NONLINEAR BOUNDARY CONDITIONS
IN SOBOLEV SPACES

THESIS

Presented to the Graduate Council of the
North Texas State University in Partial
Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Walter Brown Richardson Jr., B.S., M.S.

Denton, Texas

December, 1984
Richardson, Walter B., **Nonlinear Boundary Conditions in Sobolev Spaces**. Doctor of Philosophy (Mathematics), December, 1984, 65 pp., 2 tables, bibliography, 28 titles.

The method of dual steepest descent is used to solve ordinary differential equations with nonlinear boundary conditions. A general boundary condition is $B(u) = 0$ where $B$ is a continuous functional on the $n$th order Sobolev space $H^1[0,1]$. If $F: H^1[0,1] \rightarrow L^2[0,1]$ represents a differential equation, define $\phi(u) = 1/2 \|F(u)\|^2$ and $\zeta(u) = 1/2 \|B(u)\|^2$. Steepest descent is applied to the functional $\xi = \phi + \zeta$. Two special cases are considered. If $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ is $C^2$, a Type I boundary condition is defined by $B(u) = f(u(0),u(1))$. Given $K:\{0,1\} \times \mathbb{R} \rightarrow \mathbb{R}$ and $g:\{0,1\} \rightarrow \mathbb{R}$ of bounded variation, a Type II boundary condition is $B(u) = \int_0^1 K(x,u(x)) \, dg(x)$.

The following results are proved. A close relationship is established between the two best known requirements for convergence of general steepest descent - Rosenbloom's strict convexity condition and Neuberger's gradient inequality. If $\beta$ is strictly convex, then Neuberger's inequality holds for $\tau = 1/2 \|\nabla \beta\|^2$. In dual steepest descent, if the gradient inequality holds for $\phi$ and $\zeta$, and the cosine of the angle between $\nabla \phi$ and $\nabla \zeta$ is bounded away from $-1$, then the gradient inequality holds for $\xi$. For a Type I boundary condition, the gradient inequality is equivalent to $\|\nabla \xi\|$ being bounded away from $0$. An explicit
form for \( v \xi(u) = B'(u)^* B(u) \) where \( B \) is a Type II boundary condition is given. As a corollary, the least constant for Sobolev's imbedding inequality, \( \|f\|_\infty \leq C_n \cdot \|f\|_{n,2} \), is shown to be

\[
C_n = \left\{ \frac{1}{n+1} \sum_{k=1}^{n} \frac{2 \sin^3(k\theta)}{\tanh(\sin(k\theta))} \right\}^{1/2}
\]

where \( \theta = \frac{\pi}{n+1} \).

Finally, dual steepest descent is implemented in a FORTRAN program, and used to solve the test problem \( y' = y \).

\( y(0) = y(1)^2 \).
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>iv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>II. DUAL STEEPEST DESCENT</td>
<td>16</td>
</tr>
<tr>
<td>III. TYPE II NONLINEAR BOUNDARY CONDITIONS</td>
<td>23</td>
</tr>
<tr>
<td>IV. NUMERICAL SOLUTIONS OF SECOND ORDER PROBLEMS WITH NONLINEAR BOUNDARY CONDITIONS</td>
<td>39</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>57</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>63</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table                                                                 Page

I. Best constant for the Sobolev Imbedding                              50
   Inequality on $H^1[0,1]$.                                             

II. Output from FH2.CGR                                                51
CHAPTER I

INTRODUCTION

Differential equations, both ordinary and partial, in which the solution is required to satisfy certain boundary conditions occur in such diverse areas as mechanics, acoustics, and electrical engineering. Taking second order ordinary boundary-value problems as an example, the linear case has been exhaustively studied over the past 200 years. Much work has been done over the past forty years in developing general methods of attacking the case in which the differential equation is nonlinear. However, almost without exception, the work has involved linear boundary conditions of the form

\[ A_1 y(a) + A_2 y'(a) = 0 \]
\[ B_1 y(b) + B_2 y'(b) = 0 \]

Very little is known about such problems when the boundary conditions are nonlinear.

Mawhin (6. 3) has proposed the following problem to describe nonlinear vibrations: find a \( y \) in \( H^2[0,1] \) such that

\[ y''(x) = \frac{2 y'(x)^2 y(x)}{1 + y(x)^2} \quad \text{for} \ x \in [0,1] \]

\[ y'(0)^2 - y(0)^2 = 0 \ \text{and} \ y(1) \cdot \{y(1) - y'(1)\} = 0 \]
An example of a boundary-value problem which arises in the study of electrical network theory is given by Morosanu (7). Given physical constants $L > 0$, $R \geq 0$, $C_1 > 0$, $G \geq 0$, $R_0 > 0$, and $C > 0$; a nonlinear resistance $f_0$ which is maximal monotone in $\mathbb{R} \times \mathbb{R}$; and a voltage $e$ per unit length impressed along the line in series with it; find the current $i = -u$ flowing in the line and the voltage $v$ across the line, so that

$$
\begin{align*}
\frac{\partial u}{\partial t} - \frac{\partial v}{\partial x} + R \cdot u + e(t, x) &= 0 \\
C_1 \frac{\partial v}{\partial t} - \frac{\partial u}{\partial x} + G \cdot v &= 0, \text{ for } 0 < x < 1 \text{ and } t > 0 \\
u(0, x) &= u_0(x); \quad v(0, x) = v_0(x); \quad 0 \leq x \leq 1 \\
-u(t, 1) &= C \frac{\partial v(t, 1)}{\partial t} + f_0(v(t, 1)), \quad t > 0.
\end{align*}
$$

Historically, in deriving governing differential equations from physical principles, nonlinearities were neglected because they were difficult to handle. One wonders if many boundary conditions should actually contain nonlinear terms in order to give a more accurate picture of the physics.

One seldom encounters a definition of boundary conditions - rather a group of example boundary-value problems is presented to give a feel for what such conditions are. Taking as our setting 2nd order problems, a linear boundary condition for the differential equation $F(u) = 0$, where $F : H^2[0, 1] \rightarrow L^2$, is defined to be one of the form $B(u) = 0$, where $B$ is a bounded linear functional on $H^2[0, 1]$. This includes standard textbook examples, interior
boundary conditions, and still more general ones of the form
\[ B(u) = \int_0^1 u \, dg. \]  
A nonlinear boundary condition is \[ B(u) = 0 \]
where \( B \) is a continuous functional on \( H^2(0,1) \), often assumed
to be \( C^2 \). For Mawhin's problem mentioned earlier,
\[ B(u) = \frac{1}{2} \left( u'(0)^2 - u(0)^2 \right)^2 + \frac{1}{2} \left( u(1) \cdot \left\{ u(1) - u'(1) \right\} \right)^2. \]
Perhaps the essence of the definition lies in the fact that
the range of \( B \) is a subset of the reals, rather than that
\( B(u) \) involves only the values of \( u \) at the topological
boundary of the domain of \( u \).

There are many methods for handling nonlinear problems: fixed-point theory, generalized inverse function theorems,
homotopy and degree theory, the Newton-Kantorovich method,
and the method of steepest descent. The last two are
particularly attractive because they yield not only valuable
theoretical information, but also numerical results when
implemented on a computer. Steepest descent dates back to
Cauchy (2), and although not used extensively on
minimization problems because of its slow convergence, the
method is very robust compared to Newton's method, which may
break down completely. Variations of steepest descent are
called general gradient methods and have found increasing
favor in numerical analysis. Recently, Neuberger (8, 9, 10,
11, 12) has used steepest descent in a Sobolev space setting
to solve differential equations with linear initial and
boundary conditions.
One of the chief appeals of steepest descent is its simplicity. Suppose each of $X$ and $Y$ is a Hilbert space and $F: X \to Y$ is $C^2$; to use steepest descent to solve the equation $F(x) = 0$, one defines $\Phi: X \to \mathbb{R}$ by $
abla \Phi(x) = 1/2 \| F(x) \|_Y^2$. A zero of $\Phi$ is a solution of the original equation and a minimum of $\Phi$ would be a "least squares" solution to the equation. In a differential equation setting $\Phi$ is a general variational principle applicable to many problems where there is no classical variational principle. At the point $x_0$ in $X$, the negative gradient of $\Phi$ points in the direction to move for the fastest instantaneous decrease of the function $\Phi$. The fundamental existence and uniqueness theorem for ordinary differential equations guarantees the existence of $z: [0, \infty) \to X$ such that $z(0) = x_0$ and $z'(t) = -\nabla \Phi(z(t))$. One tries to find conditions that insure that $\lim_{t \to \infty} z(t) = u$ exists and that $u$ is a local minimum of $\Phi$. Two such conditions are the convexity condition of P. C. Rosenbloom and the gradient inequality of Neuberger. A theorem in Chapter I shows that if a functional $\Phi$ satisfies Rosenbloom's condition then $\gamma(x) = 1/2 \| \nabla \Phi(x) \|_Y^2$ satisfies Neuberger's gradient inequality.

A detailed description of using steepest descent with the Sobolev inner product to solve the test problem $y' = y$, $y(0) = 1$ on $[0,1]$ is given by Neuberger in (11). Briefly, letting $X = \{ y \in H^1[0,1]: y(0) = 0 \}$. $Y = L^2[0,1]$, and
defining $F : X \rightarrow Y$ by $F(x) = (x+1)' - (x+1)$, one has that if $z$ is a zero of $F$, then $y = z + 1$ is a solution to the original problem. A discrete steepest descent is implemented to find a numerical solution and, most importantly, the rate of convergence of the finite difference scheme is shown to be far superior to that of the same process, using the Euclidean inner product.

Suppose the problem is complicated by introducing the simplest nonlinear boundary conditions: $y' = y$, $y(0) = y(1)^2$. There are several ways one might attack the new problem. Define $F : H^1[0,1] \rightarrow L^2[0,1]$ and $B : H^1[0,1] \rightarrow \mathbb{R}$ by $F(x) = x' - x$ and $B(x) = x(0) - x(1)^2$. Neuberger in (8) uses a scheme which is based on steepest descent for the functional $\phi(x) = \frac{1}{2} \| F(x) \|^2$ and employs the family of projection operators $\langle P_x \rangle$ in $H^1[0,1]$ where $P_x$ is the orthogonal projection onto the null space of $B'(x)$. This approach is closely related to the gradient-projection method used in nonlinear optimization (14, 15, 19). It has the property that, both in the continuous and discrete steepest descent processes, one always stays in the set of all functions which satisfy the boundary conditions.

We shall use another tack and attempt to solve the problem by applying steepest descent simultaneously to the differential equation and to the boundary condition part of the problem. The name penalty function is given to this method when applied to the constrained optimization of a
real-valued function. However, the idea does not seem to have found its way into the area of differential equations with nonlinear boundary conditions. Here convergence criteria would be harder to come by, because of the infinite dimensional setting and resulting loss of local compactness.

Given $F: X \rightarrow Y$ and $B: X \rightarrow R$, both $C^2$, we seek an $x$ in $X$ such that $F(x) = 0$ and $B(x) = 0$. Define $\Phi: X \rightarrow R$ by $\Phi(x) = \frac{1}{2} \| F(x) \|^2$ and $\zeta: X \rightarrow R$ by $\zeta(x) = \frac{1}{2} \| B(x) \|^2$. These are measures of how close an element of $X$ comes to satisfying the differential equation and the boundary condition, respectively. If $G: X \rightarrow XXR$ is given by $G(x) = \begin{pmatrix} F(x) \\ B(x) \end{pmatrix}$ and

$$\xi(x) = \frac{1}{2} \| G(x) \|^2_{XXR} = \frac{1}{2} \| F(x) \|^2 + \frac{1}{2} \| B(x) \|^2 = \Phi(x) + \zeta(x),$$

then steepest descent can be used. $x$ is a solution to the pair of equations only in case it is a zero of the functional $\xi$.

A few comments about this proposed method are in order. First, although it will work on a problem with linear boundary conditions, it clearly will not be as efficient as steepest descent applied to the subspace of all functions satisfying the boundary conditions or as classical algorithms designed to solve a specific class of problems. The method will come into its own when the boundary conditions are nonlinear. While not depreciating the importance of, say, fast Poisson-solvers, there is a great
need for a widely applicable, type-independent method which handles general boundary conditions and in which there is a close relationship between theoretical and computational aspects.

Second, when the boundary conditions of the above example are written as \( y(0) = y(1)^2 \) it is easy to find an initial guess that satisfies them and then begin the projection method — never leaving the set \( K \) of all functions \( f \) such that \( f(0) = f(1)^2 \). However, if the problem were posed in an \( H^2[0,1] \) setting, the boundary conditions might have been written equivalently as \( y(0) = y'(1)^2 \) and it is no longer as easy to find an initial guess. Also, one stays in a different set \( \overline{K} \) of functions. Lastly, in partial differential equations one often encounters a situation in which it is not known a priori from physical or theoretical considerations what a "natural" set of boundary conditions is. In this case, repeated trials of the dual steepest descent using different choices for the functional \( B \) should yield valuable information about what boundary conditions are feasible.

