1993

A Variational Method For Numerical Differentiation.

Robert O'Brien Wallace Jr
University of Alabama at Birmingham

Order Number 9405301

A variational method for numerical differentiation

Wallace, Robert O'Brien, Jr., Ph.D.

University of Alabama at Birmingham, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

A VARIATIONAL METHOD FOR NUMERICAL DIFFERENTIATION

by

ROBERT WALLACE

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Joint Program in Applied Mathematics in The Graduate Schools, The University of Alabama, The University of Alabama at Birmingham and The University of Alabama in Huntsville

BIRMINGHAM, ALABAMA

1993

ABSTRACT OF DISSERTATION
GRADUATE SCHOOL, UNIVERSITY OF ALABAMA AT BIRMINGHAM

Degree Doctor of Philosophy Major Subject Applied Mathematics

Name of Candidate Robert Wallace

Title A Variational Method for Numerical Differentiation

A mathematical problem is said to be well posed if it has general properties of existence, uniqueness, and stability of solutions with respect to given metrics. Problems which are not well posed are said to be ill-posed. This notion of a mathematical problem being well posed was first introduced by J. Hadamard [27] and was believed for years to characterize mathematical problems having physical relevence. In addition, algorithms for solving ill-posed problems tend to be unstable, making numerical solution of such problems extremely difficult. For these reasons, ill-posed problems received little attention until A. N. Tikhonov gave a precise definition to the idea of approximate solutions of ill-posed problems. He together with co-workers developed a general theory of ill-posed problems that included techniques for constructing approximate solutions [41,42,43].

Among the large number of ill-posed problems that are currently studied is the broad class of inverse problems. One important subclass of inverse problems is the following variety. Given "sufficient" information about the solution of the following differential equation

$$Lu = 0, \tag{1}$$

where L is a differential operator, one seeks to recover one or more of the coefficient functions.

Many ill-posed problems can be reformulated as inverse problems of the type (1) that we just described. One such problem is that of numerical differentiation. Given a function u to be differentiated, it is clear that finding u'' is equivalent to recovering the coefficient function Q in the differential equation

$$-u'' + Qu = 0. \tag{2}$$

As variational methods are typically numerically stable, we seek to develop a variational method for solving the inverse problem associated with (2) and hence for numerical differentiation.

In this thesis, we begin by discussing some of the most common methods for numerical differentiation. Then we introduce our variational approach to numerical differentiation. Numerical results together with some details of our implementation are given in Chapter 4. We conclude with a discussion of generalizations of this method that may be applied to other inverse problems, including the computation of higher order derivatives.

Abstract Approved by: Committee Chairman ____________

Program Director ____________

Date 9/27/93 Dean of Graduate School ____________

Acknowledgments

I wish to express my appreciation to Professor Ian Knowles for guidance and encouragement during the preparation of this dissertation. In addition, I am indebted to Professor John Neuberger for many helpful discussions on the topics covered herein.

Contents

Chapter 1

Preliminaries

We begin with a description of notation that we use throughout the thesis. All functions are real-valued and defined on the closed interval $[a, b]$, unless we indicate otherwise. The set of functions that are continuous on $[a, b]$ is written $\mathcal{C}$. Any function g, for which $g^{(n)}$ is in $\mathcal{C}$ is said to be in $\mathcal{C}^n$. Absolutely continuous functions on $[a, b]$ are said to be in $\mathcal{AC}$. The Banach space of functions g, such that $|g|^p$ is integrable, is written $\mathcal{L}^p$ (for $1 \leq p < \infty$), and $\|\cdot\|_p$ denotes the usual norm on $\mathcal{L}^p$. The space, $\mathcal{L}^\infty$, of essentially bounded measurable functions, has the standard norm which we denote by $\|\cdot\|_\infty$. The Sobolev space of functions g in $\mathcal{L}^2$ for which g, g', $\cdots$, $g^{(q)}$ are in $\mathcal{L}^2$ is denoted by $\mathcal{H}^q$ (for $1 \leq q < \infty$) and the corresponding standard norm by $\|\cdot\|_{\mathcal{H}^q}$. The spaces $\mathcal{L}^2$ and $\mathcal{H}^q$ have induced inner products which we express as $(\cdot,\cdot)$ and $(\cdot,\cdot)_q$, respectively.

1.1 General Theory of Sturm-Liouville Equations

Consider the second order Sturm-Liouville equation

$$-(pv')' + qv = f \tag{1.1.1}$$

where $p > 0$ and $1/p$, q, and f are in $\mathcal{L}^1[a, b]$. Then a function u, continuous on $[a, b]$ is said to be a solution of (1.1.1) if u is in $\mathcal{AC}$ (so that u' exists), pu' is in $\mathcal{AC}$, and u satisfies (1.1.1) almost everywhere in $[a, b]$. Existence of a unique solution of (1.1.1) satisfying prescribed initial conditions is guaranteed by the following theorem.

Theorem 1.1.1 ([28, p. 323]) *Let A and A' be any two real numbers and let c be in $[a,b]$. Then there exists a unique solution u for (1.1.1) such that $u(c) = A$ and $p(c)u'(c) = A'$.*

At this point it is useful to examine the homogeneous case of (1.1.1),

$$-(pv')' + qv = 0. \tag{1.1.2}$$

Equation (1.1.2) has exactly two linearly independent solutions that we call u_1 and u_2. As the solution space is two dimensional, any solution of (1.1.2) can be expressed as a linear combination of u_1 and u_2. It can be easily shown that $W(u_1, u_2) = p(u_1'u_2 - u_1u_2')$ is identically equal to a constant for any pair of solutions u_1 and u_2 of (1.1.2). Further, $W(u_1, u_2) = 0$ if and only if u_1 and u_2 are linearly dependent. In addition, for any solution u_1 of (1.1.2) that is nonzero on $[a,b]$, the function given by $u_2(x) = u_1(x)\int_a^x [p(s)u_1(s)^2]^{-1}\,ds$ is also a solution of (1.1.2) that is linearly independent from u_1. Given independent solutions u_1 and u_2 of (1.1.2) together with any solution $\tilde{u}$ of (1.1.1), any other solution u of (1.1.1) can be written in the form $u = \tilde{u} + c_1u_1 + c_2u_2$ for some constants c_1 and c_2.

Given any two linearly independent solutions u_1 and u_2, define

$$R(x,\xi) = \kappa^{-1}[u_1(x)u_2(\xi) - u_1(\xi)u_2(x)] \tag{1.1.3}$$

where $\kappa = W(u_1, u_2)$. This function R, sometimes called the influence function or the one-sided Green's function has a number of interesting properties. First R is independent of the choice of the independent pair u_1, u_2. This may be shown directly but also follows from the fact that for ξ fixed

$$-\frac{d}{dx}(p(x)\frac{d}{dx}R(x,\xi)) + q(x)R(x,\xi) = 0$$

subject to the condition $R(x,\xi)|_{x=\xi} = 0$ and $\frac{d}{dx}R(x,\xi)|_{x=\xi} = p(\xi)^{-1}$ for x in $(\xi, b]$. Finally, R allows one to construct a particular solution for equation (1.1.1) as an integral operator applied to the function f, as stated in the following theorem.

Theorem 1.1.2 ([49, pp. 118–120]) *The unique solution of (1.1.1) subject to $v(a) = p(a)v'(a) = 0$ is given by*

$$u(x) = \int_a^x R(x,\xi)f(\xi)\,d\xi.$$

Clearly any solution of (1.1.1) can be written in the form

$$u(x) = c_1 u_1(x) + c_2 u_2(x) + \int_a^x R(x,\xi) f(\xi)\, d\xi.$$

Now let u_1 and u_2 be linearly independent solutions of (1.1.2) and assume that

$$D = u_1(a)u_2(b) - u_1(b)u_2(a) \neq 0. \tag{1.1.4}$$

Then the Green's function G defined on $[a,b] \times [a,b]$ is defined by

$$G(x,\xi) = \begin{cases} \kappa^{-1} D^{-1}[u_1(\xi)u_2(a) - u_1(\xi)u_2(a)][u_1(x)u_2(b) - u_1(x)u_2(b)], & a \le \xi \le x \le b \\ \kappa^{-1} D^{-1}[u_1(x)u_2(a) - u_1(x)u_2(a)][u_1(\xi)u_2(b) - u_1(\xi)u_2(b)], & a \le x \le \xi \le b. \end{cases}$$

The function G is independent of u_1 and u_2. As in the case of R, the fact that G is independent of u_1 and u_2 can be verified directly or can be observed as a consequence of the fact that G is the unique solution of the initial boundary value problem

$$\begin{gathered} -\tfrac{d}{dx}(p(x)\tfrac{d}{dx}G(x,\xi)) + q(x)G(x,\xi) = 0 \\ G(a,\xi) = G(b,\xi) = 0 \\ \lim_{x\to\xi^-} G(x,\xi) = \lim_{x\to\xi^+} G(x,\xi) \\ \lim_{x\to\xi^-} G_x(x,\xi) = \lim_{x\to\xi^+} G_x(x,\xi) = \tfrac{1}{p(\xi)} \end{gathered} \tag{1.1.5}$$

for x in $[a,b]$ and ξ a fixed number in $[a,b]$. The main purpose for constructing this Green's function is to allow us to construct the solution of certain boundary value problems as an integral operator applied to the given data as is done in the following theorem.

