Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

Continuation of stationary solutions to evolution problems in CONTENT

Yu.A. Kuznetsov, V.V. Levitin, A.R. Skovoroda

Department of Analysis, Algebra and Geometry

AM-R9611 July 31, 1996

Report AM-R9611
ISSN 0924-2953

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Continuation of Stationary Solutions to Evolution Problems in CONTENT

Yu.A. Kuznetsov*†, V.V. Levitin*†, and A.R. Skovoroda†

* *CWI,*
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

† *Institute of Mathematical Problems of Biology,*
Russian Academy of Sciences, Pushchino, Moscow Region,
142292 Russia

Abstract

A numerical algorithm for continuation of stationary solutions to nonlinear evolution problems representable in the form

$$\begin{array}{ll} u_t = F(u_{xx}, u_x, u, x, \alpha), & 0 < x < 1, \\ f^0(u_x, u, \alpha) = 0, & x = 0, \\ f^1(u_x, u, \alpha) = 0, & x = 1, \end{array}$$

is described as implemented in CONTENT. Here $F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $f^{0,1} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are sufficiently smooth nonlinear functions. The algorithm is based on the second-order finite-difference approximation with an adaptive non-uniform mesh selection. Special methods for efficient solution of linear systems appearing in the continuation are presented. Several examples are given.

AMS Subject Classification (1991): 35B32, 35B60, 35G30, 65N06, 65N50, 68N99

Keywords & Phrases: Numerical continuation, partial differential equations, nonlinear boundary-value problems, finite-difference approximation, interactive software.

Note: This work has been partially supported by the Visitor Program of the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO).

1. INTRODUCTION

Nonlinear evolution problems with partial derivatives

$$\begin{array}{ll} u_t = F(u_{xx}, u_x, u, x, \alpha), & 0 < x < 1, \\ f^0(u_x, u, \alpha) = 0, & x = 0, \\ f^1(u_x, u, \alpha) = 0, & x = 1, \end{array} \tag{1.1}$$

where $F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^n$ and $f^{0,1} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are sufficiently smooth nonlinear functions, appear in numerous applications. For example, so called *reaction-diffusion systems* with one spatial variable $x \in [0, 1]$,

$$u_t = D(\alpha)u_{xx} + f(u, \alpha), \quad 0 < x < 1, \tag{1.2}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a smooth function and $D(\alpha)$ is a positive diagonal $n \times n$ matrix smoothly dependent on $\alpha \in \mathbb{R}^m$, with Neumann boundary conditions

$$u_x(0) = u_x(1) = 0 \tag{1.3}$$

belong to the class (1.1). The simplest interesting type of solutions to (1.1) are the *stationary solutions* which are independent of time t and, therefore, satisfy the nonlinear boundary-value problem

$$\begin{array}{ll} F(u_{xx}, u_x, u, x, \alpha) = 0, & 0 < x < 1, \\ f^0(u_x, u, \alpha) = 0, & x = 0, \\ f^1(u_x, u, \alpha) = 0, & x = 1. \end{array} \tag{1.4}$$

If exists, such a stationary solution usually depends on *control parameters* α, and this dependence is of great interest in applications.

The computation of a curve of stationary solutions involves two major ingredients, namely

- discretization;
- continuation.

There are many methods employed in the literature to discretize the problem (1.4): shooting, Galerkin, finite-element, finite-difference, etc. (see the special issue of *SIAM J. Numer. Anal.*, 14(1), 1977, for comparision) In all of the above approaches, the problem (1.4) is reduced to a finite-dimensional *continuation problem*, i.e. computation of a curve defined by

$$\mathcal{F}(Y) = 0, \quad \mathcal{F} : \mathbb{R}^K \to \mathbb{R}^{K-1}, \tag{1.5}$$

provided the *defining function* $\mathcal{F}$ is smooth and satisfies the standard regularity conditions. Here $Y \in \mathbb{R}^K$ represents all the discretization data corresponding to the problem (1.4) and the control parameters. In most of continuation methods (see, for example, [1], [3], [4]), the prediction uses the tangent vector to the curve, and corrections are performed by Newton-like iterations, each of them involving solution of linear systems with a matrix

$$\mathcal{J} = \begin{pmatrix} \mathcal{F}_Y \\ v^T \end{pmatrix}, \tag{1.6}$$

where $\mathcal{F}_Y$ is the Jacobian matrix of the defining function $\mathcal{F}(Y)$ and $v \in \mathbb{R}^K$ is a vector. The matrix $\mathcal{J}$ has a special structure that depends upon the discretization scheme. To solve linear systems with $\mathcal{J}$, one should derive efficient algorithms taking this particular structure into account, because general linar algebra algorithms (like LU-decomposition) would be very inefficient if the dimension K of the discretization system (1.5) is big, since the computational time depends as $\sim K^3$ on K. Moreover, one should also take the advantage of the special structure of $\mathcal{J}$ in its storage and processing.

There are several noninteractive packages to continue numerically a solution to (1.4) with respect to a parameter after conversion to a first-order system. We mention only AUTO86 [5] based on the orthogonal collocation, and BPR-Q [7, 8, 9], where a finite-difference discretization is employed.

The aim of this paper is to describe a continuation technique to compute solution curves of (1.4), which is based on an efiicient finite-difference method (see, for example [16]), and its implementation in CONTENT [10]. CONTENT is an interactive environment for the arclenth continuation of various curves defined by finite-dimensional continuation problems like (1.5) and their visialization. A feature that makes CONTENT particularly useful for the development of continuation algorithms is its built-in ability to associate specific linear algebra routines to each solution type. Thus, the development of a continuation method includes only specification of the defining system (1.5) and its Jacobian matrix, including their possible adaptation along the curve, as well as the development of special routines to handle linear algebraic problems with matrices of the form (1.6). Problem specification, continuation, and visualization are interactively supported by CONTENT.

The paper is organized as follow. In Section 2 we describe the second order finite difference approximation of the problem (1.4), i.e. we define the continuation problem (1.5) and its Jacobian matrix (1.6). Section 3 is devoted to the description of routines to solve linear systems appearing in the continuation of the stationary solutions using an efficient block elimination technique. A special algorithm to compute the determinant of the matrix $\mathcal{J}$ and its certain submatrix is briefly described. These determinants are used as test functions to detect possible branching and limit points along

the solution curves. Section 4 presents an algorithm to adapt mesh distribution while continuing the solution curve. It is based on the requirement for the approximation error of the x-derivatives to be uniformly distributed over the space interval. Section 5 briefly describes how to specify systems of the form (1.1) in CONTENT. Finally, in Section 6, several examples of computations made by CONTENT are presented.

2. FINITE-DIFFERENCE DISCRETIZATION

To approximate (1.4) by a finite-dimensional equation (1.5), introduce a nonuniform mesh

$$\Theta_N = \{0 = x_0 < x_1 < \ldots < x_{N-1} < x_N = 1\} \tag{2.1}$$

with $N > 6$ and set $h_i = x_i - x_{i-1}$ for $i = 1, \ldots, N$. Denote the corresponding mesh values of the solution $u(x_i)$ by $u_i,\ i = 0, 1, \ldots, N$.