This paper investigates two specific forms of nonlinear boundary conditions. The natural setting for a first order problem is \( H^1 \); suppose \( f: \mathbb{R}^2 \rightarrow \mathbb{R} \) is a given \( C^2 \) function and define \( B:H^1[0,1] \rightarrow \mathbb{R} \) by \( B(u) = f(u(0), u(1)) \). This is the most general nonlinear boundary condition which involves
only $u(0)$ and $u(1)$ and is still manageable. A sufficient condition for the gradient inequality to hold for this boundary condition is given in Chapter II. The second type of condition uses the values of $u$ everywhere in $[0,1]$, although the nonlinearities are milder. Given $g:[0,1] \to \mathbb{R}$ and $K:[0,1] \times \mathbb{R} \to \mathbb{R}$, define $B:H^1[0,1] \to \mathbb{R}$ by $B(u) = \int_0^1 K(x,u(x)) \, dg(x)$. This includes conditions such as $u(0) = u(1)^2$ and, in fact, any problem in which the nonlinearities occur pointwise, but encounters difficulties with $u(0) \cdot u(1) + u(1/2)^2 = 0$, because of the first term.

An interesting spinoff of the above work relates to one of the large class of inequalities in analysis which go under the name of Sobolev. These state that the norm of a function in one space is dominated by a constant times the norm of its derivative in a second space. The existence of such constants dates back to Sobolev (17) and is critical to the development of theoretical partial differential equations (1). Within the last ten years several articles have been written which address the question of what is the best or least constant that can be used in a particular Sobolev inequality. Talenti (18), Rosen (13), and Lieb (4) seek the least $c$ such that $\|u\|_q \leq c \cdot \|u\|_p$ where $1 < p, q < \infty$ and $\text{domain}(u) = \mathbb{R}^n$. Using variational methods, Marti (5) found the best $c$ for the inequality $\|u\|_\infty \leq c \cdot \|u\|_{1,2}$ and gave numerical results for the higher order Sobolev spaces.
\[ H^n[0,1], \text{ where } n = 2, \ldots, 6. \] As a Corollary to a Theorem in Chapter III we show that the least \( c \) such that for all \( u \) in \( H^n[0,1] \), \( \|u\|_\infty \leq c \cdot \|u\|_n \), is given by
\[
c_n = \left\{ \frac{1}{n+1} \sum_{k=1}^{n} \frac{2 \sin^3(k\theta)}{\tanh(\sin(k\theta))} \right\}^{1/2}\] where \( \theta = \frac{\pi}{n+1} \), and require none of the machinery of the calculus of variations.

Chapter IV describes how the dual steepest descent can be implemented on a computer. Given \( F: \mathbb{R}^4 \rightarrow \mathbb{R} \) and \( B: \mathbb{R}^4 \rightarrow \mathbb{R} \), we seek a member \( u \) of \( H^2[0,1] \) such that for \( t \) in \([a,b]\),
\[ F(t,u(t),u'(t),u''(t)) = 0 \text{ and } B(u(a),u(b),u'(a),u'(b)) = 0. \]
A finite-difference scheme is used to reduce the problem to solving a set of nonlinear equations by steepest descent. Much improved convergence over traditional steepest descent is obtained by employing the Sobolev rather than the Euclidean inner product and by using a nonlinear version of Hestenes' conjugate-gradient algorithm. Finally, the code of a FORTRAN program, \( FH2.CGR \), which uses the above ideas is discussed, as is the output from the test problem \( y' = y, \]
\[ y(0) = y(1)^2. \]


CHAPTER II

DUAL STEEPEST DESCENT

From a theoretical standpoint the most difficult aspect of steepest descent is finding conditions for convergence.

In an excellent survey article, P. C. Rosenbloom (6, p. 128) establishes the following fact. If \( X \) is a Hilbert space, \( \phi: X \rightarrow \mathbb{R} \) is \( C^2 \), \( z(0) = x_0 \) and \( z'(t) = -\nabla \phi(z(t)) \) for \( t \geq 0 \), and there is an \( A > 0 \) such that for all \( x, y \) in \( X \),

\[
\langle H(x)y, y \rangle \geq A \cdot \|y\|^2
\]

where \( H \) denotes the Hessian of \( \phi \), then

\[
\lim_{t \to \infty} z(t) \equiv u \text{ exits and is the unique minimum of } \phi. \text{ If } \phi
\]
arises from linear least squares, i.e., \( \phi(x) = \frac{1}{2} \|F(x)\|^2 \)
for some positive self-adjoint \( F \), then \( \phi \) will satisfy

Rosenbloom's condition. Neuberger (2, 3, 4) gives another sufficient condition for convergence of a variational \( \phi \). If \( \Omega \) is an open subset of \( X \), \( z(0) = x_0 \) and \( z(t) = -\nabla \phi(z(t)) \), \n
\( \text{Range}(z) \subseteq \Omega \), and there exists a \( C > 0 \) such that for all \( x \)
in \( X \), \( \|\nabla \phi(x)\| \geq C \cdot \|F(x)\| \), then the \( \lim_{t \to \infty} z(t) \equiv u \) exists and

\[
F(u) = 0. \text{ The following theorem connects the two conditions.}
\]

Theorem 1 If \( \phi: X \rightarrow \mathbb{R} \) is \( C^2 \), \( \alpha \) is a positive number so that for all \( y \) in \( H \), \( \langle H_\phi(x)y, y \rangle \geq \alpha \|y\|^2 \) and for \( x \) in \( X \),

\[
\gamma(x) = \frac{1}{2} \|\nabla \phi\|^2,
\]
then \( \gamma \) satisfies Neuberger's gradient
inequality, i.e., there is a positive \( C \) so that \( \| \nabla \phi(x) \| \leq C \cdot \| \nabla \psi(x) \| \) for all \( x \) in \( X \).

**Proof:** We use primes to denote Frechet derivatives and \( H_\phi(x) \) to denote the Hessian of \( \phi \) at \( x \). Let each of \( x, h, \) and \( k \) be in \( X \).

\[
\langle (\nabla \phi)'(x)h, k \rangle = \langle (\nabla \phi)(x+h), k \rangle - \langle (\nabla \phi)(x), k \rangle \\
= \phi'(x+h)k - \phi'(x)k \\
= \phi''(x)(k, h) \\
= \langle H_\phi(x)k, h \rangle.
\]

So, \( \psi'(x)h = \langle (\nabla \phi)'(x)h, \nabla \psi(x) \rangle = \langle H_\phi(x)(\nabla \psi(x)), h \rangle \) and \( \psi \psi(x) = H_\phi(x)(\nabla \psi(x)) \). Now \( \langle H_\phi(x)y, y \rangle \geq \alpha \| y \|^2 \) just says \( H_\phi(x) \geq \alpha \cdot I \) which implies \( H_\phi(x)^2 \geq \alpha^2 \cdot I \).\n
\[
\| \nabla \psi \|^2 = \langle H_\phi(x)(\nabla \psi(x)), H_\phi(x)(\nabla \psi(x)) \rangle \\
\geq \alpha^2 \cdot \| \nabla \psi(x) \|^2 \\
= 2 \cdot \alpha^2 \cdot \gamma(x).
\]

Taking square roots we have the desired inequality.

Both Rosenbloom's and Neuberger's conditions for convergence of the steepest descent process are far from necessary. Work remains to be done to find large classes of problems for which they would give existence results not already obtained by other methods. Also, their connection with the well known Palais-Smale (S.1, p. 353) property should be investigated.

In the dual steepest descent process, it is natural to ask if the gradient inequality applies to both of the
functions $\phi$ and $\zeta$, will it apply to $\xi$? Not without some additional condition, for example, that the angle between the vectors $v\phi(x)$ and $v\zeta(x)$ be bounded away from $\pi$.

**Theorem II** If $\Omega$ is an open subset of $X$, $x \in \Omega$, and

1. $\text{Range}(z) \subset \Omega$ where $z_0 = x$, $z'(t) = -v\phi(z(t))$
2. there is a $c > 0$ and a $d > 0$ such that for all $y$ in $\Omega$,
   $$\|v\psi(y)\| \geq c \cdot \|F(y)\| \quad \text{and} \quad \|v\zeta(y)\| \geq d \cdot \|B(y)\|$$
3. there is an $\alpha$ in $(-1,1)$ such that for all $y$ in $\Omega$
   $$\langle v\psi(y), v\zeta(y) \rangle \geq \alpha \frac{\|v\psi(y)\| \cdot \|v\zeta(y)\|}{\|v\psi(y)\| + \|v\zeta(y)\|}$$

Then $\lim_{t \to \infty} z(t) \equiv u$ exists and $F(u) = 0 = B(u)$.

**Proof:** Clearly, we may assume $\alpha \in (-1,0)$.

$$\|v\zeta(y)\|^2 = \|v\psi(y) + v\zeta(y)\|^2 = \|v\psi(y)\|^2 + 2 \cdot \langle v\psi(y), v\zeta(y) \rangle + \|v\zeta(y)\|^2 \geq \|v\psi(y)\|^2 + 2 \cdot \alpha \cdot \|v\psi(y)\| \cdot \|v\zeta(y)\| + \|v\zeta(y)\|^2$$

$$= -\alpha \left\{ \|v\psi(y)\|^2 - 2 \cdot \|v\psi(y)\| \cdot \|v\zeta(y)\| + \|v\zeta(y)\|^2 \right\} + \left(1+\alpha\right) \cdot \left\{ \|v\psi(y)\|^2 + \|v\zeta(y)\|^2 \right\} \geq \left(1+\alpha\right) \cdot \min\{c^2, d^2\} \cdot \left\{ \|F(y)\|^2 + \|B(y)\|^2 \right\}$$

Letting $\kappa = \sqrt{1+\alpha} \min\{c,d\}$, we have a constant $\kappa$ such that for all $y$ in $\Omega$, $\|v\zeta(y)\| \geq \kappa \cdot \|F(y)\|$ and Neuberger's theorem assures that $\lim_{t \to \infty} z(t) \equiv u$ exists and satisfies $G(u) = 0$.

For first order problems nonlinear boundary conditions which involve only the values of the unknown function at the
endpoints of the interval can be put in the following general framework. Given \( f: \mathbb{R}^2 \rightarrow \mathbb{R}, \ C^{(2)} \); define a Type I boundary condition on \( H^0[0,1] \) to be a functional \( B(y) = f(y(0), y(1)) \). \( f \) is said to generate \( B \). To investigate convergence of the boundary part of the steepest descent process we need to compute \( \nabla \zeta(x) = B'(x)^*B(x) \). A simple argument shows

\[
B'(x)y = f_1(x(0), x(1)) \cdot y(0) + f_2(x(0), x(1)) \cdot y(1),
\]
and to compute the adjoint, we seek \((g, h)\) in \( \mathbb{R} \times H^1 \) such that for all \( y \) in \( H^1 \),

\[
< y, h >_{H^1} = < B'(x)y, g >_{\mathbb{R}}, \text{ i.e., so that}
\]

\[
\int_0^1 y \cdot h + y' \cdot h' = < f_1(P_x) \cdot y(0) + f_2(P_x) \cdot y(1) > g
\]

where \( P_x = (x(0), x(1)) \). Supposing for a moment that \( h \) has a second derivative we can use integration by parts to write the above as

\[
0 = \int_0^1 y(h-h') + y(1)(h'(1) - f_2(P_x) \cdot g) - y(0)(h'(0) + f_1(P_x) \cdot g)
\]

Which leads to the boundary value problem: find \( h \) in \( H^1[0,1] \) so that

\[
h'' = h
\]

\[
h'(1) = f_2(P_x) \cdot g \quad \text{and} \quad h'(0) = -f_1(P_x) \cdot g.
\]

The solution is

\[
h(t) = \frac{g}{\sinh(1)} \left\{ f_1(P_x) \cdot \cosh(1-t) + f_2(P_x) \cdot \cosh(t) \right\},
\]

and the following theorem can now be proved.
Theorem III. Let \( B \) be a type I boundary condition on \( H^1[0,1] \) with generating function \( f: \mathbb{R}^2 \to \mathbb{R}, f \in C(2) \) and not identically 0. The following statements are equivalent:

1. There is a \( d > 0 \) such that for \( (a,b) \) in \( \mathbb{R}^2 \),
   \[ \|\varphi f(a,b)\| \geq d. \]

2. There is a \( c > 0 \) such that for all \( x \) in \( H^1[0,1] \),
   \[ \|\varphi \zeta(x)\| \geq c \cdot \|B(x)\|. \]

Proof: \[ \|\varphi \zeta(x)\|^2 = \|B'(x)^*B(x)\|^2 \]

\[ = |B(x)|^2 \cdot \left\{ \frac{1}{\sinh(1)} \left\{ f_1(P_x) \sinh(1) + f_2(P_x) \right\} + \right. \]

\[ \left. \frac{1}{\sinh(1)} \left\{ f_2(P_x) \sinh(1) \left\{ f_1(P_x) + f_2(P_x) \sinh(1) \right\} \right\} \right\}

\[ = \frac{|B(x)|^2}{\sinh(1)} \left\langle A_{\varphi f(P_x)}, \varphi f(P_x) \right\rangle \]

where \( A = \begin{bmatrix} \cosh(1) & 1 \\ 1 & \cosh(1) \end{bmatrix} \) is a positive definite matrix with lower bound \( m = \cosh(1) - 1 \) and upper bound \( M = \cosh(1) + 1 \).

(1) \( \rightarrow \) (2) Let \( x \in H^1 \).

\[ \|B'(x)^*B(x)\|^2 = \frac{1}{\sinh(1)} \left\langle A_{\varphi f(P_x)}, \varphi f(P_x) \right\rangle \|B(x)\|^2 \]

\[ \geq \frac{m}{\sinh(1)} \|\varphi f(P_x)\|^2 \geq \frac{m}{\sinh(1)} d^2 \|B(x)\|^2 \]

and there exists a global \( c \) for \( B \).