Theorem 1.1.3 ([28, pp. 341–343]) *Assume that for some pair of linearly independent solutions of (1.1.2) that $D \neq 0$. Then the solution of (1.1.1) subject to the boundary conditions $v(a) = v(b) = 0$ is*

$$u(x) = \int_a^b G(x,\xi) f(\xi)\, dx. \tag{1.1.6}$$

1.2 Spectral Theory for Sturm-Liouville Operators

Define the operator L by

$$Lv = -(pv')' + qv \tag{1.2.7}$$

for all v in $\mathcal{D}(L)$, the set of all functions v in $\mathcal{L}^2$ such that v and pv' are absolutely continuous, Lv is in $\mathcal{L}^2$, and $v(a) = v(b) = 0$. This operator L is self-adjoint, see [16, p. 1291]. Most of the remaining discussion in this section can be found in any standard reference, for example see [6, §7.1]. As L must also be symmetric, all eigenvalues of L are real.

One way to further characterize the eigenvalues of L is to note that they are exactly those values λ for which the Green's function associated with $L - \lambda$ fails to exist. Further, if $\lambda = 0$ is not an eigenvalue of L then the boundary value problem

$$Lv = f \tag{1.2.8}$$

$$v(a) = v(b) = 0 \tag{1.2.9}$$

has a unique solution given by (1.1.6)

The collection of eigenvalues of L forms a countably infinite set $\{\lambda_n\}$ such that $\lambda_n \leq \lambda_{n+1}$ for any integer $n \geq 1$. The set $\{\lambda_n\}$ has no cluster point and is bounded below, hence the limit of $\{\lambda_n\}$ as a sequence is infinity.

Let $\mathcal{A}$ be the set of all functions v in $\mathcal{AC}$ such that pv' is in $\mathcal{AC}$, and $v(a) = v(b) = 0$. Set $\mathcal{A}_0$ to be the set of functions v in $\mathcal{AC}$ for which pv' is in $\mathcal{AC}$, Lv is in $\mathcal{L}^2$, and v has compact support strictly contained in $[a, b]$. Then define $\mathcal{A}_1$ to be the completion of $\mathcal{A}_0$ with respect to the energy norm

$$\|v\|_\varepsilon^2 = \int_a^b (pv'^2 + qv^2)\, dx.$$

Note that $\mathcal{A}$ is a subset of $\mathcal{A}_1$. Also note that the Rayleigh quotient $R(v) = (Lv, v)/\|v\|_2^2$ is defined for all v in $\mathcal{A}_1$. Consequently, we may state the following theorem.

Theorem 1.2.1 **([8, pp. 399])** *There exists a function u_1, that minimizes the Rayleigh quotient R, over all v in $\mathcal{A}_1$. Further, u_1 is in $\mathcal{A}$, $R(u_1) = \lambda_1$ where λ_1 is the smallest eigenvalue of L, and u_1 is an eigenfunction associated with λ_1.*

The eigenspace associated with each eigenvalue λ_n is one dimensional. As a result, one can associate with each eigenvalue a normalized eigenfunction u_n. This set $\{u_n\}$ forms a complete orthonormal set in $\mathcal{L}^2$. A discussion of these ideas can be found in [49, §§36, 38].

1.3 Disconjugate Equations

The second order ordinary differential equation (1.1.2) is said to be disconjugate on the interval $[a, b]$ if every nontrivial solution has at most one zero in $[a, b]$. The property of disconjugacy characterizes those equations that may be solved uniquely given arbitrarily prescribed boundary conditions. This is precisely stated in the following theorem.

Theorem 1.3.1 ([28, p. 351]) *Equation (1.1.2) is disconjugate if and only if for every pair of distinct points x_1 and x_2 in $[a, b]$ and arbitrary numbers A and B, there exists a unique solution u of (1.1.2) satisfying $u(x_1) = A$ and $u(x_2) = B$; or equivalently, if and only if every pair of linearly independent solutions v_1 and v_2 of (1.1.2) satisfy $v_1(x_1)v_2(x_2) - v_1(x_2)v_2(x_1) \neq 0$ for distinct points x_1 and x_2 in $[a, b]$.*

A corollary of Theorem 1.3.1 is that equation (1.1.2) is disconjugate on $[a, b]$ if every nontrivial solution has at most one zero in $[a, b]$. As a consequence, the following fact serves as a useful test for disconjugacy of equation (1.1.2).

Proposition 1.3.2 *Assume that $p \equiv 1$. Then, equation (1.1.2) is disconjugate on $[a, b]$ if $\int_a^b q^- \, dx \leq 4(b-a)^{-1}$ where q^- is the negative part of q.*

This proposition is a corollary of a theorem given in [28, Theorem 5.1, p. 345]. An additional property of disconjugate equations is that they have an associated variational principal that is often useful in numerical as well as theoretical applications of disconjugate boundary value problems.

Theorem 1.3.3 ([7, p. 10]) *Equation (1.1.2) is disconjugate on $[a, b]$ if and only if the functional*

$$F(v) = \int_a^b (pv'^2 + qv^2)\, dx$$

defined on $\mathcal{A}_1$ is positive.

Next we state Dirichlet's principle for the second order ODE case. It it worth noting that Dirichlet's principle has a PDE analogue that is used in Chapter 5.

Theorem 1.3.4 *If the hypotheses of Theorem 1.3.1 hold then the unique solution u of equation (1.1.1) subject to the boundary condition $u(a) = A$ and $u(b) = B$ satifies*

$$\int_a^b (pv'^2 + qv^2 - 2fv)\, dx \geq \int_a^b (pu'^2 + qu^2 - 2fu)\, dx$$

for any piecewise continuously differentiable function v satisfying the same boundary conditions as u, with equality if and only if $v = u$.

A proof of this theorem can be found in any standand text on PDEs, for example see [49, pp. 392–393].

1.4 Calculus of Variations

Let $\mathcal{X}$ and $\mathcal{Y}$ be Banach spaces and let U be a connected subset of $\mathcal{X}$ containing the zero element. Let r be a map from $\mathcal{X}$ into $\mathcal{Y}$. Then $r(x)$ is said to be $o(\|x\|_{\mathcal{X}})$ as x tends to zero if $r(x)/\|x\|_{\mathcal{X}}$ tends to zero as x tends to the zero element in $\mathcal{X}$ and we write $r(x) = o(\|x\|_{\mathcal{X}})$. The function r is $o(1)$ as x tends to zero if the limit of $r(x)$ as x goes to zero is zero. If $r(x)/\|x\|_{\mathcal{X}}$ is bounded as x tends to zero then we say that $r(x) = O(\|x\|_{\mathcal{X}})$. A more thorough discussion of the material that follows may be found in [51, §§4.1–4.3].

A function f from a neighborhood U of x in $\mathcal{X}$ is said to be Fréchet differentiable if there exists a bounded linear map T, from $\mathcal{X}$ into $\mathcal{Y}$ such that

$$f(x+h) = f(x) + Th + o(\|h\|_{\mathcal{X}})$$

as h tends to zero. The map T is called the Fréchet derivative of f at x and we write $f'(x) = T$.

The function f is said to be Gâteaux differentiable at x if there is a bounded linear map T, from $\mathcal{X}$ into $\mathcal{Y}$ such that

$$f(x+th) = f(x) + tTh + o(t)$$

as t goes to zero. T is called the Gâteaux derivative of f at x which we denote $\delta f(x)$.