For the left end point we take

$$u_x(0) \approx \tilde{u}_x(0) = \hat{\varphi}_0 u_0 + \hat{\varphi}_1 u_1 + \hat{\varphi}_2 u_2, \tag{2.2}$$

where

$$\hat{\varphi}_0 = -\frac{2h_1 + h_2}{h_1(h_1 + h_2)},\ \hat{\varphi}_1 = \frac{h_1 + h_2}{h_1 h_2},\ \hat{\varphi}_2 = -\frac{h_1}{h_2(h_1 + h_2)}.$$

For each internal mesh point with $i = 1, 2, \ldots, N-1$, introduce

$$\tilde{x}_i = x_i + \frac{1}{3}(h_{i+1} - h_i),$$

and take

$$u(\tilde{x}_i) \approx \tilde{u}(\tilde{x}_i) \ = \ \eta_{i-1} u_{i-1} + \eta_i u_i + \eta_{i+1} u_{i+1}, \tag{2.3}$$

$$u_x(\tilde{x}_i) \approx \tilde{u}_x(\tilde{x}_i) \ = \ \varphi_{i-1} u_{i-1} + \varphi_i u_i + \varphi_{i+1} u_{i+1}, \tag{2.4}$$

$$u_{xx}(\tilde{x}_i) \approx \tilde{u}_{xx}(\tilde{x}_i) \ = \ \psi_{i-1} u_{i-1} + \psi_i u_i + \psi_{i+1} u_{i+1}, \tag{2.5}$$

where

$$\eta_{i-1} = -\frac{(h_{i+1} - h_i)(2h_{i+1} + h_i)}{9h_i(h_i + h_{i+1})},\ \eta_i = \frac{(2h_i + h_{i+1})(2h_{i+1} + h_i)}{9h_i h_{i+1}},\ \eta_{i+1} = \frac{(h_{i+1} - h_i)(2h_i + h_{i+1})}{9h_{i+1}(h_i + h_{i+1})},$$

$$\varphi_{i-1} = -\frac{2h_i + h_{i+1}}{3h_i(h_i + h_{i+1})},\ \varphi_i = \frac{h_{i+1} - h_i}{3h_i h_{i+1}},\ \varphi_{i+1} = \frac{h_i + 2h_{i+1}}{3h_{i+1}(h_i + h_{i+1})},$$

and

$$\psi_{i-1} = \frac{2}{h_i(h_i + h_{i+1})},\ \psi_i = -\frac{2}{h_i h_{i+1}},\ \psi_{i+1} = \frac{2}{h_{i+1}(h_i + h_{i+1})}.$$

Finally, for the right end point we take

$$u_x(1) \approx \tilde{u}_x(1) = \hat{\varphi}_{N-2} u_{N-2} + \hat{\varphi}_{N-1} u_{N-1} + \hat{\varphi}_N u_N, \tag{2.6}$$

where

$$\hat{\varphi}_{N-2} = \frac{2h_N + h_{N-1}}{h_N(h_{N-1} + h_N)},\ \hat{\varphi}_{N-1} = -\frac{h_{N-1} + h_N}{h_{N-1} h_N},\ \hat{\varphi}_N = \frac{h_N}{h_{N-1}(h_{N-1} + h_N)}.$$

The above approximations have third-order accuracy for the function $u(x)$ and second-order accuracy for its first and second derivatives (see [2]). More precisely, if $u(x)$ is a smooth vector function, then

$$\|u(\tilde{x}_i) - \tilde{u}(\tilde{x}_i)\| = O(h^3), \quad i = 1, 2, \ldots, N-1,$$

while

$$\|u_x(0) - \tilde{u}_x(0)\| = O(h^2), \quad \|u_x(1) - \tilde{u}_x(1)\| = O(h^2),$$

and

$$\|u_x(\tilde{x}_i) - \tilde{u}_x(\tilde{x}_i)\| = O(h^2), \quad \|u_{xx}(\tilde{x}_i) - \tilde{u}_{xx}(\tilde{x}_i)\| = O(h^2),$$

for $i = 1, 2, \ldots, N-1$. Here

$$h = \max_{i=1,\ldots,N} h_i.$$

Substituting the approximations (2.2)-(2.6) into the problem (1.4), we obtain its Θ_N-discretization in the form $\mathcal{G}(Y) = 0$ with

$$Y = \begin{pmatrix} u_0 \\ u_1 \\ \cdots \\ u_{N-1} \\ u_N \\ \alpha \end{pmatrix}$$

and

$$\mathcal{G}(Y) = \begin{pmatrix} f^0(\tilde{u}_x(0), u_0, \alpha) \\ F(\tilde{u}_{xx}(\tilde{x}_1), \tilde{u}_x(\tilde{x}_1), \tilde{u}(\tilde{x}_1), \tilde{x}_1, \alpha) \\ F(\tilde{u}_{xx}(\tilde{x}_2), \tilde{u}_x(\tilde{x}_2), \tilde{u}(\tilde{x}_2), \tilde{x}_2, \alpha) \\ \ldots \\ F(\tilde{u}_{xx}(\tilde{x}_{N-2}), \tilde{u}_x(\tilde{x}_{N-2}), \tilde{u}(\tilde{x}_{N-2}), \tilde{x}_{N-2}, \alpha) \\ F(\tilde{u}_{xx}(\tilde{x}_{N-1}), \tilde{u}_x(\tilde{x}_{N-1}), \tilde{u}(\tilde{x}_{N-1}), \tilde{x}_{N-1}, \alpha) \\ f^1(\tilde{u}_x(1), u_N, \alpha) \end{pmatrix}.$$

Notice that $K = \dim Y = (N+1)n + m$. To obtain a continuation problem of the general form (1.5), one has to append $(m-1)$ scalar equations

$$\tilde{g}^i(u_0, u_1, \ldots, u_{N-1}, u_N, \alpha) = 0, \quad i = 1, 2, \ldots, m-1,$$

which represent the discretizations of some user-supplied conditions defined by vanishing of $m-1$ functionals

$$g^i[u] = 0, \quad i = 1, 2, \ldots, m-1,$$

on the continued solution. In the case of one control parameter, no such extra conditions are required.

Therefore, the continuation problem (1.5) is defined by

$$\mathcal{F}(Y) = \begin{pmatrix} \mathcal{G}(Y) \\ \tilde{g}(Y) \end{pmatrix}, \tag{2.7}$$

where $\tilde{g} = (\tilde{g}^1, \ldots, \tilde{g}^{m-1})^T$. Thus, $\mathcal{F} : \mathbb{R}^K \to \mathbb{R}^{K-1}$.

Taking into account (2.2)-(2.6), the Jacobian matrix of (2.7) can be written in the form

$$\mathcal{F}_Y = \begin{pmatrix} a_0 & b_0 & c_0 & & & & & & \delta_0 \\ a_1 & b_1 & c_1 & & & & & & \delta_1 \\ & a_2 & b_2 & c_2 & & & & & \delta_2 \\ & & & & & & & & \\ & & & & a_{N-2} & b_{N-2} & c_{N-2} & & \delta_{N-2} \\ & & & & & a_{N-1} & b_{N-1} & c_{N-1} & \delta_{N-1} \\ & & & & & a_N & b_N & c_N & \delta_N \\ q_0 & q_1 & q_2 & & & q_{N-2} & q_{N-1} & q_N & \Psi \end{pmatrix},$$

where $a_k, b_k, c_k,\ k = 0, 1, \ldots, N$ are $n \times n$ matrices, $\delta_k,\ k = 0, 1, \ldots, N$ are $n \times m$ matrices, $q_0, q_1, \ldots, q_N$ are $(m-1) \times n$ matrices, Ψ is an $(m-1) \times m$ matrix (all undisplayed elements of $\mathcal{F}_Y$ are zeros). These matrices are given by the following expressions:

$$a_0 = \frac{\partial f^0}{\partial u_0} = \frac{\partial f^0}{\partial u_x}\hat{\varphi}_0 + \frac{\partial f^0}{\partial u}\ ,\ b_0 = \frac{\partial f^0}{\partial u_1} = \frac{\partial f^0}{\partial u_x}\hat{\varphi}_1\ ,\ c_0 = \frac{\partial f^0}{\partial u_2} = \frac{\partial f^0}{\partial u_x}\hat{\varphi}_2\ ,\ \delta_0 = \frac{\partial f_0}{\partial \alpha}\ ,$$

$$\begin{aligned}
a_i &= \frac{\partial F}{\partial u_{i-1}} = \frac{\partial F}{\partial u_{xx}}\psi_{i-1} + \frac{\partial F}{\partial u_x}\varphi_{i-1} + \frac{\partial F}{\partial u}\eta_{i-1}\ ,\\
b_i &= \frac{\partial F}{\partial u_i} = \frac{\partial F}{\partial u_{xx}}\psi_i + \frac{\partial F}{\partial u_x}\varphi_i + \frac{\partial F}{\partial u}\eta_i\ ,\\
c_i &= \frac{\partial F}{\partial u_{i+1}} = \frac{\partial F}{\partial u_{xx}}\psi_{i+1} + \frac{\partial F}{\partial u_x}\varphi_{i+1} + \frac{\partial F}{\partial u}\eta_{i+1}\ ,\\
\delta_i &= \frac{\partial F}{\partial \alpha}\ ,
\end{aligned}$$

where $i = 1, 2, \ldots, N-1$,

$$a_N = \frac{\partial f^1}{\partial u_{N-2}} = \frac{\partial f^1}{\partial u_x}\hat{\varphi}_{N-2}\ ,\ b_N = \frac{\partial f^1}{\partial u_{N-1}} = \frac{\partial f^1}{\partial u_x}\hat{\varphi}_{N-1}\ ,\ c_N = \frac{\partial f^1}{\partial u_N} = \frac{\partial f^1}{\partial u_x}\hat{\varphi}_N + \frac{\partial f^1}{\partial u}\ ,\ \delta_N = \frac{\partial f^1}{\partial \alpha}\ .$$

Finally,

$$q_j = \frac{\partial \tilde{g}}{\partial u_i}\ ,\ \Psi = \frac{\partial \tilde{g}}{\partial \alpha}\ ,$$

where $j = 0, 1, \ldots, N$.

3. HANDLING OF SPECIAL LINEAR SYSTEMS

As has been pointed out in Section 1, at correction iterations in CONTENT, one has to solve linear systems of the following form

$$\begin{pmatrix} \mathcal{F}_Y \\ v^T \end{pmatrix} Y = Z,$$

where $v \in \mathbb{R}^K$ is the tangent vector to the curve at a previous point. Therefore, this system can be written as

$$\begin{pmatrix}
a_0 & b_0 & c_0 & & & & & & \delta_0 \\
a_1 & b_1 & c_1 & & & & & & \delta_1 \\
 & a_2 & b_2 & c_2 & & & & & \delta_2 \\
 & & & & & & & & \\
 & & & & a_{N-2} & b_{N-2} & c_{N-2} & & \delta_{N-2} \\
 & & & & & a_{N-1} & b_{N-1} & c_{N-1} & \delta_{N-1} \\
 & & & & & a_N & b_N & c_N & \delta_N \\
p_0 & p_1 & p_2 & & & p_{N-2} & p_{N-1} & p_N & \Phi
\end{pmatrix}
\begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ \\ Y_{N-2} \\ Y_{N-1} \\ Y_N \\ \gamma \end{pmatrix}
=
\begin{pmatrix} Z_0 \\ Z_1 \\ Z_2 \\ \\ Z_{N-2} \\ Z_{N-1} \\ Z_N \\ \mu \end{pmatrix} \qquad (3.1)$$

The right-hand side of (3.1) is composed of the given vectors $Z_0, Z_1, \ldots, Z_N \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^m$. The solution is $Y_0, Y_1, \ldots, Y_N \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}^m$. The matrix row $(a_0, b_0, c_0, \ldots, \delta_0)$ and $(a_N, b_N, c_N, \ldots, \delta_N)$ corresponds to the discretization of the left and right boundary conditions, while the matrix rows $(a_i, b_i, c_i, \ldots, \delta_i)$ with $i = 1, 2, \ldots, N-1$ are due to the discretization of the differential equations along the space interval. Finally, the bottom matrix row $(p_0, p_1, \ldots, p_N, \Phi)$ corresponds to the user conditions and the appended vector v tanget to the continued curve. In CONTENT, each correction iteration require the solution of two linear systems (3.1) with the same matrix but with different right-hand sides. Therefore, any solving procedure should be separated into a *decomposition* and a *backsubstitution*.

Since the matrix of the system (3.1) has the special structure, we solve it by a modified block elimination method (cf., [16], Ch. 2.4). Let us seek the solution of (3.1) in the form

$$Y_i = A_i Y_{i+1} + B_i + D_i\gamma, \quad i = 0, 1, \ldots, N-1, \tag{3.2}$$

where A_i and D_i are unknown matrices of dimension $n \times n$ and $n \times m$, respectively, while $B_i \in \mathbb{R}^n$ is an unknown vector. Using the representation (3.2) and the regular matric rows of (3.1), one obtains for $i = 1, 2, \ldots, N-1$

$$[(a_iA_{i-1} + b_i)A_i + c_i]Y_{i+1} + [(a_iA_{i-1} + b_i)D_i + (a_iD_{i-1} + \delta_i)]\gamma + [(a_iA_{i-1} + b_i)B_i - (Z_i - a_iB_{i-1})] = 0.$$

Since this equation must hold for arbitrary Y_{i+1} and γ, we conclude

$$\left\{ \begin{array}{lcl} (a_iA_{i-1} + b_i)A_i & = & -c_i \\ (a_iA_{i-1} + b_i)B_i & = & (Z_i - a_iB_{i-1}) \\ (a_iA_{i-1} + b_i)D_i & = & -(a_iD_{i-1} + \delta_i). \end{array} \right. \tag{3.3}$$

Equations (3.3) allow to find the matrices A_i and D_i and the vector B_i if A_{i-1}, D_{i-1} and B_{i-1} are known. Having in mind (3.3), the problem of finding the coefficients in (3.2) is reduced to determining A_0, D_0 and B_0.

From the top matrix raw in (3.1) and the representation (3.2) for Y_0 and Y_1, follows

$$[(a_0A_0 + b_0)A_1 + c_0]Y_2 + [(a_0A_0 + b_0)D_1 + (a_0D_0 + \delta_0)]\gamma + [(a_0A_0 + b_0)B_1 + (a_0B_0 - Z_0)] = 0.$$

Since this equation must also hold for arbitrary Y_2 and γ, we should have

$$\begin{aligned} &(a_0A_0 + b_0)A_1 + c_0 = 0, \\ &(a_0A_0 + b_0)B_1 + (a_0B_0 - Z_0) = 0, \\ &(a_0A_0 + b_0)D_1 + (a_0D_0 + \delta_0) = 0. \end{aligned}$$

Using these equations and (3.3) for $i = 1$, we obtain the following equations to compute A_0, B_0 and D_0:

$$\begin{array}{rcl} (a_0 - c_0c_1^{-1}a_1)A_0 & = & (c_0c_1^{-1}b_1 - b_0), \\ \left[a_0 - (a_0A_0 + b_0)(a_1A_0 + b_1)^{-1}a_1\right] B_0 & = & \left[Z_0 - (a_0A_0 + b_0)(a_1A_0 + b_1)^{-1}Z_1\right], \\ \left[a_0 - (a_0A_0 + b_0)(a_1A_0 + b_1)^{-1}a_1\right] D_0 & = & -\left[\delta_0 - (a_0A_0 + b_0)(a_1A_0 + b_1)^{-1}\delta_1\right]. \end{array}$$

The above formulas allow to compute the coefficients in (3.2).