(2) \( \rightarrow \) (1) Let \( G = \{(a,b) \in \mathbb{R}^2 : f(a,b) \neq 0 \} \) and \( (a,b) \in G \). For \( t \) in \([0,1] \), define \( x(t) = (1-t)a + tb \). Now \( x \in H^1[0,1] \), and \( \|B(x)\|^2 \frac{M}{\sinh(1)} \|\varphi f(P_x)\|^2 \geq \|\varphi \zeta(x)\|^2 \geq c^2 \cdot |B(x)|^2 \).
Since $B(x) \neq 0$, $\|vf(p_x)\|^2 \geq \frac{c^2 \sinh(1)}{M}$ and letting \(d\) be the square root of the right hand quantity, we have that for all \((a,b)\) in \(G\), $\|vf(a,b)\| \geq d$. To show that the inequality must hold on all of \(\mathbb{R}^2\), let \(S\) be the interior of the complement of \(G\) and suppose that \(S\) is nonempty. Since \(\mathbb{R}^2\) is connected, there is a limit point \(p\) of \(S\) in the complement of \(S\). Let \(\{q_k\}\) be a sequence of points of \(S\) and \(\{r_k\}\) be a sequence of points of \(G\), both of which converge to \(p\). Since \(q_k\) belongs to the open set \(S\) on which \(f\) is zero, \(vf(q_k) = 0\), and since the gradient is continuous, \(vf(p) = 0\). However, \(r_k\) in \(G\) implies $\|vf(r_k)\| \geq d$ and therefore $\|vf(p)\| \geq d > 0$. The contradiction shows that \(G\) is dense in \(\mathbb{R}^2\).


CHAPTER III

TYPE II NONLINEAR BOUNDARY CONDITIONS

An important class of boundary-value problems is the one in which the differential equation is ordinary of order $n$ and the boundary conditions involve the unknown function but none of its derivatives. Even this class is so large that we restrict ourselves to the following subset, where without loss of generality all work is done on the interval $[0,1]$. Let $n$ be a positive integer, $K:[0,1] \times \mathbb{R} \to \mathbb{R}$ be continuous and such that $K^2$ is continuous, and $g:[0,1] \to \mathbb{R}$ be a function of bounded variation. A Type II boundary condition with generating kernel $K$ is $B(u) = 0$ where $B:H^n[0,1] \to \mathbb{R}$ is given by

$$(*) \quad B(u) = \int_0^1 K(x,u(x)) \, dg(x).$$

If $K$ is defined by $K(a,b) = (1-a)b - ab^2$, and $g$ by

$$g(x) = \begin{cases} 0 & x = 0 \\ 1 & 0 < x < 1 \\ 2 & x = 1 \end{cases},$$

then the boundary condition $y(0) = y(1)^2$ of our sample problem can be put into this framework. Once again, to investigate steepest descent for the functional $\zeta(u) = 1/2 \|B(u)\|^2$, one seeks an explicit form for $\nabla \zeta(u) = B'(u)B(u)$.

Theorem IV If $y \in H^n[0,1]$ and $B: H^n[0,1] \to \mathbb{R}$ is defined by
(*), then

(a) \[ B'(y)f = \int_0^1 K_2(t, y(t)) \cdot f(t) \, dg(t) \]

(b) \[ (B'(y)x) = \int_0^1 \frac{1}{n+1} \sum_{k=1}^n \frac{\sin(k\theta)}{\sinh(\sin(k\theta))} \int_0^1 K_2(x, y(x)) \left\{ \begin{array}{l}
\cosh((1-x-t)\sin(k\theta)) \cdot \cos((t-x)\cos(k\theta)) \\
- \cos(2k\theta) \cdot \cosh((1+t-x)\sin(k\theta)) \cdot \cos((t-x)\cos(k\theta)) \\
- \sin(2k\theta) \cdot \sinh((1+t-x)\sin(k\theta)) \cdot \sin((t-x)\cos(k\theta)) \end{array} \right\} \, dg(x) \]

Proof: (a) This follows directly from the definition of Frechet derivative.

(b) We first prove the theorem in the special case that both \( K \) and \( g \) are \( C^{(2n)} \) and that for \( k=1, \ldots, 2n \), both \( g^{(k)}(0) \) and \( g^{(k)}(1) \) are 0. Using the definition of adjoint (4, p. 309), we seek a member \( h \) of \( H^n[0,1] \) such that for every \( f \) in \( H^n[0,1] \), \( \langle B'(y)f, 1 \rangle_R = \langle f, h \rangle_{H^n} \), i.e., so that

\[ \int_0^1 K_2(t, y(t)) f(t) \, dg = \sum_{k=0}^n \int_0^1 f^{(k)} h^{(k)} \, dt. \]

Supposing that \( h \) has 2n derivatives (Later this will be verified), integration by parts can be used to obtain

\[ \int_0^1 f K_2(t, y(t)) g' = \int_0^1 f \cdot n \, dt + \sum_{k=1}^n \left\{ (-1)^k \int_0^1 f \cdot h^{(2k)} \, dt + \right\} \sum_{j=0}^{k-1} \left\{ (-1)^{k-1-j} f^{(j)} \cdot h^{(2k-1-j)} \right\} \left[ \begin{array}{c}
\int_0^1 f \cdot \\
\sum_{k=0}^n (-1)^k h^{(2k)} \end{array} \right] \left[ \begin{array}{c}
\int_0^1 f \cdot \\
\sum_{k=0}^n h^{(2k)} \end{array} \right] \int_0^1 . \]
Using Fubini's Theorem for finite sums, the double sum can be rewritten as
\[ \sum_{j=0}^{n-1} \sum_{k=j+1}^{n} (-1)^{k-j} f(j) \cdot h(2k-j) \bigg|_0^1 \]
and finally as,
\[ \sum_{j=0}^{n-1} \sum_{m=j}^{n-1} (-1)^{m-j} h(2m-j+1) \bigg|_0^1 \]
By a standard argument, for the above equation to hold for every \( f \) in \( H^n[0,1] \), \( h \) must satisfy the boundary-value problem:
\[
\begin{align*}
(1) & \quad \sum_{k=0}^{n} (-1)^k h(2k) = \int_{[1,y]} g'' \\
(2) & \quad \sum_{m=j}^{n-1} (-1)^{m-j} h(2m-j+1)(0) = 0, \quad j = 0, \ldots, n-1 \\
& \quad \sum_{m=j}^{n-1} (-1)^{m-j} h(2m-j+1)(1) = 0
\end{align*}
\]
For convenience, let \( \nu \) denote \( \int_{[1,y]} g'' \). The characteristic equation, \( \sum_{k=0}^{n} (-1)^k \omega^{2k} = 0 \), can be rewritten as
\[ 0 = \sum_{k=0}^{n} (-\omega^2)^k = \frac{1 - (-\omega^2)^{n+1}}{1 - (-\omega^2)} \]
Let \( S \) be the set of solutions to \( (-\omega^2)^{n+1} = 1 \), excluding \( \pm 1 \). Let \( \theta = \frac{\pi}{n+1} \) and \( \alpha = e^{\theta i} \). An enumeration of the members of \( S \) is given by \( \langle \Lambda_k \rangle_{k=1}^{2n} \) where
\[ \Lambda_k = \begin{cases} 
\alpha^k & \text{for } 1 \leq k \leq n \\
\alpha^{-(k-n)} & \text{for } n+1 \leq k \leq 2n
\end{cases} \]
\[ z(t) = \sum_{k=1}^{2n} c_k \alpha^k t \] is the general solution of the homogeneous problem. Variation of parameters yields a
particular solution \( z_p(t) = \sum_{k=1}^{2n} u_k(t) e^{\lambda_k t} \). The differential equation in standard form is
\[
\sum_{k=0}^{n} (-1)^{n-k} h(2k) = (-1)^n v
\]
and so denoting the \( m \)th order Vandermonde determinant by 
\[\text{Vand}(\beta_j)_{j=1}^m,\]

\[
u_k(t) = (-1)^n v(t) (-1)^{2n+k} e^{\sum_{j\neq k}^{\lambda_k t} \text{Vand}(\lambda_j)_{j=1}^{2n}}
\]

\[
h(t) = \sum_{k=1}^{2n} c_k e^{\lambda_k t} + \sum_{k=1}^{2n} -\beta_k \int_0^t e^{\lambda_k(t-x)} v(x) dx
\]

where \( \beta_k = (-1)^n \frac{m!}{\prod (\lambda_m - \lambda_k)} \). Now, using the fact that \( K \) and \( g \)
are \( C^{2n} \), we have that for \( 1 \leq m \leq 2n \),
\[
h(m)(t) = \sum_{k=1}^{2n} c_k e^{\lambda_k t} + \sum_{k=1}^{2n} -\beta_k \left\{ \int_0^{m} e^{\lambda_k(t-x)} v(x) dx + \sum_{j=1}^{m} \lambda_k^j v(m-j)(t) \right\}
\]

and \( h \) has the \( 2n \) derivatives supposed earlier.

The boundary conditions specify that for \( q = 0, \ldots, n-1 \),
\[
r_q(t) = \sum_{p=q}^{n-1} (-1)^{p-q} h(2p-q+1)(t)
\]

\[
= \sum_{p=q}^{n-1} (-1)^{p-q} \left\{ \sum_{k=1}^{2n} c_k e^{\lambda_k t} + \sum_{k=1}^{2n} -\beta_k \left\{ \int_0^{2p-q+1} e^{\lambda_k(t-x)} v(x) dx + \sum_{j=0}^{2p-q} \lambda_k^j v(2p-q-j)(t) \right\} \right\}
\]

should be zero when evaluated at 0 and 1. Using the fact
that \( v^{(k)}(0) = 0 = v^{(k)}(1) \) for \( k = 0, \ldots, 2n-1 \). The above simplifies to

\[
\rho_q(t) = \sum_{k=1}^{2n} e^{\lambda_k t} \left\{ \sum_{p=q}^{n-1} (-1)^{p-q} \lambda_k^{2p-q+1} \right\} c_k + \\
\sum_{k=1}^{2n} -\beta_k e^{\lambda_k t} \int_0^t e^{-\lambda_k x} v(x) dx \left\{ \sum_{p=q}^{n-1} (-1)^{p-q} \lambda_k^{2p-q+1} \right\}
\]

and the boundary conditions can be written as \( A\mathbf{c} = \mathbf{b} \) where \( A \) is the \( 2n \times 2n \) matrix defined by

\[
A_{jk} = \begin{cases} 
\sum_{p=j}^{n} (-1)^{j-p} \lambda_k^{2p-j} & 1 \leq j \leq n \\
\lambda_k & 1 \leq k \leq 2n \\
\sum_{j-n,k}^{j-1} e^{\lambda_k x} v(x) dx & n+1 \leq j \leq 2n
\end{cases}
\]

and \( \mathbf{b} \) is the \( 2n \) vector given by

\[
(\mathbf{b})_j = \begin{cases} 
0 & 1 \leq j \leq n \\
\sum_{k=1}^{2n} \beta_k A_{jk} \int_0^t e^{-\lambda_k x} v(x) dx & n+1 \leq j \leq 2n
\end{cases}
\]

It is shown in Lemma 1 that \( A \) is non-singular, so

\[
h(t) = \sum_{j=1}^{2n} (A^{-1} \mathbf{b})_j e^{\lambda_j t} + \sum_{j=1}^{2n} -\beta_k \int_0^t e^{-\lambda_k x} v(x) dx
\]

Further investigation of the matrix \( A \) and the sequence \( (\lambda_k)_{k=1}^{2n} \) will give a more explicit form for \( h \).

Define \( M \) to be the \( 2n \times 2n \) matrix in which the nonzero entries are
\[
M_{jk} = \begin{cases}
\frac{\lambda_k}{e_k - e_k + n} & \text{if } j = k \\
\frac{1 + \lambda_k^2}{e_k - e_k + n} & \text{if } j = k + n \\
\frac{1 + \lambda_k}{e_k - e_k - n} & \text{if } j = k - n \\
\frac{1 + \lambda_k}{e_k - e_k - n} & \text{if } n + 1 \leq k \leq 2n
\end{cases}
\]

Examining the product \(A \cdot M\), one finds that

\[
A \cdot M = \begin{bmatrix}
0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0 \\
A_{n+1,1} & \cdots & A_{n+1,2n} \\
A_{2n,1} & \cdots & A_{2n,2n}
\end{bmatrix}
\]

and then applying \(A^{-1}\) to both sides.

\[
A^{-1} (0, \ldots, 0, A_{n+1,m}, \ldots, A_{2n,m})^T = \text{mth column of } M.
\]

\[
\mathcal{E}_{j=1}^{2n} (A^{-1} b) \cdot e_j^t = \sum_{j=1}^{2n} \lambda_j^t 
\]

\[
= \sum_{j=1}^{2n} \lambda_j^t \left\{ \sum_{k=1}^{2n} \beta_k \int_0^1 e^{-A_k x} v(x) dx \cdot M_{jk} \right\}
\]

\[
= \sum_{k=1}^{2n} \left\{ \beta_k \int_0^1 e^{-A_k x} v(x) dx \cdot \sum_{j=1}^{2n} \lambda_j^t M_{jk} \right\}
\]
From Lemma II, we have that for $1 \leq k \leq n$, $\beta_k = -\frac{\sin(k\theta)}{n+1} \frac{\lambda_k}{\lambda_k^2}$ and $\beta_{n+k} = \frac{\sin(k\theta)}{n+1} \frac{\lambda_{n+k}}{\lambda_{n+k}^2}$.