If the Fréchet derivative exists for all x in a set U then the mapping that takes x to $f'(x)$ is called the Fréchet derivative of f on U. A similar definition exists for the Gâteaux derivative of f on U. Higher derivatives are defined by considering f' as a function from the set U into the space of bounded functions from $\mathcal{X}$ into $\mathcal{Y}$. We then apply this definition of a derivative to f' to obtain f'', et cetera.

The following proposition describes the relationship between the Gâteaux and Fréchet derivatives.

Proposition 1.4.1 ([51, p. 137]) *Every Fréchet derivative at x is also a Gâteaux derivative at x.*

The Gâteaux derivative can be equivalently defined by the equation

$$f'(x)h = \lim_{t \to 0} \frac{f(x+th) - f(x)}{t}.$$

Geometrically, the Gâteaux derivative is the directional derivative of f at x in the "direction" h. If this limit exists uniformly for all h with norm one, then this is also a Fréchet derivative.

Both the Fréchet and Gâteaux derivatives satisfy most of the usual properties of the derivatives of functions defined on Euclidian space such as validity of the chain rule and linearity. In addition, we have the following analogue of Taylor's theorem. See [51, §4.3].

Theorem 1.4.2 ([51, pp. 148–149]) *Let $U(x)$ be an open convex neighborhood of x in the Banach space $\mathcal{X}$ and assume that f is a mapping that takes $U(x)$ into the Banach space $\mathcal{Y}$. If $f', f'', ..., f^{(n-1)}$ exist as Fréchet derivatives on $U(x)$, then we have that*

$$f(x+h) = f(x) + \sum_{k=1}^{n-1} \frac{1}{k!} f^{(k)}(x)(h,h,...,h) + R_n(x,h) \tag{1.4.10}$$

where $f^{(k)}(x)$ is the kth Fréchet of f at x, $R_n(x,h) = O(\|h\|_{\mathcal{X}}^n)$, and $(h,h,...,h)$ is an element of $\mathcal{X}^k$. Further, if $f^{(n)}$ exists on $U(x)$ then $R_n(x,h) = o(\|h\|_{\mathcal{X}}^n)$.

In addition, it is shown in [52, p. 249] that if $f^{(n)}$ is continuous on $U(x)$ then R_n has the explicit form

$$R_n(x,h) = \int_0^1 \frac{(1-t)^{n-1}}{(n-1)!} f^{(n)}(x+th)(h,h,...,h)\,dt$$

where $(h,h,...,h)$ is an element of $\mathcal{X}^n$.

These ideas have a number of useful applications in optimization theory that are exactly analogous to those in elementary calculus.

Proposition 1.4.3 ([52, p. 249]) *A convex Gâteaux differentiable function f, has a minimum at x if and only if $f'(x) = 0$. Moreover, if f is strictly convex and $f'(x) = 0$, then x is the unique global minimizer for x.*

The second derivatives often yield valuable information about the concavity of the function.

Proposition 1.4.4 ([52, pp. 247–248]) *A function f defined as in proposition 1.4.3 is convex on a convex set U if and only if $f''(x)$ is a positive quadratic form for all x in U. Further, if $f''(x)$ is positive definite for all x in U, then f is strictly convex on U.*

As one might expect, existence of derivatives implies certain continuity properties.

Theorem 1.4.5 ([52, p. 150]) *If f is Gâteaux differentiable at x and $f'(x)$ is strongly continuous then f is weakly sequentially continuous at x.*

That is, given any sequence $\{x_n\}$ such that $(x_n, y)_{\mathcal{X}}$ tends to $(x_*, y)_{\mathcal{X}}$ for some x_* in $\mathcal{X}$, we have that $f(x_n)$ converges to $f(x_*)$ in the space $\mathcal{Y}$.

Chapter 2

Numerical Differentiation

The problem of numerical differentiation is known to be ill-posed in the sense that small pertubations in the function to be differentiated may lead to large errors in the computed derivative. As a result, the traditional methods for numerical differentiation tend to be unstable.

Much has been written on this topic [1,2,4,9,12,13,14,15,18,23,24,26,29,31,32,33,38,40, 45,46,47,48], and a number of techniques have been developed. Most fall into one of three categories: difference methods, interpolation methods, and regularization methods. The first two techniques mentioned are the most commonly used and yield satisfactory results when the function to be differentiated is given very precisely. However, they do not address the inherent instability of numerical differentiation and may fail badly if the function is imparted with only a small amount of error. Regularization methods, on the other hand, do address this instability problem and usually give satisfactory results even when the function to be differentiated is not precisely given. Most of these regularization methods make use of Tikhonov's approach to solving ill-posed problems [41,43]; some examples include [1,2,9,12,13,14,31,38,40,46]. These methods typically involve reducing the numerical differentiation problem to a family of well-posed problems that depend on a regularization parameter. Once an optimal value for this parameter is found, the corresponding well-posed problem is solved to obtain an estimate for the derivative. Unfortunately, the determination of the optimal value for this parameter is generally a nontrivial task.

2.1 Difference and Interpolation Methods

Difference methods and interpolation methods are the techniques most commonly used methods to implement numerical differentiation. They usually yield satisfactory results when the function to be differentiated is given very precisely, but tend to be very sensitive to roundoff error. Further, because these methods do not address the inherent instability of numerical differentiation they may fail badly if the function is imparted with only a small amount of error.

We begin this section by studying the effects of roundoff error on some specific numerical differentiation techniques and how one might remedy these effects without resorting to regularization.

One way to approximate f' is to use the formula

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \tag{2.1.1}$$

which holds for h sufficiently small. However, it is well known that in cases where the precision with which f is given is limited, taking h small will lead to subtractive cancellation and potentially large roundoff error. (For example, see [44, p. 140].) The truncation error for a numerical differentiation formula is the "exact" error in the formula. In the case of formula (2.1.1), for example, the truncation error is defined to be

$$\epsilon(h) = \frac{f(x+h) - f(x)}{h} - f'(x). \tag{2.1.2}$$

Expanding $f(x+h)$ in a first order Taylor series about x one can easily show that $\epsilon(h) = o(1)$. Moreover, if f'' exists and is bounded on some neighborhood of x, then $\epsilon(h) = O(h)$ for h sufficiently small.

The basic problem with using (2.1.1) is that in order to make the truncation error small we must take h small which could lead to large roundoff error. One typical approach to solving this dilemma is to use a formula that yields a small trucation error without taking h so small that roundoff error becomes significant. Many such formulae may be derived using polynomial interpolation as discussed in [44, pp. 140–143]. For example, given that $f(x)$ is three times differentiable we have the result

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}, \tag{2.1.3}$$

where the truncation error is $\frac{1}{6}f'''(\xi)h^2$ for some ξ between $x-h$ and $x+h$. Formula (2.1.3) is commonly known as the central difference approximation for $f'(x)$. Additional formulae include

$$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + \frac{1}{3}h^2 f'''(\xi) \tag{2.1.4}$$

where $x - h < \xi < x + h$ and

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} + \frac{1}{30}h^4 f^{(5)}(\xi) \tag{2.1.5}$$

where $x - 2h < \xi < x + 2h$.

Any formula derived by Taylor series or polynomial interpolation (such as (2.1.2), (2.1.3), (2.1.4), and (2.1.5)) must have the form (see [44, p. 143])

$$f'(x_j) \approx \frac{1}{h}\sum_{k=1}^{n} a_k f(x_k) \tag{2.1.6}$$

where the numbers a_k are fixed numbers that define the formula and the points x_k satisfy $x_{i+1} - x_i = h$ for $i = 1, 2, ..., n$. In the special case $f(x) \equiv 1$, (2.1.6) reduces to

$$\frac{1}{h}\sum_{k=1}^{n} a_k = 0. \tag{2.1.7}$$

As h becomes small some of the individual terms a_k/h must become large in absolute value. As a result, subtractive cancellation and large rounding error will become significant even in formulae having small truncation error. Note that by this same reasoning, (2.1.6) shows that whenever $|f'(x_j)|$ is small compared to $|f(x_j)|$ the effects of subtractive cancellation are likely to be significant. (See Table 4.0.1, later.)

2.2 Regularization Methods

We begin this section with a simple illustrative example of the Tikhonov regularization method and then discuss briefly the use of regularization for numerical differentiation.