For the recurrent computation of Y_i via (3.2), the values of Y_N and γ are required. To find Y_N we notice that (3.1) implies

$$a_NY_{N-2} + b_NY_{N-1} + c_NY_N + \delta_N\gamma = Z_N.$$

This equation and the representation (3.2) for $i = N-2$ and $i = N-1$, lead to

$$\begin{array}{rcl} [(a_NA_{N-2} + b_N)A_{N-1} + c_N]\, Y_N & = & -\left[(a_NA_{N-2} + b_N)D_{N-1} + (a_ND_{N-2} + \delta_N)\right]\gamma \\ & & +\ \left[(Z_N - a_NB_{N-2}) - (a_NA_{N-2} + b_N)B_{N-1}\right] \end{array}$$

as the equation for Y_N. Therefore,

$$Y_N = U_N + W_N\gamma, \tag{3.4}$$

where vector $U_N \in \mathbb{R}^n$ and $n \times m$-matrix W_N are defined from the equations

$$\begin{aligned} &[(a_NA_{N-2} + b_N)A_{N-1} + c_N]U_N = [(Z_N - a_NB_{N-2}) - (a_NA_{N-2} + b_N)B_{N-1}], \\ &[(a_NA_{N-2} + b_N)A_{N-1} + c_N]W_N = -[(a_NA_{N-2} + b_N)D_{N-1} + (a_ND_{N-2} + \delta_N)] \end{aligned}$$

Similarly to (3.4), we represent the rest of the solution vector as

$$Y_i = U_i + W_i\gamma, \quad i = 0, 1, \ldots, N-1. \tag{3.5}$$

Substituting (3.5) into (3.2), we obtain the recurrent relations for U_i and W_i:

$$\begin{aligned} U_i &= A_i U_{i+1} + B_i, \\ W_i &= A_i W_{i+1} + D_i. \end{aligned}$$

Finally, the unknown vector $\gamma \in \mathbb{R}^m$ can be found from the bottom matrix row in (3.1), i.e.

$$\left(\sum_{i=0}^{N} p_i W_i + \Phi\right)\gamma = \mu - \sum_{i=0}^{N} p_i U_i.$$

The above solving procedure allows for the separation into the right-hand side independent decomposition and the backsubstitution with different right-hand sides.

In the manipulations above it is assumed that all matrices appearing in the intermediate linear systems are nonsingular, as well as the conditions for stability of the reccurent procedure (3.2) are valid:

$$\|b_i^{-1}a_i\| + \|b_i^{-1}c_i\| \le 1$$

(see [16]). This implicitly defines a class of nonlinear evolution problems (1.1) which stationary solutions can be continued by the described algorithm using a given mesh Θ_N.

One can see that the computational time required to solve the system (3.1) behaves as $\sim n^3 N$, i.e. linearly increases with the number of the mesh points N.

Remark: To detect branching and limit points along the solution curve, one needs to compute the determinants of the matrix $\mathcal{J}$ from (3.1) and its submatrix $\mathcal{M}$ obtained by ignoring the bottom matrix row $(p_0, p_1, \ldots, p_N, \Phi)$, as well as the right matrix column $(\delta_0, \delta_1, \ldots, \delta_N)^T$. To compute these determinants simultaneously, the matrix $\mathcal{J}$ is reduced to the upper trianglular form by means of the Gauss elimination with pivoting, taking into account its block structure. More precisely, first we look for the dominant pivot element in the submatrix $\mathcal{M}$; if it vanishes, $\det \mathcal{M} = 0$, and then the pivot elements are searched for in the whole matrix $\mathcal{J}$.

4. MESH ADAPTATION

It is well known that the accuracy of the finite-difference approximation of differential equations on any mesh strongly depends on the spatial behavior of their solutions. It means that the mesh Θ_N should be selected according to this behavior. Since the shape of the solution usually vary with parameters, the mesh has to be adaptive.

4.1 Mesh point selection criterium

One of the possible criteria for mesh selection is the requirement for the approximation error for the first derivative to be uniformely distributed over the space interval (see, for example [11], [15]).

The approximation (2.4) on the nonuniform mesh (2.1) leads to the following expression

$$u_x(\tilde{x}_i) = \tilde{u}_x(\tilde{x}_i) + R(\tilde{x}_i), \quad i = 1, 2, \ldots, N-1,$$

with the local error

$$R(\tilde{x}_i) \approx \frac{1}{18} u_{xxx}(\tilde{x}_i)(h_i^2 + h_i h_{i+1} + h_{i+1}^2),$$

where $h_i = x_i - x_{i-1} > 0, \quad i = 1, \ldots, N$. To estimate the spatial dependence of the error, assume that N is big enough and consider a smooth functiuon $h = h(x),\ h(x_i) = h_i$. Then

$$h_{i+1} \approx h_i[1 + h_x(x_i)], \quad i = 1, 2, \ldots, N-1.$$

Therefore,

$$R(\tilde{x}_i) \approx \frac{1}{6} u_{xxx}(x_i) h_i^2.$$

Clearly, if we take

$$h_i = C \|u_{xxx}(x_i)\|^{-\frac{1}{2}}$$

with a constant $C > 0$, the norm of the leading term of the local error R will be uniformely distributed over the internal points, having the value $\frac{1}{6}C^2$. Note that in this case we have $h_x = -\frac{1}{2}[\ln \|u_{xxx}\|]_x h$. Therefore, for sufficiently smooth functions $u(x)$, $|h_x| \ll 1$.

Finally, we can write our mesh point selection criterium in the form

$$S(x_i)(x_i - x_{i-1}) = C, \tag{4.1}$$

where $S(x) = \|u_{xxx}(x)\|^{\frac{1}{2}}$. In the implementation, we compute $u_{xxx}(x_i)$ using five-point finite-difference approximation on the previous mesh and take

$$S(x) = \begin{cases} \|u_{xxx}(x)\|^{\frac{1}{2}}, & \|u_{xxx}(x)\| > \varepsilon^2, \\ \varepsilon, & \|u_{xxx}(x)\| \le \varepsilon^2, \end{cases}$$

with some $\varepsilon \ll 1$. This modification is natural, since for small $\|u_{xxx}\|$ the equation (4.1) with modified $S(x)$ is satisfied on the uniform mesh, where the approximation formula (2.4) is exact for polynomial functions of degree two.

4.2 Numerical procedure

To solve (4.1) for x_i, one can use different approaches. The simplest one is to consider (4.1) as first-order analog of

$$\int_{x_{i-1}}^{x_i} S(x)dx \;=\; C \;=\; \frac{1}{N}\int_0^1 S(x)dx \tag{4.2}$$

(see [15]). Solving the equation (4.2) using the linear interpolation of $S(x)$ between the mesh points, amounts to solution of a quadric for each x_i, $i = 1, 2, \ldots, N-1$. The linear interpolation guarantees the monotonicity of the mesh, i.e. $x_i > x_{i-1}$.

Thus obtained solution Θ_N to (4.1) may be improved by Newton iterations applied to the nonlinear system

$$\begin{cases} S(x_1)x_1 - S(x_2)(x_2 - x_1) = 0, \\ S(x_i)(x_i - x_{i-1}) - S(x_{i+1})(x_{i+1} - x_i) = 0, \quad i = 2, \ldots, N-2, \\ S(x_{N-1})(x_{N-1} - x_{N-2}) - S(1)(1 - x_{N-1}) = 0, \end{cases}$$

which is another form of the presentaion of (4.1). The Jacobian matrix of this system is tridiagonal, therefore we can solve the linear systems appearing at the Newton iterations by the standard elimination method ([16], Ch. 2.1). The condition of stability for this method, as well as the monotonicity of the resulting mesh, are checked during iterations. The iterations are also aborted when the norm of the left-hand side happened to increase.