Also the expressions for the nonzero entries of the matrix can be simplified, for example, if $1 \leq k \leq n$.

$$M_{n+k,k} = \frac{\lambda_k}{1 - \lambda_k} \frac{1 + \lambda_{n+k}^2}{1 + \lambda_k^2}$$

After such simplification we have

$$\sin(k\theta) \frac{1}{n+1} \sum_{k=1}^{n} \frac{\sin(k\theta)}{\sinh(\sin(k\theta))} \int_0^1 v(x) \, dx.$$
After simplifying the terms $1/2e^{-k_0}e^{-k_0}$ and $1/2e^{-k_0}e^{-k_0}$ and working in similar fashion with $\sum_{k=1}^{n} -e^{-k_0}x+\sin(k_0)$, one has the following expression for $h(t)$:

$$h(t) = \frac{1}{n+1} \sum_{k=1}^{n} \frac{\sin(k_0)}{\sinh(\sin(k_0))} \int_{0}^{t} v(x) \, dx$$

$$+ i \left\{ \cosh((1-x-t)\sin(k_0)) \cdot \cos((t-x)\cos(k_0)) - \cos(2k_0) \cdot \cosh((1+t-x)\sin(k_0)) \cdot \cos((t-x)\cos(k_0)) - \sin(2k_0) \cdot \sinh((1+t-x)\sin(k_0)) \cdot \sin((t-x)\cos(k_0)) \right\} \, dx$$

$$+ 2 \frac{1}{n+1} \sum_{k=1}^{n} \sin(k_0) \int_{0}^{t} v(x) \, dx$$

$$+ i \left\{ \cos(2k_0) \cdot \sinh((t-x)\sin(k_0)) \cdot \cos((t-x)\cos(k_0)) + \sin(2k_0) \cdot \cosh((t-x)\sin(k_0)) \cdot \sin((t-x)\cos(k_0)) \right\} \, dx$$

Since $h(t)$ is real, the multiples of $i$ must sum to zero and we are left with the desired expression (recall that $v(x) = K_2(x,y(x)) \cdot g'(x)$). The Theorem is proved in the case that
Surprisingly, the general case follows from the special one. Let K: [0,1] x R → R be continuous and such that K_2 is continuous, g: [0,1] → R be increasing, and h = B'(y) \cdot 1. There exists a sequence (K_j)_{j=1}^{\infty} of functions such that (K_j) is C^2, K_j → K and (K_j)_{j=1}^{\infty} uniformly on [0,1] x [-\gamma, \gamma] where \gamma = \sup\{|y(t)|: t \in [0,1]\). Let (g_j)_{j=1}^{\infty} be a sequence such that g_j is C^2, g_j^{(k)} = 0 = g_j^{(k)} for k=1, \ldots, 2n, and g_j → g pointwise. With each pair (K_j, g_j) there is an associated h_j = B_j'(y) \cdot 1 which has been shown to have the desired form. Now, (K_j)_{j=1}^{\infty} uniformly, implies for fixed w and r,

\[ \int_0^1 (K_j) (x, y(x)) \cdot w(x) dx \rightarrow \int_0^1 K_2(x, y(x)) \cdot w(x) dx. \]

Also, since each g_j is nondecreasing and g_j → g pointwise, (4, p. 121) for fixed w,

\[ \int_0^1 w(x) \cdot g_j'(x) dx = \int_0^1 w(x) \cdot dg_j(x) \rightarrow \int_0^1 w(x) \cdot dg(x). \]

This shows that (h_j)_{j=1}^{\infty} must converge pointwise to a function of the desired form. Now, (K_j)_{j=1}^{\infty} uniformly implies B_j'(y) → B'(y) in norm, and using the continuity of the adjoint operation, h \equiv B'(y) \cdot 1 = \lim_{n \to \infty} B_j'(y) \cdot 1 = \lim_{n \to \infty} h_j.

The case where g is of bounded variation follows immediately from the case in which g is increasing.

Lemma 1 The matrix A defined in Theorem IV is nonsingular and if \text{Vand}(d_k)_{k=1}^{n} denotes the nth order Vandermonde determinant, then
\[ \det A = i^{n(n-1)} \left\{ \text{Vand}(2 \cos(k\theta)) \right\}_{k=1}^{n} \frac{2 \pi}{n} \left( e^{i\alpha} - e^{-i\alpha} \right) \].

**Proof:** Let \( w \) be a complex number such that \((-w^2)^{n+1} = 1\).

\[ w = \pm i. \quad \sum_{m=j}^{n} (-1)^{m-j} w^{2m-j} = w^j \sum_{m=j}^{n} (-1)^{m-j} w^{2(m-j)} = \]

\[ w^j \frac{1-(-w^2)^{n-j+1}}{1+w^2} = \frac{w^j (-(-w^2)^{n+1} - (-1)^{j} w^{-j})}{1+w^2}. \]

So, \( \det A = \frac{2n}{\pi} \sum_{k=1}^{n} \frac{1}{1+\lambda_k^2} \det \begin{bmatrix} v_k & \ldots & k = 1, \ldots, 2n \end{bmatrix} \]

where \( v_k \) is the \( n \times 1 \) vector whose \( j \)th component, \((v_k)_j\), is \( \lambda_k^j - (-1)^j \lambda_k^{-j} \). For \( 1 \leq m, j \leq n \),

\[ (v_{n+m})_j = (i\alpha^{-m})_j - (-1)^j (i\alpha^{-m})^{-j} = i^{j} \alpha^{-mj} - (-1)^j i^{j} \alpha^{mj} = (-1)^j (i\alpha^{-m})^{-j} - (i\alpha^m)_j = -(v_m)_j \]

so adding column \( m \) to column \( n+m \), \( \det A \) is

\[ \frac{1}{2n \pi} \det \begin{bmatrix} v_k & \ldots & 0 & k = 1, \ldots, n \end{bmatrix} \]

\[ \begin{bmatrix} \lambda_k^k & \ldots & -\lambda_k^{-1} & k = 1, \ldots, n \\ \lambda_k^{k} & \ldots & -\lambda_k^{-1} \end{bmatrix} \]

\[ = \frac{1}{2n \pi} \left( e^{i\lambda_k} - e^{-i\lambda_k} \right) \cdot A^2 \]

where \( A = \det[v_{n+k} \ldots n] \). For \( 1 \leq k \leq n \),

\[ (i\alpha) ^j = (-1)^j (i\alpha^{-k})^{-j} = i^{j} (\alpha^k - \alpha^{-k}) = i^{j} (2i \cdot \sin(k\theta)). \]

\[ (z^n)_{j=1} = (2i)^n \det M. \]

where \( M_{j,k} = \sin(k\theta) \). According to Muir (3. p. 187),

determinants of this type were first investigated by Prouhet.
in 1857. He used the trig identity
\[ 2^{k-1} \cos^{k-1}(\theta) \sin^k(\theta) = \sum_{p=0}^{k-1} \binom{k-1}{p} \sin((k-2p)\theta) \]
and elementary properties of determinants to show \( \det M = \det \tilde{M} \) where \( \tilde{M}_{j,k} = 2^{k-1} \cos^{k-1}(j\theta) \sin(j\theta) \) and then factored \( \sin(j\theta) \) from the jth row to arrive at the nth order Vandermonde determinant \( \text{Vand}(2\cos(j\theta))_{j=1}^n \).

If \( 1 \leq k \leq n \), then there is an \( m, 1 \leq m \leq n \), such that
\[ \lambda_{n+k} = -\lambda_m. \]
Using this fact, \( \prod_{k=1}^{2n} (1+\lambda_k^2) = \prod_{k=1}^{n} (1-\alpha^{2k})^2 = \prod_{k=1}^{n} (-\alpha^{-k} \sin(k\theta))^2 \), which is
\[ (2i)^{2n} \left( \prod_{k=1}^{n} \sin(k\theta) \right)^2. \]
Noting that \( \alpha^{n(n+1)} = 1 \) and assembling the pieces,
\[ \det A = \frac{1}{(2i)^{2n}} \left( \prod_{k=1}^{n} \sin(k\theta) \right)^2 \cdot \prod_{k=1}^{n} (e^{i\alpha^k} - e^{-i\alpha^k}) \]
\[ \cdot \prod_{k=1}^{n} \sin(k\theta) \cdot \left( \prod_{k=1}^{n} \sin(k\theta) \right)^2 \{	ext{Vand}(2\cos(j\theta))_{j=1}^n \}^2 \]
and the proof is complete.

**Lemma II** If \( 1 \leq k \leq n \), then
(a) \[ \beta_k \equiv \frac{(-1)^n}{\prod_{m \neq k} (\lambda_m - \lambda_k)} = -\frac{\sin(k\theta)}{(n+1)^2 \lambda_k} \]
(b) \[ \beta_{n+k} \equiv \frac{(-1)^n}{\prod_{m \neq n+k} (\lambda_m - \lambda_{n+k})} = \frac{\sin(k\theta)}{(n+1)^2 \lambda_{n+k}} \]

**Proof:** (a) \[ \prod_{m \neq k} (\lambda_m - \lambda_k) = \prod_{1 \leq m \leq n} (i\alpha^m - i\alpha^k) = \prod_{m=1}^{n} (i\alpha^m - i\alpha^k) \]
\[ \prod_{m \neq k} (1-\alpha^{m-k}) = \prod_{1 \leq m \leq n} (1-\alpha^{m-k}) = \prod_{m=1}^{n} (1-\alpha^{m-k}) \]

\( \prod_{1 \leq m \leq n} (i\alpha^m - i\alpha^k) \)
In his book *Plane Trigonometry*, Hobson (1) derives the equation

\[ \sqrt{n+1} = 2 \frac{\sin \frac{\pi}{n+1}}{\sin \frac{\pi}{n+1}} \prod_{j=1}^{n} \frac{1}{\sin (k\theta)} \]
\[
\sin \frac{n\pi}{2(n+1)} \quad \text{if } n \text{ is even. Using this fact, the last factor in the last equation can be simplified.}
\]

(i) \( n \) odd \( \sum_{j=1}^{n} \prod_{j=1}^{n} (1 - \cos(j\theta)) = \prod_{j=1}^{n-1} 2 \prod_{j=1}^{n+1} (1 - \cos(j\theta)) \cdot (1 - \cos(\frac{2\theta}{2})) \cdot \prod_{j=1}^{n+1} (1 - \cos(j\theta)) \)

\[
\begin{align*}
\frac{n-1}{2} & \prod_{j=1}^{n+1} (1 - \cos(j\theta)) \cdot (1 - \cos(\frac{2\theta}{2})) \cdot \prod_{j=1}^{n+1} (1 - \cos(j\theta)) \\
\sum_{j=1}^{n-1} & = 2^n \prod_{j=1}^{n} \sin^2(j\theta) \\
& = n+1
\end{align*}
\]

(ii) \( n \) even \quad \text{similar to (i).}

For further simplification, note that \( \frac{2n}{2(n+1)} = k \cdot 2n, k \cdot 2n, 2(n+1), n+1 \).

\[
\beta_k = \frac{(-1)^n}{\prod_{m \neq k} (\lambda_m - \lambda_k)} = \frac{(-1)^n}{(-1)^{n+1} \lambda_k - 2 \cdot \prod_{m \neq k} \lambda_m - \lambda_k} = \frac{-\sin(k\theta)}{n+1} \lambda_k^2
\]

(b) The techniques used to prove this are essentially the same as those used to prove (a).

Corollary If for each positive integer \( n \), \( c_n \) is the least number \( k \) such that for all \( f \) in \( H^n[0,1] \), \( \|f\|_\infty \leq \kappa \cdot \|f\|_{n,2} \), then

\[
c_n = \left\{ \frac{1}{n+1} \sum_{k=1}^{n} \frac{2 \sin^3(k\theta)}{\tanh(\sin(k\theta))} \right\}^{1/2} \quad \text{where } \theta = \frac{\pi}{n+1}.
\]

Moreover, \( \lim_{n \to \infty} c_n = \left\{ 2 \cdot \int_0^1 \frac{\sin^3(\pi x)}{\tanh(\sin(\pi x))} \, dx \right\}^{1/2} \).
Proof: Let \( n \) be a positive integer. The least \( k \) such that for all \( f \) in \( H^k([0,1]) \), \( \|f\|_\infty \leq k \cdot \|f\|_{n,2} \), is just the norm of the imbedding operator \( J: H^k([0,1]) \to C([0,1]) \). For \( a \) in \([0,1]\), define \( B_a: H^k([0,1]) \to \mathbb{R} \) by \( B_a(f) = f(a) \). Now, \( |B_a| \leq |J| \) for all \( a \). Let \( \beta = \sup\{|B_a|: a \in [0,1]\} \) and suppose \( \beta < |J| \). Then there is an \( f \) in \( H^k([0,1]) \) such that \( \|f\|_\infty > \beta \cdot \|f\|_{n,2} \).