The term regularizaton method refers to one of a broad class of functional analytical approaches to solving ill-posed problems. The idea, simply stated, is to reformulate the ill-posed problem as a parameterized family of well-posed problems. Once an optimal value for the regularization parameter is found, the corresponding well-posed problem is solved.

Suppose that A is a continuous linear operator from a real Hilbert space $\mathcal{X}$, into a real Hilbert space $\mathcal{Y}$. Then the problem of finding u such that $Au = b$ is said to be well-posed if b is in the range of A and if there exists a constant $c > 0$ such that

$$||Au||_{\mathcal{Y}} \geq c||u||_{\mathcal{X}} \tag{2.2.8}$$

for all u in $\mathcal{X}$. Otherwise the problem is ill-posed. Now in Tikhonov regularization we consider the perturbed problem $Au_\delta = b_\delta$ where $||\ b - b_\delta|| \leq \delta$ over v in $\mathcal{X}$ where $\gamma\ > 0$. One should think of b_δ an experimentally determined approximation of the exact value b. The corresponding regularized problem is to minimize

$$G_\gamma(v) = \frac{1}{2}||Av - b_\delta||^2_{\mathcal{Y}} + \frac{1}{2}\gamma||v||^2_{\mathcal{X}}. \tag{2.2.9}$$

If there exists a solution, say $v(b_\delta, \gamma)$, of the optimization problem just stated then this solution must be a stationary value of (2.2.9). In other words, the Gâteaux differential, $G'_\gamma(v)h$ of G_γ at $v(b_\delta,\gamma)$ is zero for all h in $\mathcal{X}$. Computing the Gâteaux differential explicitly and setting it to zero yields

$$G'_\gamma(v)h = (A^*Av - A^*b_\delta + \gamma Iv, h)_{\mathcal{Y}} = 0 \tag{2.2.10}$$

for h in $\mathcal{X}$. Here A^* denotes the Hilbert adjoint of A. Hence,

$$(A^*A + \gamma I)v(b_\delta, \gamma) = A^*b_\delta. \tag{2.2.11}$$

As $A^*A + \gamma I$ is strongly positive and self-adjoint, for $\gamma > 0$, it must have an inverse. Hence the only remaining difficulty as we have mentioned earlier is in choosing γ appropriately.

We now examine Cullum's procedure [9] for applying Tikhonov's regularization ideas to the problem of numerical differentiation.

Let $g(x)$ be the function to be differentiated. Then we can assume without loss of generality that the interval of definition is $[0,1]$ and that $g(0) = g(1) = 0$. Then the problem of "given g find f such that $g' = f$" can be reformulated in terms of the following integral equation

$$(Af)(x) \equiv \int_0^1 h(y-x)f(y)\,dx = g(x) \tag{2.2.12}$$

where h is the Heaviside unit step function.

Cullum then procedes to show that problem (2.2.12) can be reduced to the following family of optimization problems which she proves is well-posed. For a given α minimize

$$G_\alpha(f) = \|Af - g\|^2 + \left(\int_0^1 f\,dx\right)^2 + \alpha\|f\|_1^2 \tag{2.2.13}$$

where $\|u\|^2 = \int_0^1 u^2\,dx$ and $\|u\|_1^2 = \int_0^1 (u'^2 + u^2)\,dx$. Here α is the regularization parameter. A calculus of variations argument then shows that for a given α is the interval $(0,1]$. The minimizer for G_α must satisfy the Fredholm integral equation of the second kind given by

$$\alpha f(x) + \int_0^1 K_0(x,y)f(y)\,dy = m(y). \tag{2.2.14}$$

Where $m(x) = \int_0^1 G(x,y)\int_y^1 g(s)\,dsdy$, G is the Green's function associated with the operator $Bu = -u'' + u$ with Neumann endpoint boundary conditions, and

$$K_0(x,y) = 2 - \max(x,y) - G(x,y) + \cosh(x)/\sinh(1).$$

The solution (2.2.14) for a given α can thus be obtained using standard finite difference methods. Cullum does not offer a method for obtaining an optimal value for α.

In [1,2] two independent statistical methods are suggested for numerical differentiation. One method is based on standard regression analysis. The other method makes use of a spectral analysis of the data to define its "optimal" numerical differentiation operator. In this case it is assumed that the correct function and the error can be regarded as a stationary time series and that the spectrum of the data shows a clear division between the signal and the noise. Anderssen and Bloomfield then show that Cullum's procedure has a spectral analysis interpretation that can be used to obtain an optimal value for α.

In the next chapter we present a new regularization procedure that makes use of the assumption that the curvature of the original function is bounded.

Chapter 3

A Variational Method for Numerical Differentiation

In this chapter, we suggest a method for numerical differentiation that we believe avoids some of the problems mentioned in Chapter 1. Given a real valued, smooth function u defined on the closed interval $[a, b]$, we construct an associated functional,

$$H(q) = \int_a^b (u'^2 + qu^2)\,dx - \int_a^b (u_q'^2 + qu_q^2)\,dx \tag{3.0.1}$$

where u_q solves the boundary value problem

$$-v'' + qv = 0 \tag{3.0.2}$$

$$v(a) = A, \quad v(b) = B; \tag{3.0.3}$$

here (and later) $A = u(a)$ and $B = u(b)$. The functional H is strictly convex and has a unique global minimum, $Q = u''/u$. Hence this minimizer for H may obtained approximately using standard optimization theory. Once Q is found, we obtain an approximation for u' by numerically solving $-u'' + Qu = 0$ as a first order system under appropriate boundary conditions. Use of an $\mathcal{H}^1$ gradient for H (rather than an $\mathcal{L}^2$ gradient) is a crucial step in our development.

In §3.1 we discuss some properties of this functional and give some preliminary results. The stability of the method is examined in §3.3. A steepest descent approach to the optimization is discussed in §3.2, and convergence of this algorithm is proven under a set of conditions that can be monitored numerically in §3.4.

3.1 Preliminaries

Let f be a function that is twice differentiable with an essentially bounded second derivative. Our task is to compute f' numerically.

Set $u(x) = f(x) + k$ for $a \leq x \leq b$, where k is a constant. Clearly, u and f have the same derivative. We shall see later that, in the present context, this constant k is somewhat analogous to the Tikhonov regularization parameter mentioned in §2.2; however, for now we need only choose k sufficiently large (given enough machine precision) to guarantee that u satisfies the following technical conditions that are needed for the theory we are about to discuss. The first condition is that

$$u(x) \geq c > 0 \tag{3.1.4}$$

for $a \leq x \leq b$; this is needed to ensure that u has no zeros. We define the function Q by $Q(x) = u''(x)/u(x)$ for $a \leq x \leq b$, and assume the second technical condition that

$$\|Q\|_\infty \leq M, \tag{3.1.5}$$

where

$$0 < M < \frac{\pi^2}{(b-a)^2}. \tag{3.1.6}$$

Condition (3.1.5) serves to control the curvature of the function u in that more curvature is permitted with $b - a$ small.

Observe that as u is a positive solution of $-v'' + Qv = 0$, this equation must be disconjugate [28, p. 351] on the interval $[a, b]$. With this in mind, we define $\mathcal{D}$ to be the set of all functions q in $\mathcal{L}^1$ such that the equation (3.0.2) is disconjugate on $[a, b]$. It follows that Q is in $\mathcal{D}$. Moreover, by condition (3.1.6), the constant function $-M$ is in $\mathcal{D}$ hence, any function q in $\mathcal{L}^1$ for which $-M \leq q$ almost everywhere in $[a, b]$, must be in $\mathcal{D}$. (See [7, pp. 20].) In addition, we have

Proposition 3.1.1 *The set $\mathcal{D}$ is convex and open in $\mathcal{L}^1$ and in $\mathcal{L}^2$.*

A proof for the $\mathcal{L}^1$ case may be found in [7, pp. 10,95]. The proof of the $\mathcal{L}^2$ case is similar.

For each function q in $\mathcal{D}$, by [28, p. 351] there exists a unique solution, u_q, for (3.0.2) satisfying the boundary conditions (3.0.3). Further, each function u_q must be positive, as (3.0.2) is disconjugate and A and B are positive, by (3.1.4).

We show that problem of computing u' can be reformulated as the problem of minimizing the functional H defined for each q in $\mathcal{D}$ by (3.0.1). In §3.3 we discuss the computation of the minimizer for H, and how finding this minimizer allows us to compute u'. But first we examine some useful properties of the functional.

