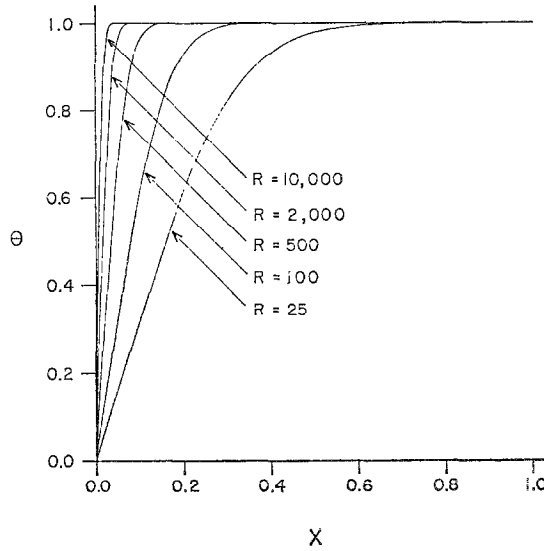


FIG. 2. Problem 2, temperature distribution.

the curves for the tangential velocity  $f'$  and the temperature distribution  $\theta$  are plotted for various values of  $R$ .

**PROBLEM 3.** The following system of equations arises in the study of the elastic stability of thin shallow spherical shells subject to uniform pressure:

$$f'' = -\mu^2 g + fg - (3/x)f' - 2\tau, \quad 0 < x < 1, \quad (13)$$

$$g'' = \mu^2 f - \frac{1}{2}f^2 - (3/x)g',$$

$$f'(0) = g'(0) = 0, \quad f(1) = 0, \quad g'(1) + \frac{2}{3}g(1) = 0. \quad (14)$$

Here  $x$  is the normalized polar angle,  $f$ -normalized angular deflection,  $g$ -normalized stress,  $\mu^2 = 144 =$  parameter characterizing the geometry of the shell, and  $\tau$ -load parameter (see [8] and references therein).

As is typical for problems arising from a dimensional reduction from partial differential equations due to symmetry, this problem has singular coefficients, but smooth solution at the origin. Unlike shooting codes, COLSYS handles this situation without any modifications. A similar situation arises in problem 5.

It is known for this problem that there exists a critical value  $\tau_c$  (buckling load) such that for  $\tau > \tau_c$ , system (13), (14) has no solution. In [8] the following result was reached using a sophisticated multiple shooting code with an adaptive continuation strategy:

$$19964.6 \leq \tau_c < 19971. \quad (15)$$

Using COLSYS we want to bracket  $\tau_c$  more accurately. We compute a chain of simple

continuations, i.e., a sequence of solutions to problems with increasing  $\tau$ , using the solution of the  $(j - 1)$ st problem as the initial approximation for the  $j$ th one.

With  $k = 4$ ,  $tol(f) = tol(g) = 10^{-7}$ , the following continuation chain in  $\tau$  was obtained (for  $\tau = 0$  the solution is trivial,  $f \equiv g \equiv 0$ ):

$$\tau: 0 \rightarrow 19969.6 \rightarrow 19969.62 \rightarrow 19969.63.$$

Each continuation step consumed about 2 sec. The last  $\tau$  was obtained at a relatively great effort, the last mesh size being  $N = 64$ . Attempts to increase  $\tau$  by 0.001 failed. Additional experiments, with additional tolerances on  $f'$  and  $g'$ , lead us to conclude that

$$\tau_c = 19969.63 \pm 0.001. \tag{16}$$

**PROBLEM 4.** The following problem arises in the theory of the optimal use of an exhaustible two-grade resource:

$$k = k^a r^b - \theta r - c, \quad 0 \leq t < \infty, \tag{17}$$

$$(1 - b) k^a r^b = c, \tag{18}$$

$$\theta(t) = \theta_1, \quad 0 \leq t \leq T, \tag{19}$$

$$= \theta_2, \quad T < t < \infty,$$

$$k(0) = k_0, \quad \int_0^T r(t) dt = \bar{D}_1, \quad \int_T^\infty r(t) dt = \bar{D}_2. \tag{20}$$

Here  $t$  is time;  $r$ -resource flow per head;  $k$ -capital stock per head;  $c$ -maximum sustainable constant consumption level;  $\theta$ -unit extraction cost;  $T$ -switching instance from low-cost to high-cost resource; and  $a, b, k_0, \bar{D}_1, \bar{D}_2, \theta_1$ , and  $\theta_2$  are known positive constants,  $0 < a + b < 1, b < a, \theta_1 < \theta_2$ . See [11, 12] for full details and presentation of the economic model.