Remark: To recompute the approximate solution of (1.4) using the new mesh, two types of interpolation are employed: either the interpolation by the global cubic spline or the linear interpolation.

5. PROBLEM SPECIFICATION IN CONTENT

To specify a new problem (1.1) in CONTENT, one should open the system specification window and fill in its fields with the names and formulas describing the system. The following table gives the correspondence between usual mathematical notations and those used in CONTENT.

term	specification
u_t	`u_t`
u_x	`u_x`
u_{xx}	`u_x_x`
f^0_{i+1}	`BC_i`
f^1_{i+1}	`BC_i`

As an example let us specify the following two-dimensional problem which will also be considered in the next Section:

$$\begin{cases} u_t = u_{xx} - \frac{n(n+1)}{\varepsilon^2} u^{1+\frac{2}{n}}, \\ v_t = v_{xx} - \frac{n^2}{\varepsilon^2} v, \\ u(0) = 1, \quad v(0) = \exp\left(-\frac{n}{\varepsilon}\right), \\ u(1) = \left(1 + \frac{1}{\varepsilon}\right)^{-n}, \quad v(1) = 1. \end{cases} \tag{5.1}$$

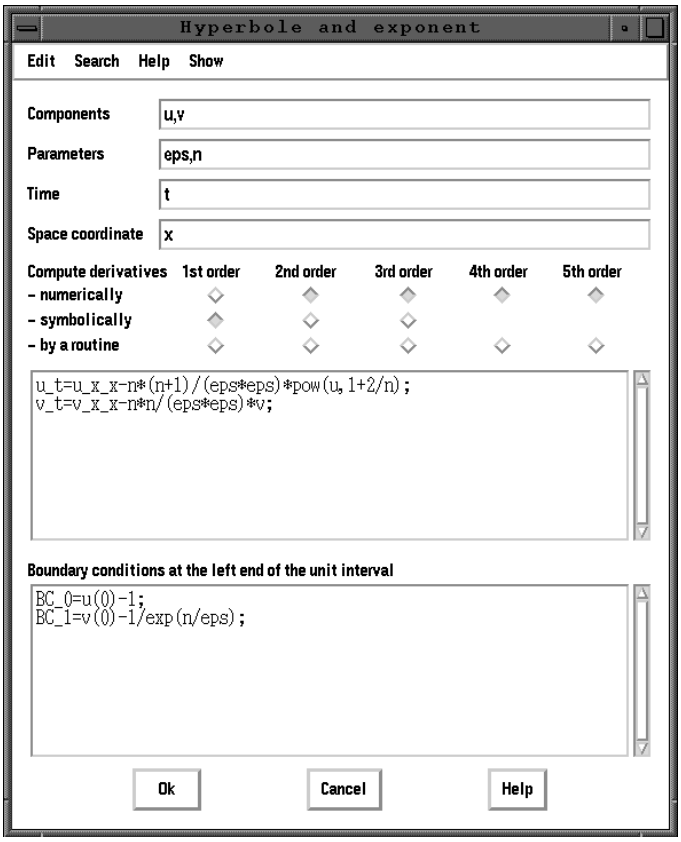

Figure 5.1: Example of system specification

Figure 5.1 shows the system specification window for this example. Four top fields contain the names `u,v,eps,n,t,x` of all the variables used: components (u and v), parameters (ε and n), time (t), and space corrdinate (x). The equations are specified in the field in the middle of the window. Note that in CONTENT these are assignment statements in the `C` programming languge. The boundary conditions at the left end of the unit interval are specified in the field at the bottom of the window. The values of components and their derivatives are denoted by `u(0)`, `v(0)`, `u_x(0)`, and `v_x(0)`. `BC_0` and `BC_1` are the artificial names used to refer to the first, second, etc. boundary conditions. Similarly, the boundary conditions at the right end of the unit interval can be specified after switching to an

appropriate input field via the window's menu. Input fields for boundary conditions overlap each other with only one field visible at a time. The values of the components and their derivatives at the right end of the interval are denoted by `u(1)`, `v(1)`, `u_x(1)`, and `v_x(1)`. This allows one to use the same names (`BC_0`, `BC_1`, etc.) for both left and right boundary conditions.

Check boxes in the middle of the window allow one to choose the method CONTENT will use when computing derivatives of the equations and the boundary conditions. The values of the derivatives are used to fill in the matrix in (3.1). In our example the first order derivatives will be computed symbolically, that is by a program for derivatives which CONTENT internally creates using the equations and boundary conditions. For more information see [12].

6. EXAMPLES

Here we present several examples of computations made by CONTENT.