For each q in $\mathcal{D}$, let the operator A_q be defined by $A_q v = -v'' + qv$, on functions v in $\mathcal{H}^2$ satisfying homogeneous Dirichlet boundary conditions at the end points of the interval $[a, b]$. In addition, let G_q be the Green's function associated with the operator A_q. Recall that G_q is continuous, hence bounded, on $[a, b] \times [a, b]$.

Lemma 3.1.2 *For fixed q in $\mathcal{D}$ and h in $\mathcal{L}^1$ such that*

$$\|h\|_1 < \frac{1}{2}\|G_q\|_\infty^{-1}, \tag{3.1.7}$$

we have the following estimates:

$$\|u_{q+h}\|_\infty \leq 2\|u_q\|_\infty; \tag{3.1.8}$$

$$\|u_{q+h} - u_q\|_\infty \leq 2\|G_q\|_\infty \|u_q\|_\infty \|h\|_1. \tag{3.1.9}$$

Proof. Subtracting the equations $-u''_{q+h} + (q+h)u_{q+h} = 0$ and $-u''_q + qu_q = 0$, and observing that $u_{q+h} - u_q$ lies in $\mathcal{D}(A_q)$, we obtain $A_q(u_{q+h} - u_q) = -hu_{q+h}$. As q is in $\mathcal{D}$, the operator A_q is positive [28, p. 352], and hence A_q may be inverted; thus

$$u_{q+h}(x) = u_q(x) - \int_a^b G_q(x, \xi) u_{q+h}(\xi) h(\xi)\, d\xi. \tag{3.1.10}$$

Applying the Hölder inequality yields $|u_{q+h}(x)| \leq \|u_q\|_\infty + \|G_q\|_\infty \|u_{q+h}\|_\infty \|h\|_1$ for $a \leq x \leq b$. Hence

$$\|u_{q+h}\|_\infty \leq \|u_q\|_\infty + \|G_q\|_\infty \|u_{q+h}\|_\infty \|h\|_1 \leq \|u_q\|_\infty + \frac{1}{2}\|u_{q+h}\|_\infty, \tag{3.1.11}$$

on using (3.1.7); (3.1.8) follows.

To see that (3.1.9) holds observe that from (3.1.10) one readily obtains

$$|u_{q+h}(x) - u_q(x)| \le \|G_q\|_\infty \|u_{q+h}\|_\infty \|h\|_1 \tag{3.1.12}$$

for $a \le x \le b$. By taking the maximum of the left side of (3.1.12) over $a \le x \le b$ and applying (3.1.8) to the right side of (3.1.12) we have (3.1.9), whenever (3.1.7) holds.

Proposition 3.1.3 *For each q in $\mathcal{D}$, the Fréchet differential of H is given by*

$$H'(q)h = \int_a^b (u^2 - u_q^2)h\,dx \tag{3.1.13}$$

for all h in $\mathcal{L}^1$, and the gradient of H in $\mathcal{L}^2$ is given by

$$\nabla H(q) = u^2 - u_q^2. \tag{3.1.14}$$

Proof. For h in $\mathcal{L}^1$ let $I(h)$ be given by

$$I(h) = H(q+h) - H(q) - \int_a^b (u^2 - u_q^2)h\,dx.$$

Using the definition of H, integration by parts, the equations for u_q and u_{q+h}, and the fact that u_q and u_{q+h} agree on the boundary of $[a,b]$, we have that

$$\begin{aligned} I(h) &= \int_a^b [-(u_q'' + u_{q+h}'') + q(u_q + u_{q+h})](u_q - u_{q+h})\,dx \\ &\qquad + \int_a^b (u_q^2 - u_{q+h}^2)h\,dx \\ &= \int_a^b (u_q - u_{q+h})u_q h\,dx. \end{aligned} \tag{3.1.15}$$

Applying Hölder's inequality to (3.1.15) and then using (3.1.9) yields

$$|I(h)| \le \|u_q - u_{q+h}\|_\infty \|u_q\|_\infty \|h\|_1 \le 2\|G_q\|_\infty \|u_q\|_\infty^2 \|h\|_1^2,$$

whenever (3.1.7) holds. The desired result follows.

Proposition 3.1.4 *For each q in $\mathcal{D}$, the second Fréchet differential of H is given by*

$$H''(q)[h,h] = 2\left(A_q^{-1}(u_q h), u_q h\right) \tag{3.1.16}$$

for each h in $\mathcal{L}^1$. Further, for each q in $\mathcal{D}$, $H''(q)$ is a positive definite quadratic form.

Proof. Let h and k be in $\mathcal{L}^1$ and define $J(h)$ by

$$J(h) = H'(q+h)k - H'(q)k - 2\int_a^b \int_a^b G_q(x,\xi)u_q(x)u_q(\xi)k(x)h(\xi)\,d\xi dx.$$

Then, by using the expression for H' given in Proposition 3.1.3, we see that

$$\begin{aligned} J(h) &= \int_a^b (u_q^2 - u_{q+h}^2)k\,dx - 2\int_a^b \int_a^b G_q(x,\xi)u_q(x)u_q(\xi)k(x)h(\xi)\,d\xi dx \quad (3.1.17)\\ &= \int_a^b [u_q(x) + u_{q+h}(x)]k(x)\int_a^b G_q(x,\xi)u_{q+h}(\xi)h(\xi)\,d\xi dx \\ &\quad -2\int_a^b \int_a^b G_q(x,\xi)u_q(x)u_q(\xi)k(x)h(\xi)\,d\xi dx, \quad (3.1.18) \end{aligned}$$

upon factoring the first integrand in (3.1.17) and using (3.1.10) to eliminate $u_q - u_{q+h}$. Applying Hölders inequality to (3.1.18) then gives

$$\begin{aligned} |J(h)| &\leq \|G_q\|_\infty \int_a^b \int_a^b |\{[u_q(x) + u_{q+h}(x)]u_{q+h}(\xi) - 2u_q(x)u_q(\xi)\}\,k(x)h(\xi)|\,d\xi dx \\ &= \|G_q\|_\infty \int_a^b \int_a^b |u_q(x)k(x)[u_{q+h}(\xi) - u_q(\xi)]h(\xi) + u_{q+h}(x)k(x)[u_{q+h}(\xi) \\ &\qquad -u_q(\xi)]h(\xi) + u_q(\xi)h(\xi)[u_{q+h}(x) - u_q(x)]k(x)|\,d\xi dx \\ &\leq \|G_q\|_\infty \int_a^b |u_q k|\,dx \int_a^b |u_{q+h} - u_q||h|\,d\xi \\ &\qquad +\|G_q\|_\infty \int_a^b |u_{q+h}k|\,dx \int_a^b |u_{q+h} - u_q||h|\,d\xi \\ &\qquad +\|G_q\|_\infty \int_a^b |u_q h|\,dx \int_a^b |u_{q+h} - u_q||k|\,dx \\ &\leq \|G_q\|_\infty \|u_q\|_\infty \|k\|_1 \|u_{q+h} - u_q\|_\infty \|h\|_1 + \|G_q\|_\infty \|u_{q+h}\|_\infty \|k\|_1 \|u_{q+h} - u_q\|_\infty \|h\|_1 \\ &\qquad + \|G_q\|_\infty \|u_q\|_\infty \|h\|_1 \|u_{q+h} - u_q\|_\infty \|k\|_1. \end{aligned}$$

Then, applying Lemma 3.1.2 yields $|J(h)| \leq 8\|G_q\|_\infty^2 \|u_q\|_\infty^2 \|h\|_1^2 \|k\|_1$, given that (3.1.7) holds.

Hence

$$H''(q)[h,h] = 2\int_a^b \left(\int_a^b G_q(x,\xi)u_q(\xi)h(\xi)\,d\xi\right) u_q(x)h(x)\,dx = 2(A_q^{-1}(u_q h), u_q h),$$

which is (3.1.16). Finally, notice that $H''(q)[h,h] = 2(y, A_q y)$, where $A_q y = u_q h$. As A_q is positive, y is the trivial solution if and only if $u_q h$ (and hence h) is the zero function, and thus we have that, for each q in $\mathcal{D}$, $H''(q)$ is a positive definite form. ∎

The next proposition summarizes some of the more useful properties of H.

Proposition 3.1.5 *The functional H has the following properties.*

(1) $H(q) \geq 0$ for q in $\mathcal{D}$, and $H(q) = 0$ if and only if $q = Q$.

(2) H is strictly convex on $\mathcal{D}$.