We demonstrate here how one can deal with an unknown switching point  $T$  (at which the coefficient  $\theta$  has a jump discontinuity) and with an infinite interval when a slowly decaying function  $r$  is present. In order to bring conditions (20) into the form of (2), define

$$D(t) = \int_0^t r(\tau) d\tau \tag{21}$$

obtaining  $D(0) = 0, D(T) = \bar{D}_1, D(\infty) = \bar{D}_1 + \bar{D}_2$ . Then we get

$$k = \frac{cb}{1 - b} - \theta D \tag{22}$$

and this can be integrated, using the known values of  $k(0)$  and  $D(0)$  and continuity of  $k$  and  $D$  at  $t = T$ . Also, to handle the unknown switching point  $T$ , define  $x = t/T$  as the new independent variable for  $0 \leq t \leq T$  and  $D_1 \equiv D$  there. Moreover, for  $T \leq t$



$< \infty$  define  $D_2 \equiv D$  and  $x = (T/t)^p$ , where  $p$  is a constant chosen so that  $p \leq (a-b)/b$  (we take  $p = \frac{1}{3}$ ). The resulting problem to be solved is

$$\begin{aligned} T' &= 0, \\ c' &= 0, \end{aligned} \quad 0 < x < 1, \quad (23)$$

$$\begin{aligned} D_1' &= T \left( \frac{c}{1-b} \right)^{1/b} \left( \frac{cb}{1-b} Tx - \theta_1 D_1 + k_0 \right)^{-a/b}, \\ D_2' &= -\frac{T}{p} x^{-(1/p)-1} \left( \frac{c}{1-b} \right)^{1/b} \left( \frac{cbT}{1-b} x^{-1/p} - \theta_2 D_2 + k_0 + (\theta_2 - \theta_1) \bar{D}_1 \right)^{-a/b} \\ D_1(0) &= 0, \quad D_1(1) = \bar{D}_1, \quad D_2(0) = \bar{D}_1 + \bar{D}_2, \quad D_2(1) = \bar{D}_1. \end{aligned} \quad (24)$$

Note that, whereas in many cases integration on a finite interval  $[T, L]$  with  $L$

TABLE III  
Problem 4

	Case	$c$	$T$	Time	$N(t)^a$
I	$b = 0.05, k_0 = 2.4$ $\bar{D}_1 = 10, \bar{D}_2 = 50$	1.1406	11.03	1.8	5(6), 10, 20, 10, 20
II	$b = 0.1, k_0 = 2.4$ $\bar{D}_1 = 10, \bar{D}_2 = 50$	1.1386	6.676	1.0	5(4), 10, 20
III	$b = 0.15, k_0 = 2.4$ $\bar{D}_1 = 10, \bar{D}_2 = 50$	1.0674	9.840	0.6	5(4), 10
IV	$b = 0.05, k_0 = 2.4$ $\bar{D}_1 = 10, \bar{D}_2 = 25$	1.1392	11.49	1.0	5(4), 10, 20
V	$b = 0.1, k_0 = 2.4$ $\bar{D}_1 = 10, \bar{D}_2 = 25$	1.1080	9.815	0.6	5(4), 10
VI	$b = 0.15, k_0 = 2.4$ $\bar{D}_1 = 10, \bar{D}_2 = 25$	0.98053	24.01	0.6	5(5), 10.
VII	$b = 0.05, k_0 = 2.4$ $\bar{D}_1 = 25, \bar{D}_2 = 50$	1.1590	21.82	1.8	5(6), 10, 20, 10, 20
VIII	$b = 0.1, k_0 = 2.4$ $\bar{D}_1 = 25, \bar{D}_2 = 50$	1.1740	14.30	0.5	5(3), 10
IX	$b = 0.15, k_0 = 2.4$ $\bar{D}_1 = 25, \bar{D}_2 = 50$	1.1128	27.92	0.6	5(5), 10
X	$b = 0.05, k_0 = 4.8$ $\bar{D}_1 = 10, \bar{D}_2 = 50$	1.3036	11.33	1.9	5(5), 10, 20, 40
XI	$b = 0.1, k_0 = 4.8$ $\bar{D}_1 = 10, \bar{D}_2 = 50$	1.2675	8.984	0.6	5(4), 10
XII	$b = 0.15, k_0 = 4.8$ $\bar{D}_1 = 10, \bar{D}_2 = 50$	1.1222	18.11	0.6	5(4), 10

<sup>a</sup> One Newton iteration was needed for any mesh but the first in each case.

reasonably large is sufficiently accurate to replace integration on the infinite interval  $[T, \infty)$ , here very large  $L$  are insufficient (for  $a = 0.2$  and  $b = 0.15$ ,  $L = 0.5 + 7$  is not large enough for three digits in  $T$ ) because  $r(t)$  decreases slowly as  $t$  increases; viz, for  $t$  large,  $k(t)$  increases like a straight line and  $r \sim k^{-a/b}$ .