Let \( a \in [0,1] \) such that \( |f(a)| = \|f\|_\infty \). Then

\[
\|f\|_\infty = |f(a)| = |B_a(f)| \leq |B_a| \cdot \|f\|_{n,2} \leq \beta \cdot \|f\|_{n,2} < \|f\|_\infty
\]

and we have a contradiction, so \( \beta = |J| \). Now, finding \( |B_a| \) directly from the definition is as difficult as finding \( |J| \) directly. Instead, we use the fact that \( |B_a| = |B_a^*| \) and Theorem IV. Define \( \phi: [0,1] \to \mathbb{R} \) by \( \phi(a) = |B_a^*|^2 \). From the definition of adjoint, for all \( f \) in \( H^k([0,1]) \), \( \langle f, B_a^*1 \rangle_{H^n} = \langle B_a^*f, 1 \rangle_{H^n} = f(a) \).

\[
\phi(a) = |B_a^*|^2 = \|B_a^*1\|_{n,2}^2 = \langle B_a^*1, B_a^*1 \rangle_{H^n} = (B_a^*1)(a).
\]

Applying the theorem with \( K \) defined by \( K(a,b) = b \) and \( g \) the step function with unit jump at \( a \), we obtain

\[
\phi(a) = \frac{1}{n+1} \sum_{k=1}^{n} \frac{\sin(k\theta)}{\sinh(\sin(k\theta))} \left\{ \cosh((1-2a)\sin(k\theta)) - \cos(2k\theta)\cosh(\sin(k\theta)) \right\}.
\]

\( \phi \) is symmetric and since

\[
\phi''(a) = \frac{1}{n+1} \sum_{k=1}^{n} \frac{4 \sin^3(k\theta)}{\sinh(\sin(k\theta))} \cosh((1-2a)\sin(k\theta))
\]

is greater than 0, \( \phi \) is convex. Therefore it must assume its maximum at 0 and 1. Evaluating at 0 and doing some algebra, the first part of the theorem is proved. The second part follows from the first and the definition of
Riemann integral.

As mentioned in Chapter 1, the determination of the best $c$ for the case $n = 1$ was made by Marti (2) using methods from the calculus of variations. The Corollary extends significantly Marti's result by giving explicitly the optimum $c$ for all $n$. Table I lists the values of $c_n$ for $1 \leq n \leq 10$. 
CHAPTER BIBLIOGRAPHY


CHAPTER IV

NUMERICAL SOLUTIONS OF SECOND ORDER PROBLEMS WITH NONLINEAR BOUNDARY CONDITIONS

As mentioned in the introduction, one of the most attractive features of steepest descent is that, in addition to giving a valuable theoretical framework, the method is amenable to modern numerical methods. This chapter discusses the practical aspects of implementing dual steepest descent to solve second order problems and explains the FORTRAN program FH2.CGR. A listing of the code is given in the Appendix.

Suppose \( F: \mathbb{R}^4 \rightarrow \mathbb{R} \) and \( B: \mathbb{R}^4 \rightarrow \mathbb{R} \), both \( C^2 \), are given. Find a member \( u \) of \( H^2[a,b] \) such that for \( x \) in \( [a,b] \),

\[
F(x,u(x),u'(x),u''(x)) = 0
\]

\[
B(u(a),u(b),u'(a),u'(b)) = 0
\]

We approximate the general problem using finite differences (finite elements could also be used, giving more accuracy at the expense of a more complex code). Let \( n \) be a positive integer: find \( \{u_i\}_{i=1}^n \) which satisfies within a preassigned tolerance the nonlinear difference equations:

\[
(1) \quad F(\delta^2,(i-1),u_i,\frac{u_{i+1} - u_{i-1}}{2\delta},\frac{u_{i+1} - 2u_i + u_{i-1}}{\delta^2}) = 0
\]
where \(2 \leq i \leq n-1\) and \(\delta = 1/(n-1)\). Note that (1) is the only natural difference scheme to use if the equation to be solved is truly second order, i.e., \(F_4\) is not identically zero, and the standard finite-difference approximation to \(u''(x)\) is used. However, for the boundary condition portion of the problem there are several other reasonable choices, for example.

\[
B(u_1, u_n, \frac{u_{n-1} - u_{n-2}}{2\delta}, \frac{u_{n} - u_{n-1}}{2\delta}) = 0, \quad \text{or}
\]

\[
B(u_2, u_{n-1}, \frac{u_{n} - u_{n-1}}{2\delta}, \frac{u_{n-1} - u_{n-2}}{2\delta}) = 0.
\]

Computations should show if one is superior.

To solve (*) by steepest descent, first define

\[
T: \mathbb{R}^n \to (\mathbb{R}^4)^{n-2} \text{ by } (Tu)_i = (\delta \cdot (i-1), u_i, (D_1 u)_i, (D_2 u)_i),
\]

\(2 \leq i \leq n-1\), where \(D_1\) and \(D_2\) are the standard difference operators defined above. Define \(H: \mathbb{R}^n \to \mathbb{R}^{n-2}\) by \(H(u)_i = F((Tu)_i)\) and \(\phi: \mathbb{R}^n \to \mathbb{R}\) by \(\phi(u) = 1/2 \|Hu\|^2_{\mathbb{R}^{n-2}} : \phi\) measures how close a vector \(u\) comes to satisfying (1). Similarly, define \(\zeta: \mathbb{R}^n \to \mathbb{R}\) by

\[
\zeta(u) = 1/2 \left| B(u_1, u_n, \frac{u_{n-1} - u_{n-2}}{2\delta}, \frac{u_{n} - u_{n-1}}{2\delta}) \right|^2.
\]

Finally, define \(\xi: \mathbb{R}^n \to \mathbb{R}\) by \(\xi(u) = s_1 \phi(u) + s_2 \zeta(u)\), where \(s_1\) and \(s_2\) are positive scale factors or weights which allow the boundary conditions to be emphasized over the
differential equation or vice versa. If \( u \in \mathbb{R}^n \) such that \( \xi(u) = 0 \), then \( u \) is a solution to (1) and (2), and every solution is a zero of \( \xi \).

Following Neuberger in (7), for \( 2 \leq i \leq n-2 \),

\[
(H(v) - H(u))_i = F((Tu)_i) - F((Tu)_i)
\]

\[
= F_2((Tu)_i)(v_i - u_i) + F_3((Tu)_i)(D1(v-u)_i) +
\]

\[
F_4((Tu)_i)(D2(v-u)_i) + \phi((Tv)_i, (Tu)_i)
\]

where the last term goes to zero as \((Tv)_i \to (Tu)_i\).

So, \( H'(u)(w)_i = F_2((Tu)_i) \cdot w_i + F_3((Tu)_i) \cdot (D1w)_i +
\]

\[
F_4((Tu)_i) \cdot (D2w)_i.
\]

Using the chain rule,

\[
\phi'(u)w = \langle Hu, H'(u)w \rangle_{\mathbb{R}^{n-2}} = \sum_{i=2}^{n-1} (Hu)_i \langle H'(u)w \rangle_{I}
\]

\[
= \sum_{i=2}^{n-1} F(Tu)_i \cdot F_2(Tu)_i \cdot w_i + F(Tu)_i \cdot F_3(Tu)_i \cdot D1w_i +
\]

\[
F(Tu)_i \cdot F_4(Tu)_i \cdot D2w_i
\]

\[
= \langle H_2^* u \cdot Hu + D1^* H_3^* u \cdot Hu + D2^* H_4^* u \cdot Hu, w \rangle_{\mathbb{R}^{n-2}} + \langle H_2^* u \cdot Hu, D1w \rangle_{\mathbb{R}^{n-2}} + \langle H_2^* u \cdot Hu, D2w \rangle_{\mathbb{R}^{n-2}}
\]

where \((Jr)_i \equiv \begin{cases} 0 & i=1 \text{ or } n \\ r_i & 2 \leq i \leq n-1 \end{cases} \). The last sum can be written as \( \langle J^* H_2^* u \cdot Hu + D1^* H_3^* u \cdot Hu + D2^* H_4^* u \cdot Hu, w \rangle_{\mathbb{R}^{n-2}} \), so

\[
(v\phi)(u) = J^* H_2^* u \cdot Hu + D1^* H_3^* u \cdot Hu + D2^* H_4^* u \cdot Hu.
\]

Note that the adjoints above refer to \( \mathbb{R}^n \) and \( \mathbb{R}^{n-2} \) equipped with the Euclidean inner product.

To compute \( v\phi \), define \( S: \mathbb{R}^n \to \mathbb{R}^4 \) by

\[
Su = (u_1 \cdot u_n - u_{n-2}, 2 \cdot \delta, 2 \cdot \delta)
\]

and \( Gu = B(su) \). Then
Using the chain rule, $v_G(u) = B(Su) \cdot v_G(u)$.

At this point we depart from the standard steepest descent method by employing the Sobolev inner product. Widely used in theoretical work, the Sobolev inner product has been used by Neuberger (3, 4, 5, 6, 7) and by Glowinski (8) to give markedly superior results in the numerical case.

Consider $\mathbb{R}^n$ equipped with the inner product

$$
\langle u, v \rangle_S = \sum_{i=1}^{n-1} u_i v_i + \langle D^1 u \rangle_i \langle D^1 v \rangle_i + \langle D^2 u \rangle_i \langle D^2 v \rangle_i
$$

Recalling that both gradients and adjoints depend on the inner product, we propose to move in the direction $-v_S f(u)$ to seek a decrease in $f$. The given gradients are converted to Sobolev ones by noting that $\langle u, v \rangle_S = \langle u, Av \rangle$ where $A = J^* J + D^1 D^1 + D^2 D^2$ and using the standard result: If $(X, \langle \cdot, \cdot \rangle)$ is a Hilbert space, $A \in L(X, X)$ is positive and self-adjoint, $\langle (x, y) \rangle = \langle x, Ay \rangle$, and $f : X \to \mathbb{R}$ has gradient at $x$, then the
gradient of $f$ in the space $(X, ((\cdot, \cdot)_S))^\perp$, $(\psi^A_\phi f)(x)$, is just $A^{-1}\nabla f(x)$. As noted earlier, it is useful to compute the cosine of the angle between the vectors $\psi^A_\phi(u)$ and $\psi^A_\phi\xi(u)$ in the Sobolev space. When the angle is small, the process moves in a direction that better satisfies both the differential equation and the boundary condition, whereas if the angle is large, the two "pull in opposing directions."

In any descent algorithm, one moves from the present position $u$ in a prescribed direction $z$ (usually the negative gradient) to $u + \alpha \cdot z$; there are many methods of determining how far one should move, i.e., of minimizing the function $h: [0, \infty) \to \mathbb{R}$ given by $h(\alpha) = \xi(u + \alpha \cdot z)$. Following Hestenes (I, p. 59) we use the secant approximation to Newton's method: to find $\gamma$ such that $h'(\gamma) = 0$, try

$$\gamma = \frac{h'(0)}{h''(0)} \approx \frac{h'(0)}{h'(\varepsilon) - h'(0)}$$

where $\varepsilon$ is a small positive number, say $10^{-3}$ or $10^{-4}$.

Defining $L: \mathbb{R} \to X$ by $L(\alpha) = u + \alpha \cdot z$, $h = \xi[L]$ and $h'(\alpha) \beta = \xi'(L(\alpha)) L'(\alpha) \beta = \langle \psi^A_\phi \xi(L(\alpha)), L'(\alpha) \beta \rangle_S = \langle \psi^A_\phi \xi(u + \alpha \cdot z), \beta \rangle_S$,

so

$$\gamma = \frac{\varepsilon \langle \psi^A_\phi \xi(u), z \rangle_S}{\langle \psi^A_\phi \xi(u + \varepsilon z) - \psi^A_\phi \xi(u), z \rangle_S}$$

Many numerical studies indicate that a nonlinear version of Hestenes' original conjugate-gradient algorithm gives superior results to the basic steepest descent process. Following Glowinski, Keller, and Reinhart (8)
we shall use the Polak-Ribiere (2) variant of the conjugate-gradient algorithm:

(1) Pick \( u^0 \) : \( g^0 = v_S f(u^0) \); \( z^0 = g^0 \)

(2) For \( n \) greater than 0, \( \rho_n \) minimizes \( h(\alpha) = f(u^n - \alpha z^n) \),

\[
\begin{align*}
u^{n+1} &= u^n - \rho_n z^n \\
g^{n+1} &= v_S f(u^{n+1}) \\
\gamma_n &= \frac{\langle g^{n+1} - g^n, g^{n+1} \rangle_S}{\| g^n \|_S^2} \\
z^{n+1} &= g^{n+1} + \gamma_n z^n
\end{align*}
\]

One of the most attractive features of a conjugate-gradient algorithm is that in the special case where \( f \) is quadratic, i.e., the underlying equations are linear, it is well known that convergence theoretically must occur in fewer than \( n+1 \) steps, where \( X = \mathbb{R}^n \). In the nonlinear case, as a solution is approached, the \( f \) function will increasingly approximate a quadratic, giving good convergence. In numerical applications it is difficult to maintain the conjugate-orthogonality, on which the success of the method depends, over a large number of iterations. For this reason, one should restart the algorithm every ten or fifteen iterations.

The program FH2.CGR is a FORTRAN implementation of the above ideas; the code is briefly reviewed. Section 1 dimensions the needed arrays - 2200 8-byte words are required - and defines the functions \( F \) and \( B \), together with their partials. Note that the construction allows for very general nonlinear differential equations and boundary
conditions and that the storage requirements are quite moderate. The various parameters used by the program are set up in Section 2—number of iterations, scale factors, interval of solution, etc. In Section 3, the main and 2 super diagonals of the pentadiagonal, positive definite, symmetric matrix \( ABD = J^*J + D1^*D1 + D2^*D2 \) are computed and stored in a format to be input to the LINPACK subroutine DPBFA which factors \( ABD \) so that later the linear system 
\[(ABD)w = v_\xi(x)\] can be solved.