(3) For any q in $\mathcal{D}$,

$$H(q) = \int_a^b [(u' - u_q')^2 + q(u - u_q)^2]\,dx. \tag{3.1.19}$$

(4) For any q_1, q_2 in $\mathcal{D}$,

$$H(q_1) - H(q_2) = \int_a^b (q_1 - q_2)(u^2 - u_{q_1}u_{q_2})\,dx. \tag{3.1.20}$$

Proof. To see that the first property holds, note that by Dirichlet's principle applied to the boundary value problem (3.0.2)–(3.0.3),

$$\int_a^b (u'^2 + qu^2)\,dx \geq \int_a^b (u_q'^2 + qu_q^2)\,dx$$

with equality if and only if $u = u_q$. Hence $H(q) > 0$ for all q in $\mathcal{D} - \{Q\}$ and $H(Q) = 0$. The strict convexity follows from the fact that $H''(q)$ is positive definite for each q in $\mathcal{D}$.

To see that property (3) holds, note that, on rearranging (3.0.1),

$$\begin{aligned} H(q) \quad = \quad & \int_a^b (u'^2 - 2u'u_q' + u_q'^2)\,dx + \int_a^b q(u^2 - 2uu_q + u_q^2)\,dx \\ & +2\int_a^b [(u' - u_q')u_q' + q(u - u_q)u_q]\,dx. \end{aligned}$$

Further, integration by parts shows that

$$\int_a^b [(u' - u_q')u_q' + q(u - u_q)u_q]\,dx = (u - u_q)u_q'|_a^b + \int_a^b (-u_q'' + qu_q)(u - u_q)\,dx,$$

which is clearly zero, as u and u_q agree at the endpoints of $[a, b]$ and u_q is a solution of (3.0.2); property (3) follows.

To verify property (4), we observe that after applying the definition of H and rearranging terms

$$\begin{aligned} H(q_1) - H(q_2) &= \int_a^b (u_{q_2}'^2 + q_1 u_{q_2}^2)\,dx - \int_a^b (u_{q_1}'^2 + q_1 u_{q_1}^2)\,dx + \int_a^b (u^2 - u_{q_2}^2)(q_1 - q_2)\,dx \\ &= \int_a^b [(u_{q_1}' - u_{q_2}')^2 + q_1(u_{q_1} - u_{q_2})^2]\,dx + \int_a^b (u^2 - u_{q_2}^2)(q_1 - q_2)\,dx \\ &= \int_a^b [-(u_{q_1}'' - u_{q_2}'') + q_1(u_{q_1} - u_{q_2})](u_{q_1} - u_{q_2})\,dx \\ &\qquad + \int_a^b (u^2 - u_{q_2}^2)(q_1 - q_2)\,dx, \end{aligned}$$

by a computation similar to that used to prove property (3) above, as u_{q_1} and u_{q_2} agree on the boundary of $[a, b]$. Replacing u_{q_1}'' and u_{q_2}'' by $q_1 u_{q_1}$ and $q_2 u_{q_2}$, respectively, and combining like terms, completes the proof.

3.2 The Variational Algorithm

Here we discuss one approach to the problem of minimizing the functional H, and how one estimates u' once a suitable approximation for this minimizer is found. As mentioned previously, our optimization strategy makes use of a steepest descent procedure.

First choose some initial function q_0 in $\mathcal{D}$ satisfying $\|q_0\|_\infty \leq M$, where M is described by (3.1.6). Then the $\mathcal{L}^2$ direction of steepest descent for H at q_0 is $-\nabla H(q_0) = -(u^2 - u_{q_0}^2)$. However, there are numerical problems associated with using the $\mathcal{L}^2$ gradient in the descent procedure stemming from the fact that the $\mathcal{L}^2$ gradient is always zero on the boundary of $[a, b]$. So instead we use the $\mathcal{H}^1$ gradient, $g_0 = \nabla_{\mathcal{H}^1} H(q_0)$, of H at q_0 given by

$$H'(q_0)h = (g_0, h)_1 \qquad (3.2.21)$$

for all h in $\mathcal{H}^1$. It is not difficult to show that g_0 is the solution of the Neumann boundary value problem

$$\begin{aligned} -v'' + v &= \nabla H(q_0) \\ v'(a) &= v'(b) = 0; \end{aligned} \tag{3.2.22}$$

in particular, one can see from equation (3.2.22) that the $\mathcal{H}^1$ gradient is just a preconditioned (or, smoothed) $\mathcal{L}^2$ gradient. The use of the $\mathcal{H}^1$ gradient for this purpose was originally suggested by Neuberger [35]. Note that the function $f_0(\alpha) = H(q_0 - \alpha g_0)$ is strictly decreasing in some neighborhood of $\alpha = 0$ as $f_0'(0) = -\|g_0\|^2_{\mathcal{H}^1} < 0$. This function is "minimized" by using the quadratic approximation

$$f_0(\alpha) \approx H(q_0) - \alpha H'(q_0)g_0 + \frac{1}{2}\alpha^2 H''(q_0)[g_0, g_0].$$

Here the minimizing value for α is given by $\alpha_0 = H'(q_0)g_0/H''(q_0)[g_0, g_0]$, where

$$H'(q_0)g_0 = \int_a^b (u^2 - u_{q_0}^2)g_0\, dx$$

and

$$H''(q_0)[g_0, g_0] = 2(A_{q_0}^{-1}(u_{q_0}g_0), u_{q_0}g_0) = 2\int_a^b w u_{q_0} g_0\, dx,$$

with w being the solution of the Dirichlet problem

$$\begin{aligned} -w'' + q_0 w &= u_{q_0} g_0 \\ w(a) &= w(b) = 0. \end{aligned}$$

Then we set $q_1 = q_0 - \alpha_0 g_0$. Invariant embedding [20, p. 117] allows us to convert the boundary value problem

$$-v'' + q_1 v = 0 \tag{3.2.23}$$

$$v(a) = A, \quad v(b) = B \tag{3.2.24}$$

to a pair of first order initial value problems. We proceed as follows.

Taking $z = v'$, the boundary value problem (3.2.23)–(3.2.24) can be rewritten as the system

$$v' = z \tag{3.2.25}$$

$$z' = q_1 v \tag{3.2.26}$$

$$v(a) = A, \quad v(b) = B. \tag{3.2.27}$$

We then introduce new functions $t(x)$ and $w(x)$ that satisfy

$$v + tz = w. \tag{3.2.28}$$

Eliminating v and z from equations (3.2.25), (3.2.26), and (3.2.28) yields

$$(w' - q_1 tw) - z(t' + 1 - q_1 t^2) = 0. \tag{3.2.29}$$

Now (3.2.29) will be satisfied if t and w solve the initial value problem

$$t' = q_1 t^2 - 1 \tag{3.2.30}$$

$$w' = q_1 tw \tag{3.2.31}$$

$$t(a) = 0, \quad w(a) = A. \tag{3.2.32}$$

Once t and w are found as solutions for the system (3.2.30)–(3.2.31)–(3.2.32) we see that $v(b) + t(b)z(b) = w(b)$. Hence if we set $\delta = (w(b) - B)/t(b)$, then v and z must satisfy the system (3.2.25)–(3.2.26) together with the initial conditions $v(b) = B$ and $z(b) = \delta$. This gives our first approximation:

$$u_{q_1} \approx u, \quad u'_{q_1} \approx u'. \tag{3.2.33}$$

This procedure is repeated with q_n replaced by $q_{n+1} = q_n - \alpha_n g_n$, for $n = 1, 2, \ldots$, where $g_n = \nabla_{\mathcal{H}^1} H(q_n)$ and α_n is chosen to "minimize" $f_n(\alpha) = H(q_n - \alpha g_n)$ in the manner described above, until H fails to descend. We use (3.1.20) to check the descent.

3.3 Stability

Typically, a function that one might wish to differentiate numerically is given with a certain amount of error. For this reason, it is necessary to examine how small perturbations in u, the function to be differentiated, affect the computed value of the derivative.

Let $\tilde{u} = u + \Delta$ where Δ is small in $\mathcal{L}^1$ and where $\Delta(a) = \Delta(b) = 0$. It is known that, with no further restrictions on Δ, $\|u' - \tilde{u}'\|_2$ can be arbitrarily large. We will show that given certain conditions on $\tilde{u}''$, having Δ small in $\mathcal{L}^1$ will imply that $\tilde{u}'$ is close to u' in $\mathcal{L}^2$.