Various results are accumulated in Table III. In all cases we took  $a = 0.2$ ,  $\theta_1 = 0.03$ ,  $\theta_2 = 0.09$ ,  $k = 4$ , and  $tol_j = 10^{-5}$ ,  $j = 1, \dots, 4$ . The initial approximation was  $T = 10$ ,  $c = 1.2$ , and a linear interpolation of the boundary conditions for  $D_1, D_2$ . Note the relative insensitivity of the maintainable consumption level  $c$  to changes in the resource endowment  $\bar{D}_1, \bar{D}_2$ .  $c$  is a little more sensitive to changes in the initial stock of capital  $k_0$ . Note also that our transformation for  $D_2$  does better for the case where  $b = 0.1$  or  $b = 0.15$  than where  $b = 0.05$ , even though the latter case is easier in the sense that  $r(t)$  decays faster. The singular coefficient introduced by this transformation causes no problem to COLSYS.

**PROBLEM 5.** Consider a homogeneous, isotropic, spherical shell of constant thickness  $h$  and middle surface radius  $a$ , subject only to an axisymmetric normal distributed surface load  $P_n(\xi)$ , where  $\xi$  is the angle between the meridional tangent at a point of the midsurface of the undeformed shell and the base plane [13]. The governing equations for the finite deformation elastostatics of the shell are given in dimensionless form by

$$\begin{aligned} \mu[\psi'' + \cot \xi \psi' + (\nu - \cot^2 \xi) \psi] - \frac{1}{\sin \xi} (\cos \beta - \cos \xi) \\ = \mu \left( \nu P' + (1 + \nu) \cot \xi P - \frac{1}{\sin \xi} (\sin^2 \xi \rho_H)' - \nu \cos \xi \rho_H \right), \end{aligned} \quad (25)$$

$$\begin{aligned} \epsilon^4 / \mu \left( \phi'' + \cot \xi \phi' + \frac{\cos \beta}{\sin^2 \xi} (\sin \beta - \sin \xi) \right. \\ \left. - \frac{\nu}{\sin \xi} (\cos \beta - \cos \xi) \right) + \frac{\sin \beta}{\sin \xi} \psi = \frac{\cos \beta}{\sin \xi} P. \end{aligned} \quad (26)$$

Here  $\psi$  is a stress function;  $\phi$ -the meridional angle change of the deformed middle surface;  $\beta (= \xi - \phi)$ -the meridional angle of the deformed middle surface;  $P_i$ -representative magnitude of the inward portion of  $P_n(\xi)$ ;  $E$ -Young's modulus;  $\nu$ -Poisson's ratio;  $\mu = p_1 a / (Eh)$ ;  $\epsilon^4 = h^2 / (12(1 - \nu^2) a^2)$ ;  $\rho_H = -P_n \sin \beta / P_i$ ;  $P = -\int^\xi P_n / P_i \cos \beta \sin \eta d\eta$ .

Following [13] we now consider polar dimpling of a complete spherical shell under a particular load distribution of the form

$$P_n(\xi) = P_i(1 - \delta \sin \xi) \quad (27)$$

$\delta > 1$ . This gives

$$P = - \int_0^\xi (1 - \delta \sin \eta) \cos \beta \sin \eta d\eta, \quad \rho_H = - \sin \beta (1 - \delta \sin \xi) \quad (28)$$

and by symmetry it suffices to consider the hemisphere  $0 \leq \xi \leq \pi/2$  under the boundary conditions

$$\phi(0) = \psi(0) = \phi(\pi/2) = \psi(\pi/2) = 0. \tag{29}$$

To evaluate  $P$ , Wan [13] used the approximation  $\cos \beta \approx \cos \xi$ . Here we add another differential equation and boundary condition

$$P' = -(1 - \delta \sin \xi) \cos \beta \sin \xi, \quad P(0) = 0. \tag{30}$$

The computations for system (25), (26), (29), (30) confirm those in [13], which were performed using COLSYS as well. In Fig. 3 we plot curves of  $\beta$  for various values of  $\epsilon$  and  $\mu$ . Numerically, the existence of a dimple means an interior layer in  $\beta$ , around which many mesh elements have to be concentrated for an accurate approximation to this difficult problem. This is done automatically by the code. With  $k = 4$  and  $tol = 10^{-5}$  on all five components of  $\mathbf{z}$ , chains of continuations in  $(\mu, \epsilon)$  were constructed, starting with  $\mathbf{z} \equiv 0$  and a uniform mesh of  $N = 8$  for  $(\mu, \epsilon) = (0.3, 0.3)$ .