The remaining code is within a DO loop which controls the steepest descent iteration. Sections 4 and 5 compute the Euclidean gradients \( v_\xi(u) \) and \( v_\zeta(u) \). Mention should be made of the method in which adjoints are computed; in the ODE case it is not critical, but careful coding now will pay off in the PDE case. We take the operator \( D2 \) as an example: in FORTRAN it is implemented by

\[
\text{DIMENSION } U(N), \text{ D2U(N)}
\]

\[
\text{DO 100 } I = 2, (N-1)
\]

\[
100 \quad \text{D2U(I)} = (U(I+1) - 2*U(I) + U(I-1))/(DX**2)
\]

Where the \( n \)-dimensional array \( D2U \) is treated as a \( (n-2) \)-vector by considering that only the entries 2 through \( n-1 \) contain data. To compute \( D2^* \), one could find the matrix representation of \( D2 \), take its transpose, and determine that \( D2^* \) is implemented by
DIMENSION V(N), D2A(N)
D2A(1) = V(2)/DX**2
D2A(2) = (-2*V(2) + V(3))/DX**2
DO 100 I = 3, (N-2)
100 D2A(I) = (V(I+1) + 2*V(I) + V(I-1))/DX**2
D2A(N-1) = (V(N-2) - 2*V(N-1))/DX**2
D2A(N) = V(N-1)/DX**2

A much easier way of computing $D2^*$ is obtained by considering how it acts on a unit vector $e_j$, $2 \leq j \leq n-1$ in the space $\mathbb{R}^{n-2}$. For all $u$ in $\mathbb{R}^n$ we want $\langle u, D2^*e_j \rangle_{\mathbb{R}^n} = \frac{u_{j+1} - 2u_j + u_{j-1}}{\delta^2}$, so we must have

$$ (D2^*e_j)_i = \begin{cases} 
1 & i=j-1 \text{ or } j+1 \\
\frac{\delta^2}{2} & i=j \\
0 & \text{otherwise}
\end{cases} $$

Using this and the linearity of $D2^*$, the following particularly simple code computes $D2^*$:

DIMENSION V(N), D2A(N)
DO 100 I = 2, (N-1)
D2A(I-1) = V(I)/DX**2
D2A(I) = -2*V(I)/DX**2
100 D2A(I+1) = V(I)/DX**2

This difference in perspective - taking $V(I)$ and seeing which elements of $D2A$ it affects rather than taking $D2A(I)$ and using elements of $V$ to compute its value, results in substantial simplification in the corresponding PDE code where five adjoints are computed. Moreover, rather than computing the adjoints separately the work can all be done
in one DO loop.

Section 6 computes the gradient $\nabla f(u)$ and the value $f(u)$. If $f(u)$ is less than the preassigned tolerance, a solution has been found to the desired accuracy and the program exits. The LINPACK subroutine DPBSL is called twice in Section 7 to compute $\nabla S f(u)$ and $\nabla S g(u)$, and then the cosine of the angle between them is found. Section 8 implements the conjugate-gradient algorithm, determining the direction $z$ in which to move. Note the variable $IREST$ which controls how often the CG algorithm is restarted. The secant formula is used in Section 9 to determine the optimum distance to move.

Finally, in Section 10, we move to the trial vector $w = u^n - R_2 \cdot z^n$ and compute $f(w)$. The ratio $RR = \frac{f(w)}{f(u^n)}$ should be $\leq 1$, but because of the nonlinearities we may have moved too far. If $RR > 1$, we go back and move just half as far in the direction $z$. The process continues up to ten times until $f(w) < f(u^n)$, whereupon $u^{n+1} \leftarrow w$, and the ratios $\frac{f(u^{n+1})}{f(u^n)}$, $\frac{\nabla f(u^{n+1})}{\nabla f(u^n)}$, and $\frac{\nabla g(u^{n+1})}{\nabla g(u^n)}$ are printed out.

Table II lists the output of several runs of FH2.CGR for the test problem $y' = y$, $y(0) = y(1)^2$. With an initial guess of the constant 1 function, the program quickly converges to the solution $e^{x-2}$. Starting at -1 or 0.1, the
other solution, 0, is reached. In runs #4 and #5, different weights are attached to the differential equation and boundary conditions. Note that the ratios \( \frac{\phi(u^{n+1})}{\phi(u^n)} \) and \( \frac{\xi(u^{n+1})}{\xi(u^n)} \) can vary widely, but \( \frac{\xi(u^{n+1})}{\xi(u^n)} \) stays below 1, indicating continued descent. Finally in run #6, the conjugate gradient method is restarted at every step, so regular steepest descent is done. With IREST = 1, convergence is slower, but much less erratic.
CHAPTER BIBLIOGRAPHY


TABLE I

Least $C$ for Sobolev's Imbedding Inequality

\[ \|f\|_\infty \leq C \cdot \|f\|_{n,2} \]

<table>
<thead>
<tr>
<th>$N$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.14587751766902701</td>
</tr>
<tr>
<td>2</td>
<td>1.11280304092518190</td>
</tr>
<tr>
<td>3</td>
<td>1.11229455523920118</td>
</tr>
<tr>
<td>4</td>
<td>1.1128314131571282</td>
</tr>
<tr>
<td>5</td>
<td>1.11228287031259754</td>
</tr>
<tr>
<td>6</td>
<td>1.1122828380032815</td>
</tr>
<tr>
<td>7</td>
<td>1.11228286384338547</td>
</tr>
<tr>
<td>8</td>
<td>1.11228286383960049</td>
</tr>
<tr>
<td>9</td>
<td>1.11228286363950901</td>
</tr>
<tr>
<td>10</td>
<td>1.11228286363950679</td>
</tr>
</tbody>
</table>
### TABLE II

**Program**: FH2.CCR  
**Equation**: \( u' = u \); \( u(0) = u(1)^2 \)

On the interval \([0.0, 1.0]\)

**Run #1**

\( N = 15 \)  \( \text{ITER} = 20 \)  \( \text{TOLERANCE} = 0.0000001 \)  \( \text{EPSILON} = 0.001 \)

\( \text{SCALE1} = 1.0 \)  \( \text{SCALE2} = 1.0 \)  \( \text{RESTART INT.} = 5 \)

**Initial guess**: \( u(x) = 1.0 \)

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \xi_{\text{new}} )</th>
<th>( \phi_{\text{new}} )</th>
<th>( \xi_{\text{old}} )</th>
<th>( \phi_{\text{old}} )</th>
<th>Distance Moved</th>
<th>Cosine</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.02297</td>
<td>0.018</td>
<td>0.0</td>
<td>0.0</td>
<td>0.507</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0.055857</td>
<td>0.051</td>
<td>0.042</td>
<td>5.250</td>
<td>0.438</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.344713</td>
<td>0.959</td>
<td>0.848</td>
<td>0.733</td>
<td>0.648</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.337422</td>
<td>0.311</td>
<td>0.458</td>
<td>5.403</td>
<td>-0.119</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.514833</td>
<td>0.488</td>
<td>0.598</td>
<td>0.514</td>
<td>-0.850</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.455134</td>
<td>0.514</td>
<td>0.206</td>
<td>9.146</td>
<td>-0.268</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.175810</td>
<td>0.213</td>
<td>0.015</td>
<td>50.312</td>
<td>-0.589</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.921007</td>
<td>0.913</td>
<td>1.456</td>
<td>0.571</td>
<td>-0.815</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.675432</td>
<td>0.688</td>
<td>0.241</td>
<td>24.612</td>
<td>-0.292</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.959044</td>
<td>0.954</td>
<td>1.566</td>
<td>0.924</td>
<td>-0.572</td>
<td></td>
</tr>
</tbody>
</table>

**APPROXIMATE SOLUTION AFTER 10 ITERATIONS**

| 0.124455 | 0.144280 | 0.155511 | 0.158852 | 0.179032 | 0.192125 |
| 0.206179 | 0.221248 | 0.237454 | 0.254949 | 0.273920 | 0.294529 |
| 0.316819 | 0.340533 | 0.354837 | | | |

| 0.773107 | 0.784 | 0.042 | 6.200 | 0.695 |
| 0.945754 | 0.537 | 11.492 | 58.711 | -0.280 |
| 0.963397 | 0.970 | 0.853 | 8.117 | -0.253 |
| 0.491338 | 0.464 | 2.328 | 7.769 | -0.883 |
| 0.380245 | 0.962 | 0.082 | 52.412 | -0.838 |
| 0.930735 | 0.929 | 1.117 | 4.835 | -0.581 |
| 0.772209 | 0.777 | 0.124 | 2.197 | 0.759 |
| 0.901011 | 0.902 | 0.259 | 44.488 | -0.004 |
| 0.870451 | 0.864 | 17.129 | 1.110 | -0.893 |
| 0.935846 | 0.936 | 0.973 | 7.972 | -0.220 |

**APPROXIMATE SOLUTION AFTER 20 ITERATIONS**

| 0.135973 | 0.146267 | 0.156390 | 0.168578 | 0.131033 | 0.194378 |
| 0.290699 | 0.224108 | 0.240700 | 0.258540 | 0.277687 | 0.298164 |
| 0.320253 | 0.344197 | 0.389135 | | | |
Run # 2

\[ N = 15 \quad \text{ITER} = 20 \quad \text{TOLERANCE} = 0.0000001 \quad \text{EPSILON} = 0.001 \]

\[ \text{SCALE1} = 1. \quad \text{SCALE2} = 1. \quad \text{RESTART INT.} = 5 \]

Initial guess: \( u(x) = -1 \).

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \frac{\delta_{\text{new}}}{\delta_{\text{old}}} )</th>
<th>( \frac{\phi_{\text{new}}}{\phi_{\text{old}}} )</th>
<th>( \frac{\delta_{\text{new}}}{\delta_{\text{old}}} )</th>
<th>Distance Moved</th>
<th>Cosine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.024627</td>
<td>0.011</td>
<td>0.069</td>
<td>0.419</td>
<td>0.587</td>
</tr>
<tr>
<td>2</td>
<td>0.270031</td>
<td>0.353</td>
<td>0.227</td>
<td>4.101</td>
<td>0.603</td>
</tr>
<tr>
<td>3</td>
<td>0.334403</td>
<td>0.267</td>
<td>0.478</td>
<td>0.498</td>
<td>0.810</td>
</tr>
<tr>
<td>4</td>
<td>0.192316</td>
<td>0.068</td>
<td>0.249</td>
<td>6.337</td>
<td>-0.625</td>
</tr>
<tr>
<td>5</td>
<td>0.243803</td>
<td>1.037</td>
<td>0.826</td>
<td>1.179</td>
<td>-0.854</td>
</tr>
<tr>
<td>6</td>
<td>0.147816</td>
<td>0.195</td>
<td>0.154</td>
<td>10.560</td>
<td>-0.879</td>
</tr>
<tr>
<td>7</td>
<td>0.897035</td>
<td>1.236</td>
<td>0.759</td>
<td>1.424</td>
<td>-0.606</td>
</tr>
<tr>
<td>8</td>
<td>0.715835</td>
<td>1.125</td>
<td>0.445</td>
<td>2.440</td>
<td>-0.921</td>
</tr>
<tr>
<td>9</td>
<td>0.424352</td>
<td>0.191</td>
<td>0.816</td>
<td>0.740</td>
<td>-0.923</td>
</tr>
<tr>
<td>10</td>
<td>0.762809</td>
<td>1.101</td>
<td>0.630</td>
<td>5.825</td>
<td>-0.714</td>
</tr>
</tbody>
</table>

Approximate solution after 10 iterations:
-0.010818 -0.012083 -0.013298 -0.014534 -0.015814 -0.018495 -0.019897 -0.021364 -0.022940 -0.024683

Tolerance met after 20 iterations:
-0.000140 -0.000202 -0.000263 -0.000325 -0.000387 -0.000449 -0.000511 -0.000573 -0.000635 -0.000697

Solution to tolerance after 20 iterations:
-0.000140 -0.000202 -0.000263 -0.000325 -0.000387 -0.000449 -0.000511 -0.000573 -0.000635 -0.000697
Run # 3

\( N = 15 \quad ITER = 20 \quad TOLERANCE = 0.0000001 \quad EPSILON = 0.001 \)

\( SCALE1 = 1. \quad SCALE2 = 1. \quad RESTART INT. = 5 \)

Initial guess : \( u(x) = 0.1 \)

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \frac{\xi_{\text{new}}}{\xi_{\text{old}}} )</th>
<th>( \frac{\phi_{\text{new}}}{\phi_{\text{old}}} )</th>
<th>( \frac{\epsilon_{\text{new}}}{\epsilon_{\text{old}}} )</th>
<th>Distance Moved</th>
<th>Cosine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.018135</td>
<td>0.016</td>
<td>0.045</td>
<td>0.491</td>
<td>0.987</td>
</tr>
<tr>
<td>2</td>
<td>0.106167</td>
<td>0.017</td>
<td>0.628</td>
<td>8.948</td>
<td>-0.309</td>
</tr>
<tr>
<td>3</td>
<td>0.988131</td>
<td>0.689</td>
<td>1.035</td>
<td>10.50</td>
<td>-0.963</td>
</tr>
<tr>
<td>4</td>
<td>0.317371</td>
<td>2.782</td>
<td>0.557</td>
<td>81.733</td>
<td>-0.947</td>
</tr>
<tr>
<td>5</td>
<td>0.951598</td>
<td>0.905</td>
<td>1.190</td>
<td>1.730</td>
<td>-0.785</td>
</tr>
<tr>
<td>6</td>
<td>0.531684</td>
<td>0.530</td>
<td>0.540</td>
<td>6.751</td>
<td>0.547</td>
</tr>
<tr>
<td>7</td>
<td>0.273018</td>
<td>0.098</td>
<td>0.939</td>
<td>19.300</td>
<td>-0.642</td>
</tr>
<tr>
<td>8</td>
<td>0.972873</td>
<td>1.206</td>
<td>0.880</td>
<td>2.979</td>
<td>-0.934</td>
</tr>
<tr>
<td>9</td>
<td>0.025587</td>
<td>0.071</td>
<td>0.001</td>
<td>21.369</td>
<td>-0.948</td>
</tr>
<tr>
<td>10</td>
<td>0.848208</td>
<td>0.855</td>
<td>0.555</td>
<td>4.434</td>
<td>-0.101</td>
</tr>
</tbody>
</table>