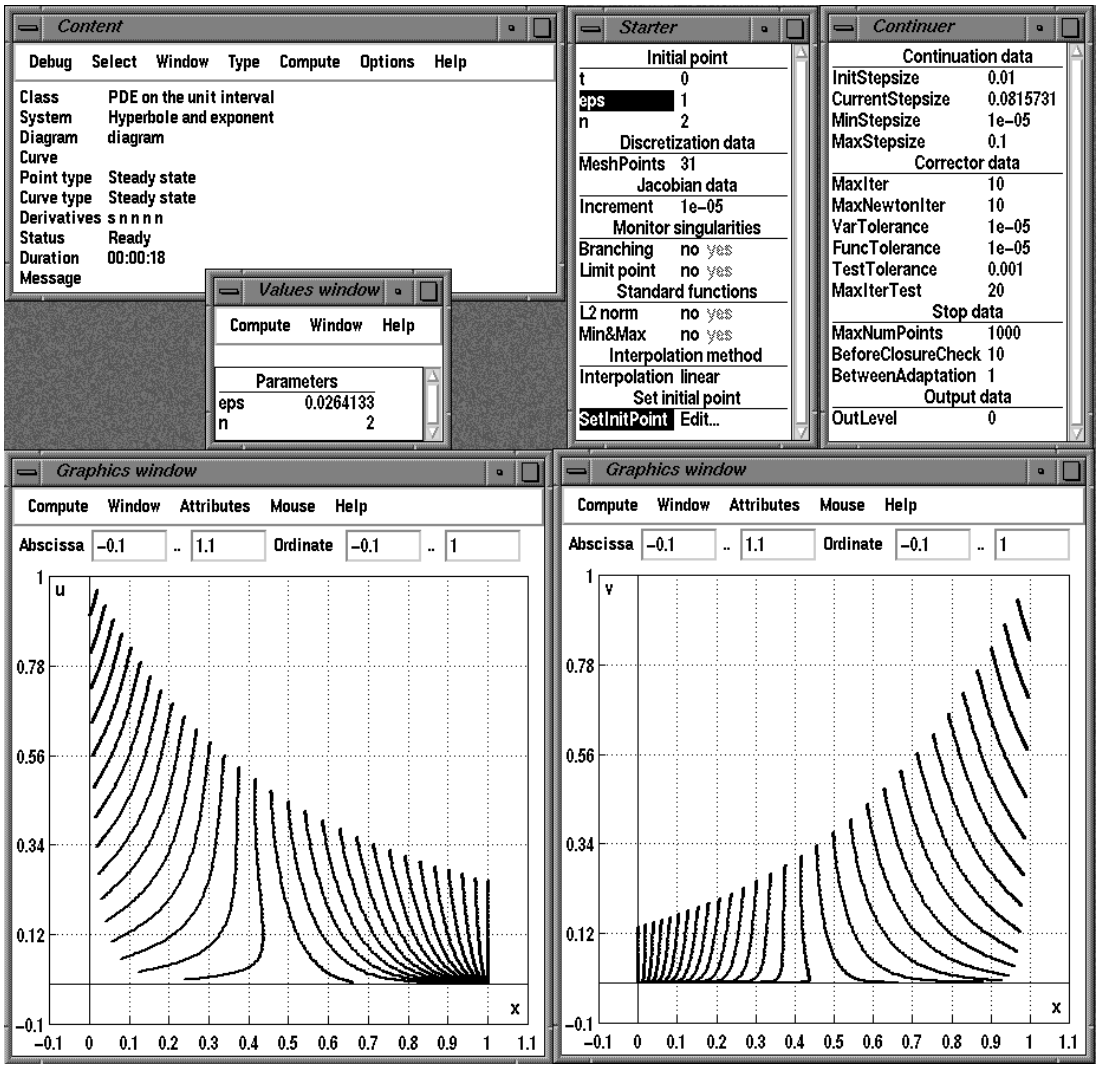

Figure 6.1: Tutorial example

6.1 Tutorial example

Consider the problem given by (5.1) from the previous Section. It has an explicit stationary solution:

$$u(x) \;=\; \left(1+\frac{x}{\varepsilon}\right)^{-n}, \quad v(x) \;=\; \exp\left[\frac{n(x-1)}{\varepsilon}\right].$$

Clearly, $\|u_x\|^2 + \|v_x\|^2 \to \infty$ at both ends of the interval $[0,1]$ as $\varepsilon \to 0$, due to the u-componet at $x = 0$ and the v-componet at $x = 1$, respectively. The figure illustrates the mesh adaptation while continuing the solution to the problem starting from $\varepsilon = 1$ to $\varepsilon = 0.026\ldots$. Here the value of the parameter $n = 2$ and the number of mesh points is $N = 31$. As the initial distribution of $u(x)$ and $v(x)$ at $\varepsilon = 1$ we took the corresponding explicit solutions given above. The Figure 6.1 presents the trajectories of the mesh points as the parameter varies; they visibly accumulate on the ends of the interval.

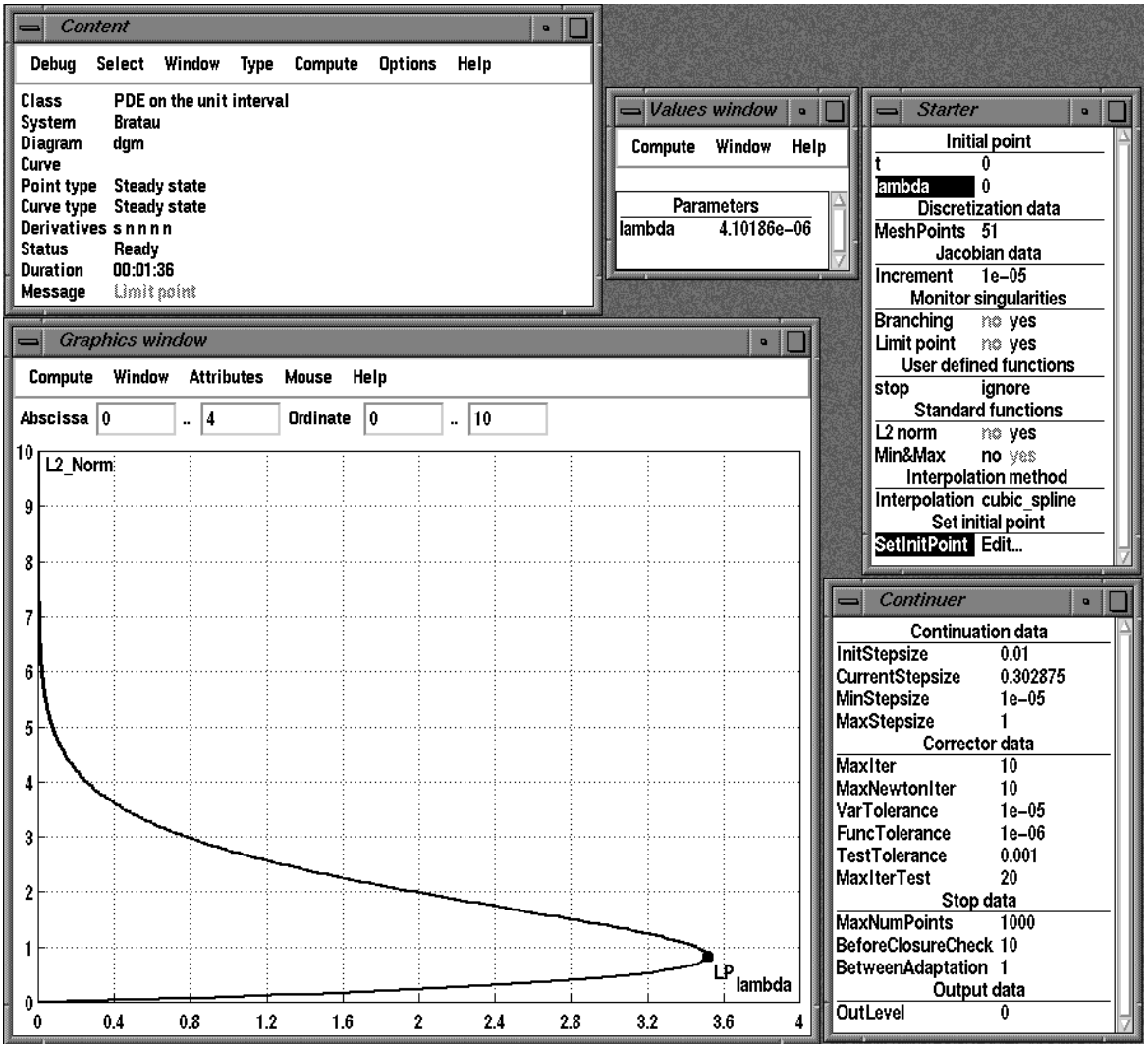

Figure 6.2: Bratu equation

6.2 Bratu equation

As the second example, consider the following scalar boudary-value problem

$$u_{xx} + \lambda e^u = 0, \quad u(0) = u(1) = 0.$$

At $\lambda = 0$, the distribution $u(x) \equiv 0$ is a regular solution to the problem, which was taken as the initial point for the continuation. The dependence of the L_2-norm of the solution on λ is given in Figure 6.2 with the number of mesh points $N = 51$. There is a limit (fold) point on the solution curve at $\lambda = 3.51\ldots$, that has been detected as $\det \mathcal{M} = 0$ (see Remark in Section 3) and is indicated by LP in the figure. The continuation can be successefully performed up to very small $\lambda < 10^{-20}$.

6.3 A singular-perturbed problem
Next, consider a singular-perturbed boudary-value problem by J. Lorenz [13]

$$\begin{cases} \varepsilon u_{xx} - \frac{1}{4}\lambda[(u^2-1)^2]_x - u = 0, \\ u(0) = A, \quad u(1) = \gamma, \end{cases}$$

where $\varepsilon \ll 1$. This problem is used as the demo example `spb` in AUTO94 [6].