In Table IV we list four such chains. In the column entitled “ $N$ ” the final mesh size for each run is listed. For the chains ends we also list, under “layer,” an interval containing the thin layer in  $\beta$  and under “ $N$ -Layer,” the number of elements of the last mesh inside that interval. Chains 2, 3, and 4 are each started where chain 1 ends. All these produce solutions with a dimple in the elastic shell, corresponding to a transition layer in  $\beta$ . The small continuation steps for  $\mu = 0.0001$ ,  $\epsilon = 0.01$  were needed because the layer moves to the left, necessitating drastic mesh adaptation. Also for  $\mu = 0.0001$ ,  $\epsilon = 0.01$ , a second solution branch with no dimple is produced from the initial approximation  $\mathbf{z} = 0$ .

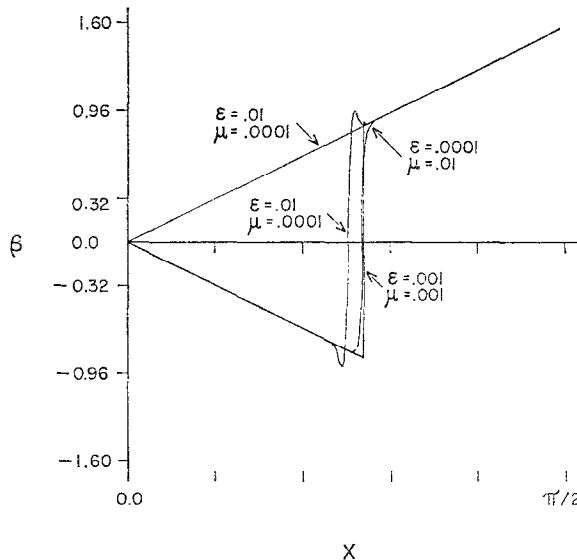


FIG. 3. Problem 5, deformed meridional angle.

TABLE IV  
Problem 5

Chain	$\mu$	$\epsilon$	$N$	Layer	$N$ -Layer
1	0.3	0.3	16		
	0.1	0.1	32		
	0.05	0.05	32		
	0.01	0.01	64	(0.79, 0.91)	33
2	0.01	0.01	64		
	0.003	0.003	108		
	0.001	0.001	72	(0.82, 0.87)	53
3	0.01	0.01	64		
	0.01	0.001	52		
	0.01	0.0001	104	(0.80, 0.91)	58
4	0.01	0.01	64		
	0.005	0.01	64		
	0.002	0.01	64		
	0.001	0.01	64		
	0.0005	0.01	110		
	0.0002	0.01	110		
	0.00015	0.01	110		
	0.0001	0.01	110	(0.73, 0.93)	72

#### ACKNOWLEDGMENT

I would like to express my sincere gratitude to my colleagues Bob Russell and Jan Christiansen, the coauthors of COLSYS, from whom I have gained invaluable experience. I am also very grateful to my colleague Fred Wan, who introduced me to the last two problems.

#### REFERENCES

1. U. ASCHER, J. CHRISTIANSEN, AND R. D. RUSSELL, *Math. Comp.* **33** (1979), 659–679.
2. U. ASCHER, J. CHRISTIANSEN, AND R. D. RUSSELL, in Springer Lecture Notes Computer Science **76** (1979).
3. C. DE BOOR, *SIAM J. Numer. Anal.* **14** (1977), 441–472.
4. U. ASCHER AND R. D. RUSSELL, "Evaluation of B-Splines for Solving Systems of Boundary Value Problems," Computer Sciences Technical Report 77-14, University of British Columbia, Vancouver, 1977.
5. C. DE BOOR AND R. WEISS, "Solveblok; A Package for Solving Almost Block Diagonal Linear Systems, with Applications to Spline Approximation and the Numerical Solution of Ordinary Differential Equations," MRC TSR #1625 Madison, Wisc., 1976.

6. B. A. FINLAYSON, "Method of Weighted Residuals and Variational Principles," Academic Press, New York, 1972.
7. C. DE BOOR AND B. SWARTZ, *SIAM J. Numer. Anal.* **10** (1973), 582-606.
8. P. DEUFLHARD, *Numer. Math.* **33** (1979), 115-146.
9. A. B. VASIL'eva AND V. G. STEL'MAKH, *USSR Comput. Math. Math. Phys.* **17** (1977), 48-58.
10. C. L. HUANG, *Int. J. Non-Linear Mech.* **13** (1978), 55-60.
11. R. M. SOLOW AND F. Y. WAN, *Bell J. Econ.* **7** (1976), 359-370.
12. U. ASCHER AND F. Y. WAN, *SIAM J. Scient. Stat. Comp.* **1** (1980), in press.
13. F. Y. WAN, "Polar Dimpling of Complete Spherical Shells," Technical Report 78-28, Institute of Applied Mathematics, University of British Columbia, Vancouver, 1978.