Approximate solution after 10 iterations:

\[
\begin{array}{cccccc}
-0.000060 & -0.000045 & -0.000049 & -0.000052 & -0.000080 & -0.000075 \\
-0.000081 & -0.000101 & -0.000103 & -0.000105 & -0.000118 & -0.000148 \\
-0.000183 & -0.000186 & -0.000188 & -0.000198 & -0.000207 & -0.000216 \\
\end{array}
\]

Tolerance met after 11 iterations: \( 0.806867 \)

Solution to tolerance after 12 iterations:

\[
\begin{array}{cccccc}
-0.000103 & -0.000102 & -0.000110 & -0.000122 & -0.000133 \\
-0.000157 & -0.000169 & -0.000172 & -0.000175 & -0.000187 & -0.000215 \\
-0.000249 & -0.000250 & -0.000250 & -0.000250 & -0.000250 & -0.000250 \\
\end{array}
\]
Run # 4

\[ N = 15 \quad \text{ITER} = 20 \quad \text{TOLERANCE} = 0.0000001 \quad \text{EPSILON} = 0.001 \]

\[ \text{SCALE1} = .8 \quad \text{SCALE2} = .2 \quad \text{RESTART INT.} = 5 \]

Initial guess: \( u(x) = 1 \).

<table>
<thead>
<tr>
<th>( k )</th>
<th>( t_{\text{new}} )</th>
<th>( t_{\text{old}} )</th>
<th>( \phi_{\text{new}} )</th>
<th>( \phi_{\text{old}} )</th>
<th>Distance Moved</th>
<th>Cosine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.017840</td>
<td>0.016</td>
<td>0.0359</td>
<td>0.0359</td>
<td>0.635</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.013668</td>
<td>0.016</td>
<td>0.0537</td>
<td>0.016</td>
<td>0.022</td>
<td>0.469</td>
</tr>
<tr>
<td>3</td>
<td>0.979743</td>
<td>0.982</td>
<td>1.078</td>
<td>1.078</td>
<td>0.622</td>
<td>0.324</td>
</tr>
<tr>
<td>4</td>
<td>0.748617</td>
<td>0.743</td>
<td>0.425</td>
<td>0.425</td>
<td>0.722</td>
<td>-0.484</td>
</tr>
<tr>
<td>5</td>
<td>0.870341</td>
<td>0.849</td>
<td>0.910</td>
<td>0.910</td>
<td>-0.324</td>
<td>0.722</td>
</tr>
<tr>
<td>6</td>
<td>0.240660</td>
<td>0.239</td>
<td>0.244</td>
<td>0.244</td>
<td>81.295</td>
<td>-0.484</td>
</tr>
<tr>
<td>7</td>
<td>0.722348</td>
<td>0.647</td>
<td>0.848</td>
<td>0.848</td>
<td>27.015</td>
<td>-0.558</td>
</tr>
<tr>
<td>8</td>
<td>0.958381</td>
<td>0.927</td>
<td>1.001</td>
<td>1.001</td>
<td>1.489</td>
<td>-0.558</td>
</tr>
<tr>
<td>9</td>
<td>0.311413</td>
<td>0.352</td>
<td>0.293</td>
<td>0.293</td>
<td>18.958</td>
<td>0.707</td>
</tr>
<tr>
<td>10</td>
<td>0.784809</td>
<td>0.840</td>
<td>0.697</td>
<td>0.697</td>
<td>64.004</td>
<td>-0.695</td>
</tr>
</tbody>
</table>

APPROXIMATE SOLUTION AFTER 10 ITERATIONS

| 0.144920 | 0.155857 | 0.167201 | 0.179488 | 0.192792 | 0.207186 |
| 0.222649 | 0.239198 | 0.256886 | 0.275817 | 0.296166 | 0.318174 |
| 0.342076 | 0.387825 | 0.394433 |

| 0.867801 | 0.942   | 0.728   | 0.990   | 0.723   |
| 0.659712 | 0.666   | 0.640   | 20.750  | -0.869  |
| 0.634153 | 0.982   | 0.811   | 10.760  | -0.358  |
| 0.790322 | 0.443   | 0.155   | 6.514   | -0.846  |
| 0.811835 | 0.903   | 0.599   | 17.570  | -0.101  |
| 0.984002 | 0.938   | 1.026   | 1.815   | -0.748  |
| 0.605000 | 0.641   | 0.271   | 33.438  | -0.408  |
| 0.829881 | 0.854   | 0.243   | 125.376 | -0.430  |
| 0.963788 | 0.964   | 0.933   | 0.764   | -0.879  |
| 0.988040 | 0.969   | 0.729   | 18.928  | -0.096  |

APPROXIMATE SOLUTION AFTER 20 ITERATIONS

| 0.136204 | 0.148413 | 0.157100 | 0.168715 | 0.181223 | 0.194639 |
| 0.209035 | 0.224499 | 0.241117 | 0.258983 | 0.276092 | 0.298579 |
| 0.320643 | 0.344636 | 0.369704 |
Run # 5

N = 15  ITER  =  20  TOLERANCE  =  0.0000001  EPSILON  =  0.001  
SCALE1  =  .2      SCALE2  =  .8  RESTART INT.  =  5

Initial guess: u(x) = 1.

<table>
<thead>
<tr>
<th>k</th>
<th>$u_{old}$</th>
<th>$u_{new}$</th>
<th>$u_{old}$</th>
<th>$u_{new}$</th>
<th>Distance Moved</th>
<th>Cosine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.039410</td>
<td>0.017</td>
<td>0.0</td>
<td>2.511</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.224598</td>
<td>0.367</td>
<td>0.122</td>
<td>8.069</td>
<td>0.292</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.855773</td>
<td>0.978</td>
<td>0.590</td>
<td>3.622</td>
<td>0.830</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.220850</td>
<td>0.234</td>
<td>0.175</td>
<td>16.210</td>
<td>0.332</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.566544</td>
<td>0.638</td>
<td>0.225</td>
<td>2.558</td>
<td>0.894</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>0.036403</td>
<td>0.085</td>
<td>0.106</td>
<td>31.715</td>
<td>0.282</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>0.957242</td>
<td>0.917</td>
<td>1.399</td>
<td>13.683</td>
<td>-0.830</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>0.571315</td>
<td>0.624</td>
<td>0.193</td>
<td>26.933</td>
<td>0.344</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>0.893075</td>
<td>0.651</td>
<td>1.688</td>
<td>4.780</td>
<td>-0.895</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>0.756445</td>
<td>0.834</td>
<td>0.050</td>
<td>32.105</td>
<td>-0.388</td>
<td>0.0</td>
</tr>
</tbody>
</table>

APPROXIMATE SOLUTION AFTER 10 ITERATIONS

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0.132629</td>
<td>0.143223</td>
<td>0.154031</td>
<td>0.165389</td>
<td>0.177505</td>
<td>0.190524</td>
</tr>
<tr>
<td>12</td>
<td>0.204563</td>
<td>0.219736</td>
<td>0.236162</td>
<td>0.253980</td>
<td>0.273231</td>
<td>0.294010</td>
</tr>
<tr>
<td>13</td>
<td>0.316206</td>
<td>0.339502</td>
<td>0.363222</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0.878304</td>
<td>0.875</td>
<td>1.321</td>
<td>3.299</td>
<td>-0.887</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.354105</td>
<td>0.311</td>
<td>4.765</td>
<td>420.374</td>
<td>-0.496</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.869705</td>
<td>0.829</td>
<td>0.477</td>
<td>2.818</td>
<td>0.835</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.642283</td>
<td>0.692</td>
<td>0.006</td>
<td>44.747</td>
<td>-0.575</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.984188</td>
<td>0.980</td>
<td>7.565</td>
<td>17.111</td>
<td>-0.533</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.868679</td>
<td>0.860</td>
<td>2.581</td>
<td>24.281</td>
<td>0.752</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.747449</td>
<td>0.756</td>
<td>0.186</td>
<td>118.908</td>
<td>-0.347</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.580877</td>
<td>0.593</td>
<td>0.031</td>
<td>488.056</td>
<td>-0.321</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.643902</td>
<td>0.567</td>
<td>389.243</td>
<td>3.477</td>
<td>-0.901</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.869443</td>
<td>0.986</td>
<td>0.013</td>
<td>134.321</td>
<td>-0.911</td>
<td></td>
</tr>
</tbody>
</table>

APPROXIMATE SOLUTION AFTER 20 ITERATIONS

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0.135070</td>
<td>0.145327</td>
<td>0.156005</td>
<td>0.167553</td>
<td>0.179977</td>
<td>0.193290</td>
</tr>
<tr>
<td>12</td>
<td>0.207582</td>
<td>0.222972</td>
<td>0.239545</td>
<td>0.257332</td>
<td>0.276350</td>
<td>0.296708</td>
</tr>
<tr>
<td>13</td>
<td>0.319683</td>
<td>0.342532</td>
<td>0.367423</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Run # 6

N = 15    ITER = 20    TOLERANCE = 0.0000001    EPSILON = 0.001
SCALE1 = 1.    SCALE2 = 1.    RESTART INT. = 1

Initial guess : \( u(x) = 1 \).

<table>
<thead>
<tr>
<th>k</th>
<th>$\xi_{\text{new}}$</th>
<th>$\phi_{\text{new}}$</th>
<th>$\xi_{\text{old}}$</th>
<th>$\phi_{\text{old}}$</th>
<th>Distance</th>
<th>Cosine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.022237</td>
<td>0.015</td>
<td>0.0</td>
<td>0.507</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.226452</td>
<td>0.202</td>
<td>0.045</td>
<td>4.408</td>
<td>0.438</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.477460</td>
<td>0.405</td>
<td>1.791</td>
<td>0.533</td>
<td>-0.563</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.501032</td>
<td>0.572</td>
<td>0.209</td>
<td>3.971</td>
<td>-0.654</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.543372</td>
<td>0.447</td>
<td>1.623</td>
<td>0.534</td>
<td>-0.516</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>0.588106</td>
<td>0.667</td>
<td>0.337</td>
<td>3.875</td>
<td>-0.705</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>0.650501</td>
<td>0.525</td>
<td>1.416</td>
<td>0.536</td>
<td>-0.742</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>0.715709</td>
<td>0.824</td>
<td>0.470</td>
<td>3.942</td>
<td>0.171</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>0.757874</td>
<td>0.658</td>
<td>1.265</td>
<td>0.536</td>
<td>-0.742</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>0.837491</td>
<td>0.992</td>
<td>0.577</td>
<td>4.195</td>
<td>-0.742</td>
<td>0.0</td>
</tr>
</tbody>
</table>

APPROXIMATE SOLUTION AFTER 10 ITERATIONS

| 11 | 0.156999           | 0.168928           | 0.181078           | 0.193824           | 0.207462 | 0.22233 |
|    | 0.238321           | 0.255872           | 0.274981           | 0.295691           | 0.317980 | 0.341742 |
|    | 0.366759           | 0.392681           | 0.418885           |                   |          |          |

| 12 | 0.809937           | 0.793              | 1.138              | 0.544              | -0.775   | 0.0    |
|    | 0.917074           | 0.850              | 1.163              | 0.548              | -0.801   | 0.0    |
|    | 0.922720           | 0.681              | 4.638              | -0.446             | 0.0      | 0.0    |
|    | 0.927977           | 0.869              | 1.161              | 0.550              | -0.817   | 0.0    |
|    | 0.930080           | 1.006              | 0.703              | 4.700              | -0.482   | 0.0    |
|    | 0.932175           | 0.878              | 1.164              | 0.551              | -0.825   | 0.0    |
|    | 0.933520           | 1.001              | 0.715              | 4.714              | -0.492   | 0.0    |
|    | 0.934907           | 0.884              | 1.166              | 0.551              | -0.831   | 0.0    |
|    | 0.936072           | 0.989              | 0.721              | 4.717              | -0.494   | 0.0    |

APPROXIMATE SOLUTION AFTER 20 ITERATIONS

| 13 | 0.148437           | 0.159734           | 0.171272           | 0.189426           | 0.198453 | 0.210543 |
|    | 0.225843           | 0.242477           | 0.260553           | 0.280158           | 0.301342 | 0.324083 |
|    | 0.348240           | 0.373456           | 0.399102           |                   |          |          |

0.56
APPENDIX

C PROGRAM NAME: FH2.CGR  STEEPEST DESCENT, SOBOLEV NORM
C SOLVES  F(X,U,U',U''')=0, B(U(0),U(1),U'(0),U'(1))
C BEGIN DATE: 22 FEB 1984
C REVISED:  24 OCT 1984