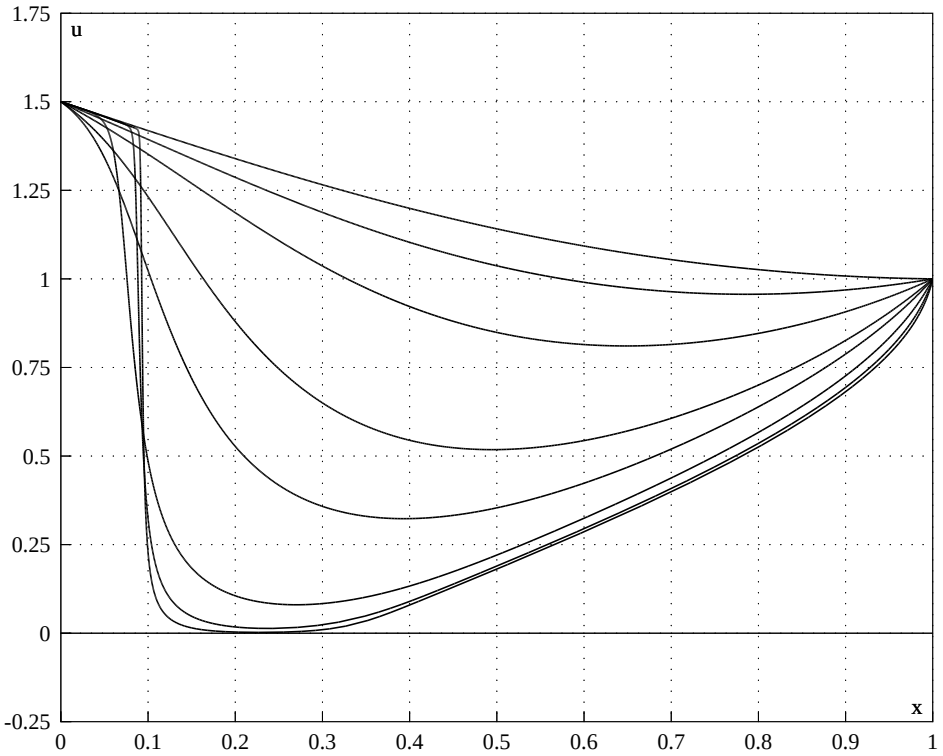

Figure 6.3: J. Lorenz's example: continuation in ε.

Figure 6.3 shows the results of the continuation of the solution for fixed

$$A = \frac{3}{2}, \quad \gamma = 1, \quad \lambda = 1,$$

with respect to ε from $\varepsilon = 1$ to $\varepsilon = 0.001$, starting from the exact solution to the problem at $\lambda = 0$ and using $N = 501$. The solutions are plotted for $\varepsilon = 1.0, 0.5, 0.25, 0.1, 0.005, 0.0025$, and 0.001. The value of ε decreases "up down". The formation of an internal boundary layer near $x = 0.1$ is clearly visible.

Taking the solution corresponding to $\varepsilon = 0.0025$ as initial, we have continued it with respect to the parameter γ. Representative solutions are depicted in Figure 6.4, where the appearance of the second internal boundary layer is evident. This boundary layer moves from the right to the left as γ decreases and collides with the first boundary layer forming a single layer, which continues to drift to left.

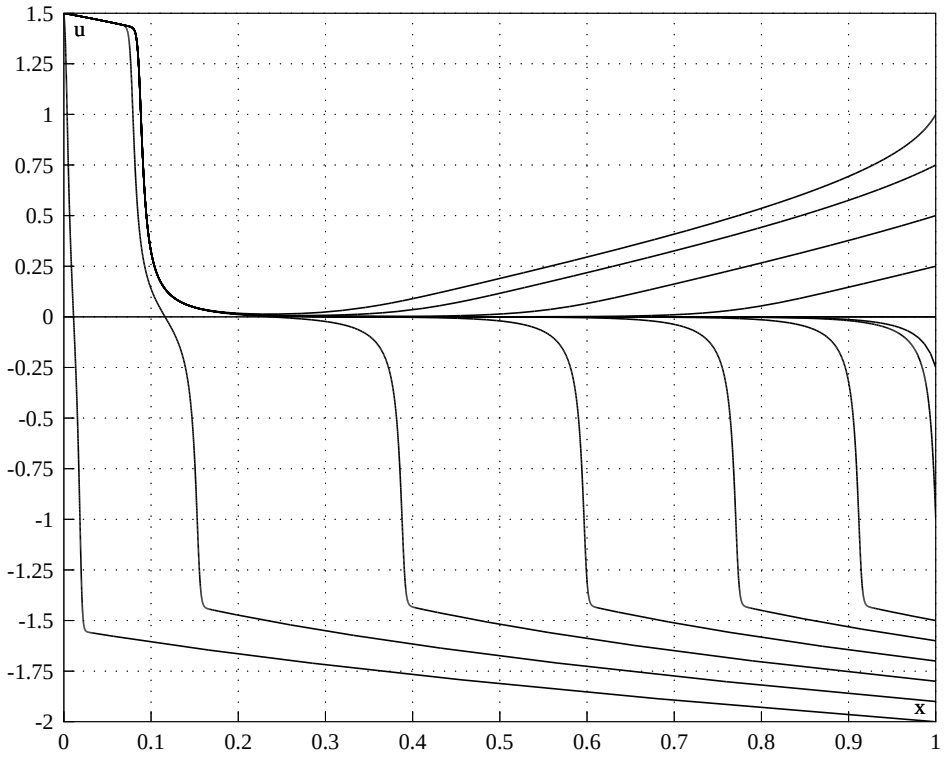

Figure 6.4: J. Lorenz's example: continuation in γ

Because of the presense of the internal boundary layers, the linear interpolation was employed to recompute the solution of (1.4) on the new mesh (see Remark in Section 4).

6.4 Nonlinear elasticity problem
Finally, let us consider the following equation

$$u_{rr}\left[(1+u_r)^2+u_r\left(1+\frac{u_r}{2}\right)+2Aq\left(1+\frac{q}{2}\right)\right]+\frac{2A}{r}(u_r-q)(1+q)(1+u_r)+$$
$$\frac{2}{r}\left\{u_r\left(1+\frac{u_r}{2}\right)(2B+u_r-Aq)+2q\left(1+\frac{q}{2}\right)[A(u_r-q)-B(1+q)]\right\}=0,$$

with the boundary conditions

$$u(1)=\delta,\quad u(R)=\frac{\delta}{R^2},$$

for $R\gg 1$. Here

$$q=\frac{u}{r}.$$

This is a generalization of the classical spherically symmetric linear Lame problem to the case of finite deformations that can be obtained following Rabotnov [14] (Ch. 7.9). In this case $u(r)$ is a scaled radial displacement as the function of the radius r,

$$A=\frac{\lambda}{\lambda+2\mu},\quad B=\frac{\mu}{\lambda+2\mu},$$

where λ and μ are Lame elastic coefficients. In this example the term with the highest derivative can vanish at the left end point $r=1$.

Figure 6.5 presents the continuation of the solution to the above boubdary-value problem with respect to δ at $A=1,\ B=0$, that corresponds to the incompressible media.