Lemma 3.3.1 *Suppose that q is a function in $\mathcal{L}^1$ such that*

$$\|q\|_\infty \leq M, \tag{3.3.34}$$

where M satisfies (3.1.6). Then

$$\|u_q\|_\infty \leq \max(A, B)\left\{1 + \frac{M(b-a)^{1/2}}{\frac{\pi^2}{(b-a)^2} - M}\right\}. \tag{3.3.35}$$

Proof. Clearly, as $-M \leq q(x)$ almost everywhere in $[a, b]$, from [7, p. 20] q is in $\mathcal{D}$ and, if the smallest eigenvalues of A_q and A_{-M} are denoted by λ_q and λ_{-M}, respectively, then we have that $\lambda_{-M} \leq \lambda_q$ where

$$\lambda_{-M} = \left[\frac{\pi^2}{(b-a)^2} - M\right]. \tag{3.3.36}$$

In consequence,

$$\|A_q^{-1}\|_2 \leq \|A_{-M}^{-1}\|_2 = \left[\frac{\pi^2}{(b-a)^2} - M\right]^{-1}, \tag{3.3.37}$$

where $\|A_q^{-1}\|_2$ is the operator norm of A_q^{-1} defined by $\|A_q^{-1}\|_2 = \sup_{\|h\|_2=1} \|A_q^{-1}h\|_2$.

Now let $l(x) = A + (B - A)(x - a)/(b - a)$ and $v = u_q - l$. It follows that v is in the domain of A_q and $-v'' + qv = -ql$. Hence $v = -A_q^{-1}(ql)$, i.e., $u_q = l - A_q^{-1}(ql)$. If we now apply the Hölder inequality and make use of (3.3.34) and (3.3.37), we have that

$$\begin{aligned} |u_q(x)| &\leq \|l\|_\infty + \|A_q^{-1}\|_2\|ql\|_2 \\ &\leq \|l\|_\infty\{1 + M\|A_{-M}^{-1}\|_2(b-a)^{1/2}\} \end{aligned}$$

$$= \max(A,B)\left\{1+\frac{M(b-a)^{1/2}}{\frac{\pi^2}{(b-a)^2}-M}\right\}.$$

Lemma 3.3.2 *Suppose that q is an $\mathcal{L}^1$ function that satisfies (3.3.34). Then*

$$H(q) \geq \tau \|u-u_q\|^2_{\mathcal{H}^1}$$

where

$$\tau = \frac{\frac{\pi^2}{(b-a)^2}-M}{1+\frac{\pi^2}{(b-a)^2}} > 0, \tag{3.3.38}$$

whenever M satisfies (3.1.6).

Proof. Let h be in $\mathcal{H}^1$ with $h(a) = h(b) = 0$ and observe that by (3.3.34) we have

$$\int_a^b (h'^2 + qh^2)dx \geq \int_a^b (h'^2 - Mh^2)\,dx. \tag{3.3.39}$$

Observe also that

$$\int_a^b (h'^2 - Mh^2)\,dx \geq \lambda_{-M} \int_a^b h^2\,dx,$$

where $\lambda_{-M} > 0$ is the smallest eigenvalue of A_{-M}. By (3.1.6), it follows that $0 < \tau < 1$. As $(1-\tau)\lambda_{-M} - M\tau = \tau$, from (3.3.36) and (3.3.38),

$$\begin{aligned}
\int_a^b (h'^2 - Mh^2)\,dx &= (1-\tau)\int_a^b (h'^2 - Mh^2)\,dx + \tau\int_a^b (h'^2 - Mh^2)\,dx \\
&\geq (1-\tau)\lambda_{-M}\int_a^b h^2\,dx + \tau\int_a^b h'^2\,dx - \tau M\int_a^b h^2\,dx \\
&= \tau\int_a^b h'^2\,dx + [(1-\tau)\lambda_{-M} - M\tau]\int_a^b h^2\,dx \\
&= \tau\int_a^b (h'^2 + h^2)\,dx.
\end{aligned} \tag{3.3.40}$$

Combining (3.3.39) and (3.3.40) and replacing h by $u - u_q$ yields, via (3.1.19), the desired result.

Theorem 3.3.3 *Let $\tilde{u}$ be a positive twice differentiable function and let $\tilde{Q}$ be given by $\tilde{Q} = \tilde{u}''/\tilde{u}$. Suppose that $\tilde{Q}$ satisfies (3.3.34) and assume that $\tilde{u}(a) = A$ and $\tilde{u}(b) = B$. Then*

$$\|u-\tilde{u}\|^2_{\mathcal{H}^1} \leq k_2\|u-\tilde{u}\|_1$$

where

$$k_2 = 2M\|u\|_\infty \frac{1 + \frac{\pi^2}{(b-a)^2}}{\frac{\pi^2}{(b-a)^2} - M}.$$

Proof. As in the previous lemma $\tilde{Q}$ is in $\mathcal{D}$. So, making use of (3.1.20) we have that

$$H(\tilde{Q}) = \int_a^b (\tilde{Q} - Q)u(u - \tilde{u})\, dx \tag{3.3.41}$$

as $H(Q) = 0$. Applying Hölder's inequality to (3.3.41) gives

$$H(\tilde{Q}) \leq \|\tilde{Q} - Q\|_\infty \|u\|_\infty \|u - \tilde{u}\|_1 \leq 2M\|u\|_\infty \|u - \tilde{u}\|_1. \tag{3.3.42}$$

Combining this with the result of Lemma 3.3.2 proves the theorem. ∎

This theorem says that if we restrict our attention to the class of functions q that satisfy (3.3.34), then the problem of numerical differentiation becomes well-posed. The condition (3.3.34) serves to control the curvature of any perturbation that may be introduced into the function. This result is similar in spirit to the (optimal) stability result in [46].

3.4 Convergence

In this section we prove that any sequence $\{q_n\}$ in $\mathcal{D}$ such that (3.3.34) holds and such that $\{H(q_n)\}$ tends to zero must converge weakly to Q in $\mathcal{L}^2$. Further, the sequence $\{u_{q_n}\}$ produced by $\{q_n\}$ must converge strongly to u in $\mathcal{H}^1$. We then show that under certain conditions that can be monitored numerically, the sequence $\{q_n\}$ produced by the steepest descent algorithm described in §3.2 has the property that $\{H(q_n)\}$ tends to zero.

Theorem 3.4.1 *Suppose that $\{q_n\}$ is any sequence of functions in $\mathcal{D}$ satisfying (3.3.34). If $\{H(q_n)\}$ tends to zero, then $\{q_n\}$ converges weakly to Q in $\mathcal{L}^2$ and $\{u_{q_n}\}$ converges strongly to u in $\mathcal{H}^1$.*

Proof. That $\{u_{q_n}\}$ tends to u in $\mathcal{H}^1$ follows trivially from Lemma 3.3.2. To see that $\{q_n\}$ converges weakly to Q in $\mathcal{L}^2$, let h be in the domain of the self-adjoint operator A_Q. In other words, h' is absolutely continuous, $-h'' + Qh$ is in $\mathcal{L}^2$, and $h(a) = h(b) = 0$. It can be easily shown that $k = h/u$ is also in the domain of A_Q. So, following the same line of reasoning used to get (3.1.10), we have that

$$A_Q(u - u_{q_n}) = (q_n - Q)u_{q_n} = (q_n - Q)u + (q_n - Q)(u_{q_n} - u). \tag{3.4.43}$$

Multiplying through (3.4.43) by k and integrating by parts yields

$$\int_a^b [(u' - u'_{q_n})k' + Q(u - u_{q_n})k]\,dx = \int_a^b (q_n - Q)h\,dx + \int_a^b (q_n - Q)(u_{q_n} - u)k\,dx.$$

Using (3.1.5) and setting $C = \max(1, M)$ we have,

$$\begin{aligned} \left|\int_a^b (q_n - Q)h\,dx\right| &\leq \left|\int_a^b [(u' - u'_{q_n})k' + Q(u - u_{q_n})k]\,dx\right| + \left|\int_a^b (q_n - Q)(u_{q_n} - u)k\,dx\right| \\ &\leq C\|u - u_{q_n}\|_{\mathcal{H}^1}\|k\|_{\mathcal{H}^1} + 2M\|u_{q_n} - u\|_2\|k\|_2 \end{aligned} \tag{3.4.44}$$

that tends to zero as n tends to infinity.