C-----------------------------------------------SECTION 1

IMPLICIT REAL*8 (A-H,0-Z) .
REAL*8 U(200),GR(200), BGR(200),XIGR(200),SXIGR(200)
REAL*8 WR1(200),Z(200),SGOLD(200)
REAL*8 ABD(3,200)
REAL*8 F,F2,F3,F4,B,B1,B2,B3,B4

C###

F(X,U,U1,U2)= U1 - U
F2(X,U,U1,U2)=-1.D0
F3(X,U,U1,U2)= 1.D0
F4(X,U,U1,U2)= 0.D0
B(A,B,C,D)= A - B**2
B1(A,B,C,D)= 1.D0
B2(A,B,C,D)=-2.D0*B
B3(A,B,C,D)= 0.D0
B4(A,B,C,D)= 0.D0
WRITE(6,1)
FORMAT(IX,'PGM=FH2.CGR  '/,
     * 1X,'EQ: IT  ' = U '  {'
     * 1X,'BC: U(0)  = U(1)* *2'

C-----------------------------------------------SECTION 2

N=15
IF(N.GT.200) CALL EXIT
NM1=N-1
NM2=N-2
NM3=N-3
NM1SQ=NM1**2
PI = 4.D0*DATAN(1.D0)
ELFT= 0.D0
ERHT= 1.D0
DX=(ERHT-ELFT)/(N-1)
ITER =20
ITMOD=1
MODPNT=10
IREST=1
RIN = 1.0
TOLER=1.0D-7
EPSILN= 1.0D-03
SCALE1=1.0
SCALE2=1.0
WRITE(6,10) ELFT,ERHT,N,ITER,TOLER,EPSILN,
* SCALE1,SCALE2,IREST
10 FORMAT(1X,'ON INTERVAL ',F5.2,' TO ',F5.2,/,
* 1X,'N= ',I5,' ITER= ',I10,
* ' TOLERANCE= ',E15.7,' EPSILON=',F10.7/,,
* 1X,' SCALE1= ',F5.3,' SCALE2= ',F5.3,
* ' RESTART INT.=',I4/,,
* 20X,' INITIAL GUESS: ')
DO 20 I=1,N
20 U(I)=RIN
WRITE(6,990) (U(I),I=1,N)

C---------------------------------------------SECTION 3

ALP= 1./(4.*DX**2)
BET= 1./(DX**4)
ABD(1,1)=0.
ABD(1,2)=0.
DO 35 J=3,N
35 ABD(1,J)= -ALP + BET
ABD(2,1)= 0.
ABD(2,2)= -2.*BET
DO 40 J=3,NM1
40 ABD(2,J)= -4.*BET
ABD(2,N)= -2.*BET
ABD(3,1)= ALP + BET
ABD(3,2)= 1. + ALP + 5.*BET
DO 45 J=3,NM2
45 ABD(3,J)= 1. + 2.*ALP + 6.*BET
ABD(3,N-1)= 1. + ALP + 5.*BET
ABD(3,N)= ALP + BET
CALL DPBFA(ABD,3,N,2,INFO)
IF(INFO.NE.0) CALL EXIT

C---------------------------------------------SECTION 4

PHIOLD=0.DO
DO 80 I=1,N
80 GR(I)=0.
DO 100 I=2,NM1
CX=DX*(I-1)
CU=U(I)
CU1= (U(I+1)-U(I-1))/(2.*DX)
CU2= (U(I+1)-2.*U(I)+U(I-1))/(DX**2)
CF=F(CX,CU,CU1,CU2)
CF2=F2(CX,CU,CU1,CU2)
CF3=F3(CX,CU,CU1,CU2)
CF4=F4(CX,CU,CU1,CU2)
PHIOLD=PHIOLD + CF*CF
GR(I-1)=GR(I-1) - (CF3*CF)/(2.*DX) + (CF4*CF)/(DX**2)
GR(I)=GR(I) + CF2*CF - (2.*CF4*CF)/(DX**2)
100 GR(I+1)=GR(I+1) + (CF3*CF)/(2.*DX) + (CF4*CF)/(DX**2)
PHIOLD=.5*PHIOLD
C--------------------------------------------------SECTION 5

DO 380 I=1,N

380 BGR(I) = 0.

C1 = (U(N)-U(1))/(2.*DX)
C2 = (U(N)-U(N-2))/(2.*DX)
CB = B(U(1),U(N),C1,C2)
CB1 = B1(U(1),U(N),C1,C2)
CB2 = B2(U(1),U(N),C1,C2)
CB3 = B3(U(1),U(N),C1,C2)
CB4 = B4(U(1),U(N),C1,C2)

BGR(1) = CB1*CB - (CB3*CB)/(2.*DX)
BGR(3) = (CB3*CB)/(2.*DX)
BGR(N-2) = - (CB4*CB)/(2.*DX)
BGR(N) = CB2*CB + (CB4*CB)/(2.*DX)
BCOLD = .5*CB**2

C--------------------------------------------------SECTION 6

DO 400 I=1,N

400 XIGR(I) = SCALE1*GR(I) + SCALE2*BGR(I)

XIOLD = SCALE1*PHIOLD + SCALE2*BCOLD
IF(XIOLD.GT.TOLER) GO TO 550
WRITE(6,992)
992 FORMAT(1X,' TOLERANCE MET')
GO TO 1100

C--------------------------------------------------SECTION 7

C SOBOLEV GRADIENT
CALL DPBSL(ABD,3,N,2,GR)
CALL DPBSL(ABD,3,N,2,BGR)

C** COMPUTE COSINE IN H2
SUM1=0.D0
SUM2=0.D0
DOT=0.D0

DO 625 I=2,NM1
CG1 = (GR(I+1)-GR(I-1))/(2.*DX)
CG2 = (GR(I+1)-2.*GR(I)+GR(I-1))/(DX**2)
CB1 = (BGR(I+1)-BGR(I-1))/(2.*DX)
CB2 = (BGR(I+1)-2.*BGR(I)+BGR(I-1))/(DX**2)
SUM1=SUM1+ GR(I)**2 + CG1**2 + CG2**2
SUM2=SUM2 + BGR(I)**2 + CB1**2 + CB2**2

625 DOT=DOT + GR(I)*BGR(I) + CG1*CB1 + CG2*CB2
DENOM=DSQRT(SUM1*SUM2)
COSINE=0.D0
IF(DENOM.GT.0.) COSINE= DOT/DENOM

DO 635 I=1,N
635 SGOLD(I)=SXIGR(I)

DO 640 I=1,N
640 SXIGR(I) = SCALE1*GR(I) + SCALE2*BGR(I)

C--------------------------------------------------SECTION 8

IF(MOD(KK,IREST).NE.0) GO TO 644

DO 642 I=1,N
642 Z(I)=SXIGR(I)
GO TO 650
644 CONTINUE
DO 645 I=1,N
645 WR1(I)=SXIGR(I)-SGOLD(I)
DOT=0.DO
DO 646 I=2,NM1
CW1= (WR1(I+1)-WR1(I-1))/(2.*DX)
CW2= (WR1(I+1)-2.*WR1(I)+WR1(I-1))/(DX**2)
CS1= (SXIGR(I+1)-SXIGR(I-1))/(2.*DX)
CS2= (SXIGR(I+1)-2.*SXIGR(I)+SXIGR(I-1))/(DX**2)
646 DOT = DOT + WR1(I)*SXIGR(I) + CW1*CS1 + CW2*CS2
SUM1=0.DO
DO 647 I=2,NM1
647 SUM1=SUM1+ SGOLD(I)**2 +
*   ((SGOLD(I+1)-SGOLD(I-1))/(2.*DX))**2 +
*   ((SGOLD(I+1)-2.*SGOLD(I)+SGOLD(I-1))/(DX**2))**2
GAMMA=0.DO
IF(SUM1.GT.0.DO) GAMMA= DOT/SUM1
DO 648 I=1,N
648 Z(I)=SXIGR(I) + GAMMA*Z(I)
DO 600 I=1,N
800 R1 = 0.DO
DO 805 I=2,NM1
CZ1= (Z(I+1)-Z(I-1))/(2.*DX)
CZ2= (Z(I+1)-2.*Z(I)+Z(I-1))/(DX**2)
CS1= (SXIGR(I+1)-SXIGR(I-1))/(2.*DX)
CS2= (SXIGR(I+1)-2.*SXIGR(I)+SXIGR(I-1))/(DX**2)
805 R1 = R1 + Z(I)*SXIGR(I) + CZ1*CS1 + CZ2*CS2
DO 810 I=1,N
810 GR(I)=0.
DO 850 I=2,NM1
CX=DX*(I-1)
CUM1=U(I-1) + EPSILN*Z(I-1)
CU= U(I) + EPSILN*Z(I)
CUP1=U(I+1) + EPSILN*Z(I+1)
CU1= (CUP1-CUM1)/(2.*DX)
CU2= (CUP1-2.*CU+CUM1)/(DX**2)
CF=F(CX,CU,CU1,CU2)
CF2=F2(CX,CU,CU1,CU2)
CF3=F3(CX,CU,CU1,CU2)
CF4=F4(CX,CU,CU1,CU2)
GR(I-1)=GR(I-1) - (CF3*CF)/(2.*DX) + (CF4*CF)/(DX**2)
GR(I)=GR(I) + CF2*CF - (2.*CF4*CF)/(DX**2)
GR(I+1)=GR(I+1) + (CF3*CF)/(2.*DX) + (CF4*CF)/(DX**2)
DO 890 I=1,N
890 BGR(I)=0.
CW1=U(1)+EPSILN*Z(1)
CW3=U(3)+EPSILN*Z(3)
CWNM2=U(N-2)+EPSILN*Z(N-2)
CWN = U(N) + EPSILN*Z(N)
C1 = (CW3-CW1)/(2.*DX)
C2 = (CWN-CWNM2)/(2.*DX)
CB = B(CW1,CWN,C1,C2)
CB1 = B1(CW1,CWN,C1,C2)
CB2 = B2(CW1,CWN,C1,C2)
CB3 = B3(CW1,CWN,C1,C2)
CB4 = B4(CW1,CWN,C1,C2)
BGR(1) = CB1*CB - (CB3*CB)/(2.*DX)
BGR(3) = (CB3*CB)/(2.*DX)
BGR(N-2) = -(CB4*CB)/(2.*DX)
BGR(N) = CB2*CB + (CB4*CB)/(2.*DX)
DO 900 I=1,N
900 WR1(I) = SCALE1*GR(I) + SCALE2*BGR(I)
CALL DPBSL(ABD,3,N,2,WR1)
R2 = 0.DO
DO 920 I=2,NM1
CW1 = (WR1(I+1)-WR1(I-1))/(2.*DX)
CW2 = (WR1(I+1)-2.*WR1(I)+WR1(I-1))/(DX**2)
CZ1 = (Z(I+1)-Z(I-1))/(2.*DX)
CZ2 = (Z(I+1)-2.*Z(I)+Z(I-1))/(DX**2)
920 R2 = R2 + WR1(I)*Z(I) + CW1*CZ1 + CW2*CZ2
R3 = (EPSILN*R1)/(R2-R1)
C-----------------------------SECTION 10
ICOUNT=0
925 DO 930 I=1,N
930 WR1(I) = U(I) - R3*Z(I)
PHINEW = 0.DO
DO 935 I=2,NM1
CX = DX*(I-1)
CU = WR1(I)
CU1 = (WR1(I+1)-WR1(I-1))/(2.*DX)
CU2 = (WR1(I+1)-2.*WR1(I)+WR1(I-1))/(DX**2)
935 PHINEW = PHINEW + F(CX, CU, CU1, CU2)**2
PHINEW = .5*PHINEW
C1 = (WR1(3)-WR1(1))/(2.*DX)
C2 = (WR1(N)-WR1(N-2))/(DX**2)
BCNEW = .5*B(WR1(1),WR1(N),C1,C2)**2
XINEW = SCALE1*PHINEW + SCALE2*BCNEW
RR = 0.DO
RPHI = 0.DO
RBC = 0.DO
IF(XIOLD.NE.0.0) RR = XINEW/XIOLD
IF(PHIOLD.NE.0.0) RPHI = PHINEW/PHIOLD
IF(BCOLD.NE.0.0) RBC = BCNEW/BCOLD
IF(RR.LT.0.999999999) GO TO 950
WRITE(6,996) KK,RR
993 FORMAT(1X,I10,' RR = ',F12.6,', .GT. 0.999999999')
R3 = .5*R3
ICOUNT = ICOUNT+1
IF(ICOUNT.LT.5) GO TO 925
WRITE(6,996)
IF(MOD(KK,ITMOD).EQ.0) WRITE(6,994) KK,RR,RPHI,RBC,R3,COSINE
   994 FORMAT(1X,I7,F10.6,3F10.3,5X,F10.3)
   IF(MOD(KK,MODPNT).EQ.0) WRITE(6,990) (U(I),I=1,N)
   990 FORMAT(1X,6F12.6)
C---------------------------------------------------------------SECTION 10---------------
1000 CONTINUE
C**
1100 CONTINUE
   WRITE(6,994) KK,RR,RPHI,RBC,R3
   WRITE(6,990) (U(I),I=1,N)
   CON=0.62737639353D0
   CALL EXIT
END
BIBLIOGRAPHY

Books


Articles


---


Rosen, G. "Minimum Value for c in the Sobolev Inequality $\|u\|_p \leq c \|u\|_2$." *SIAM Journal on Applied Mathematics*, 21 (July, 1971), 30-32.


Reports