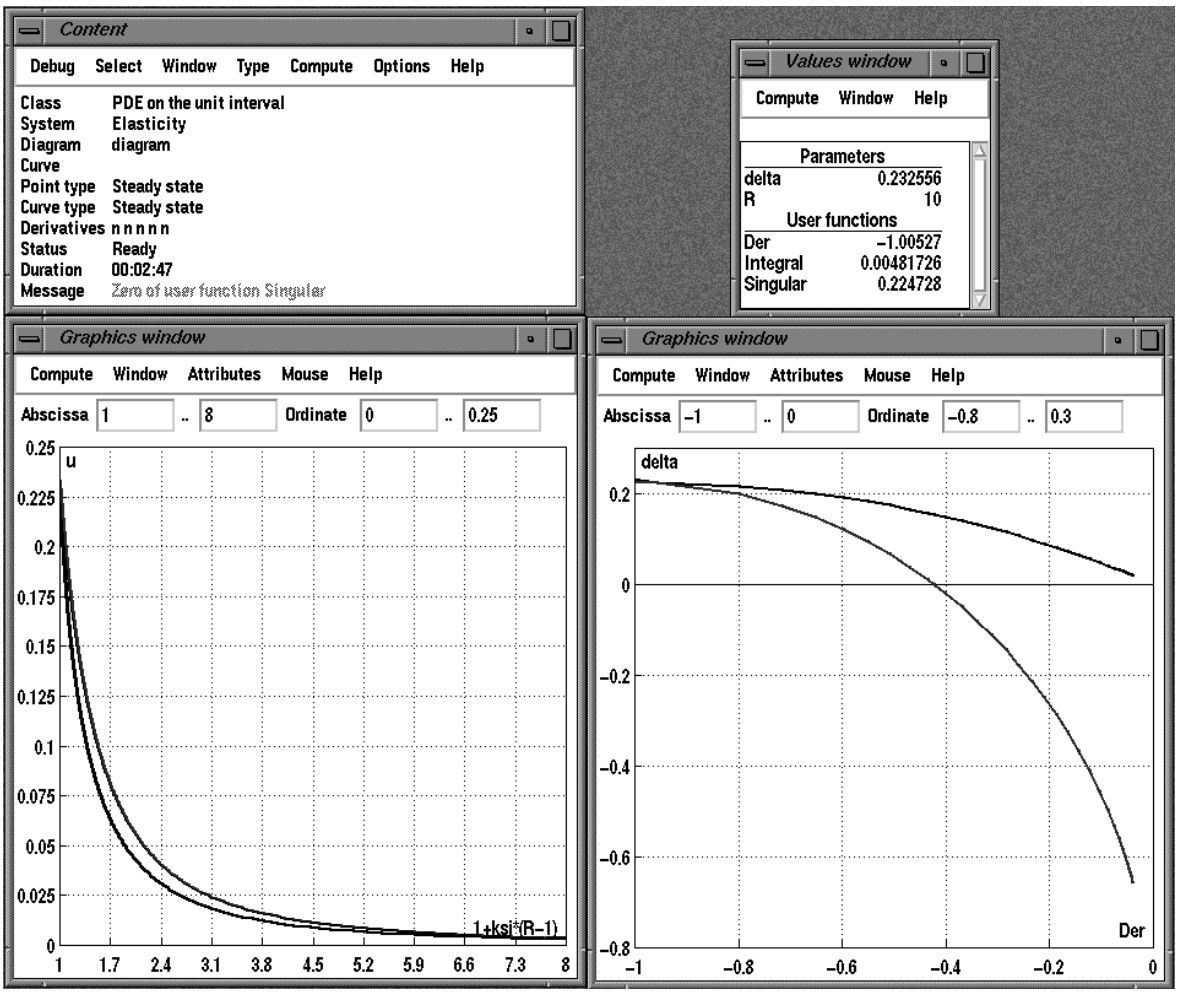

Figure 6.5: Nonlinear elasticity problem

The top curve in the right window represents the dependence of $\texttt{Der} = u_r(1)$ on the boundary value $\texttt{delta} = u(1)$; the bottom curve corresponds to zero of the coefficient of the u_{rr}-term at $r = 1$. Using $N = 51$ mesh points, a common point with $\texttt{Der} = -1$ has been reached.

As can be shown, for $A = 1,\ B = 0$, the original equation has the first integral

$$u_r \left(1 + \frac{u_r}{2}\right) + q(2 + q) = 0$$

The mean squared value of this integral over the interval $[1, R]$ was monitored in the δ-continuation. In the present example this value increased from $\sim 10^{-7}$ at the begining of the curve ($\delta = 0.02$) to $0.0048\ldots$ at the last point of the continuation.

In the left window in Figure 6.5, the dependences $u(r)$ are presented at the last point of the continuation for the linear approach (top curve) and the nonlinear approach (bottom curve). Note, that in the linear case (when $u_r \ll 1$), the function

$$u(r) = \frac{\delta}{r^2}$$

represents the exact solution to the problem.

Acknowledgement

The authors are thankful to Prof. E.Doedel (Concordia University, Montreal) for reading the draft of the manuscript and useful suggestions that allowed to improve the accuracy of approximation of the boundary-value problem.

REFERENCES

1. E.L. Allgower and K. Georg. *Numerical Continuation Methods.* Springer-Verlag, Berlin, Heidelberg, New York, 1990.
2. E. Doedel. The construction of finite difference approximations to ordinary differential equations. *SIAM J. Numer. Anal.*, 15:450–465, 1978.
3. E.J. Doedel, H.B. Keller, and J.P. Kernévez. Numerical analysis and control of bifurcation problems: (I) Bifurcation in finite dimensions. *Int. J. Bifurcation and Chaos*, 1:493–520, 1991.
4. E.J. Doedel, H.B. Keller, and J.P. Kernévez. Numerical analysis and control of bifurcation problems: (II) Bifurcation in infinite dimensions. *Int. J. Bifurcation and Chaos*, 1:745–772, 1991.
5. E.J. Doedel and J.P. Kernévez. AUTO: Software for continuation problems in ordinary differential equations with applications. Technical report, California Institute of Technology, 1986. Applied Mathematics.
6. E.J. Doedel, X.J. Wang, and T.F. Fairgrieve. AUTO94 *Software for continuation and bifurcation problems in ordinary differential equations. CRPC-95-2.* Pasadena, CA 91125, 1994.
7. S.I. Fadeev. A program for numerical solution of nonlinear boundary-value problems for systems of ordinary differential equations. Part I. Technical report, Institute of Mathematics, SO AN SSSR, Novosibirsk, 1988. Preprint 31 [In Russian].
8. S.I. Fadeev. A program for numerical solution of nonlinear boundary-value problems for systems of ordinary differential equations. Part II. Technical report, Institute of Mathematics, SO AN SSSR, Novosibirsk, 1988. Preprint 32 [In Russian].
9. S.I. Fadeev. A program for numerical solution of nonlinear boundary-value problems for systems of ordinary differential equations. Part III. Technical report, Institute of Mathematics, SO AN SSSR, Novosibirsk, 1988. Preprint 33 [In Russian].
10. Yu.A. Kuznetsov and V.V. Levitin. CONTENT: A multiplatform environment for analyzing dynamical systems. Dynamical Systems Laboratory, Centrum voor Wiskunde en Informatica, Amsterdam, 1996.
11. M. Lentini and V. Pereyra. An adaptive finite difference solver for nonlinear two point boundary problems with mild boundary layers. *SIAM J. Numer. Anal.*, 14:91–111, 1977.
12. V.V. Levitin. Computations of functions and their derivatives in CONTENT. Technical report, Centrum voor Wiskunde en Informatics, 1995. Report AM-R9512.
13. J. Lorenz. Nonlinear boundary value problems with turning points and properties of difference schemes. In W. Eckhaus and E.M. de Jager, editors, *Singular Perturbation Theory and Applications*, page Lecture Notes in Mathematics 942, New York, Berlin, Heidelberg, 1982. Springer-Verlag.
14. Yu.N. Rabotnov. *Mechanics of Deformable Solid Body.* Nauka, Moscow, 1979. [In Russian].
15. R.D. Russell and J. Christiansen. Adaptive mesh selection strategies for solving boundary value problems. *SIAM J. Numer. Anal.*, 15:59–80, 1978.
16. A.A. Samarskii and E.S. Nikolaev. *Numerical Methods for Grid Equations, Vol. I and II.* Birkhäuser Verlag, Basel, Boston, Berlin, 1989.