Now let h be any function in $\mathcal{L}^2$ and let $\varepsilon > 0$ be given. By the density of the domain of A_Q in $\mathcal{L}^2$ there exists a sequence $\{h_j\}$ in the domain of A_Q that converges strongly to h in $\mathcal{L}^2$. So, choose j_0 sufficiently large that $\|h_{j_0} - h\|_2 \leq \frac{1}{4}\varepsilon\ M^{-1}(b-a)^{-1/2}$. Then, fixing $h = h_{j_0}$ in (3.4.44), one may choose N large enough to insure that

$$\left|\int_a^b (q_n - Q)h_{j_0}\,dx\right| < \frac{\varepsilon}{2}$$

for $n \geq N$. Consequently, for $n \geq N$

$$\begin{aligned} \left|\int_a^b (q_n - Q)h\,dx\right| &\leq \left|\int_a^b (q_n - Q)(h - h_{j_0})\,dx\right| + \left|\int_a^b (q_n - Q)h_{j_0}\,dx\right| \\ &\leq \|q_n - Q\|_2\|h - h_{j_0}\|_2 + \varepsilon/2 \\ &\leq 2M(b-a)^{1/2}\|h - h_{j_0}\|_2 + \varepsilon/2 \\ &\leq \varepsilon/2 + \varepsilon/2 = \varepsilon. \end{aligned}$$

Lemma 3.4.2 *If the sequence $\{q_n\}$ from the algorithm in §3.2 tends to a function q_* in $\mathcal{L}^2$, then the sequence of $\mathcal{H}^1$ gradients of H at q_n must converge to zero in $\mathcal{H}^1$.*

Proof. Proceeding as in Lemma 3.1.2 we have that $\{u_{q_n}\}$ tends uniformly to u_{q_*} as $\{q_n\}$ tends to q_* in $\mathcal{L}^2$. Hence $\{u_{q_*}^2 - u_{q_n}^2\}$ converges uniformly to zero. Now, setting $g_n = \nabla_{\mathcal{H}^1} H(q_n)$

and $w_n = g_n - g_*$ we have that w_n satisfies the boundary value problem

$$-v'' + v = u_{q_*}^2 - u_{q_n}^2 \qquad (3.4.45)$$
$$v'(a) = v'(b) = 0;$$

i.e.,

$$w_n(x) = \int_a^b K(x,\xi)(u_{q_*}^2(x) - u_{q_n}^2(\xi))\, d\xi,$$

where $K(x,\xi)$ is the Green's function associated with (3.4.45). So

$$\|w_n\|_\infty \le \|K\|_\infty \|u_{q_*}^2 - u_{q_n}^2\|_1 \le (b-a)\|K\|_\infty \|u_{q_*}^2 - u_{q_n}^2\|_\infty. \qquad (3.4.46)$$

It follows from (3.4.45) and (3.4.46) that $\{w_n''\}$ tends uniformly to zero. We then have that

$$\|w_n'\|_\infty = \left\| \int_a^x w_n''\, d\xi \right\|_\infty \le \int_a^b \|w_n''\|_\infty\, d\xi = (b-a)\|w_n''\|_\infty, \qquad (3.4.47)$$

and, from (3.4.46) and (3.4.47) it easily follows that $\{g_n\}$ tends strongly to g_* in $\mathcal{H}^1$. Finally, note that in the case of steepest descent with exact line search,

$$(g_n, g_{n+1})_1 = H'(q_{n+1})g_n = -\frac{d}{d\alpha}H(q_n - \alpha g_n)\Big|_{\alpha=\alpha_n} = 0 \qquad (3.4.48)$$

for all $n \ge 0$, as α_n is a minimum for $f_n(\alpha) = H(q_n - \alpha g_n)$.

Suppose now that $g_* \neq 0$. We can assume without loss of generality that $g_n \neq 0$ for all n. Arguing as in [11], set $z_n = g_n/\|g_n\|_{\mathcal{H}^1}$ for each n, set $z_* = g_*/\|g_*\|_{\mathcal{H}^1}$ and note that $\|z_n\|_{\mathcal{H}^1} = \|z_*\|_{\mathcal{H}^1} = 1$. By the earlier argument $\{g_n\}$ tends strongly to g_* in $\mathcal{H}^1$, hence $\{\|g_n\|_{\mathcal{H}^1}\}$ converges to $\|g_*\|_{\mathcal{H}^1}$. It is then a simple matter to show that $\{z_n\}$ converges strongly to z_* in $\mathcal{H}^1$. So choose N such that for $n > m > N$

$$\|z_n - z_m\|_{\mathcal{H}^1}^2 < 1. \qquad (3.4.49)$$

However, by (3.4.48) $\|z_n - z_{n+1}\|_{\mathcal{H}^1}^2 = \|z_n\|_{\mathcal{H}^1}^2 + \|z_{n+1}\|_{\mathcal{H}^1}^2 = 2.$ for any n. This contradicts (3.4.49), and in consequence $g_* = 0$. ∎

For the case of inexact line search the requirement (3.4.48) on α_n can be weakened to

$$|(g_n, g_{n+1})| \leq \frac{1}{2}||g_n||_{\mathcal{H}^1}||g_{n+1}||_{\mathcal{H}^1}$$

by a trivial modification of this proof.

Theorem 3.4.3 *Let $\{q_n\}$ be the sequence generated by the algorithm discussed in §3.2. Assume that each function q_n satisfies (3.3.34) and that there exists a constant $\rho > 0$ such that*

$$H(q_n) - H(q_{n+1}) \geq \alpha_n \rho ||g_n||_{\mathcal{H}^1}||g_n||_{\mathcal{L}^2} \tag{3.4.50}$$

for $n \geq 0$ where $\{g_n\}$ and $\{\alpha_n\}$ are as described in §3.2. Then the sequence $\{H(q_n)\}$ converges to zero as n tends to infinity.

Proof. First we claim that there is a subsequence $\{g_{\varphi(n)}\}$ tending strongly to zero in $\mathcal{H}^1$. To see this, assume by way of contradiction that there is a number $\delta > 0$ such that

$$||g_n||_{\mathcal{H}^1} \geq \delta \tag{3.4.51}$$

for all n. Then from (3.4.50) and (3.4.51)

$$H(q_n) - H(q_{n+1}) \geq \rho||g_n||_{\mathcal{H}^1}||q_{n+1} - q_n||_{\mathcal{L}^2} \geq \rho\delta||q_{n+1} - q_n||_{\mathcal{L}^2},$$

for all n. Hence for $n > r$

$$\begin{aligned} H(q_n) - H(q_{n+1}) &\geq \rho\delta||q_{n+1} - q_n||_{\mathcal{L}^2}, \\ H(q_{n-1}) - H(q_n) &\geq \rho\delta||q_n - q_{n-1}||_{\mathcal{L}^2}, \\ &\vdots \\ H(q_{r+1}) - H(q_{r+2}) &\geq \rho\delta||q_{r+2} - q_{r+1}||_{\mathcal{L}^2}. \end{aligned}$$

The triangle inequality then implies

$$H(q_{r+1}) - H(q_{n+1}) \geq \rho\delta \sum_{i=r+1}^{n} ||q_{i+1} - q_i||_{\mathcal{L}^2} \geq \rho\delta||q_{r+1} - q_{n+1}||_{\mathcal{L}^2}.$$

Now $\{H(q_n)\}$ is convergent as it is monotonically decreasing and bounded below; hence $\{q_n\}$ is Cauchy in $\mathcal{L}^2$ and must converge strongly to some function q_* in $\mathcal{L}^2$. So by Lemma 3.4.2, the sequence $\{g_n\}$ converges to zero strongly in $\mathcal{H}^1$, and this contradicts (3.4.51). It follows that we can find a subsequence $\{g_{\varphi(n)}\}$ converging to zero strongly in $\mathcal{H}^1$. In addition, as the sequence $\{q_{\varphi(n)}\}$ is bounded in $\mathcal{L}^2$ we can assume that a subsequence, $\{q_{\psi(n)}\}$, of $\{q_{\varphi(n)}\}$ converges weakly to some function $\tilde{q}$ in $\mathcal{L}^2$ [16, p. 68, Theorem II.3.28]. Note that, as $q_{\varphi(n)}$ is in $\mathcal{D}$ and satisfies (3.3.34) for each n, we have that for all functions v in $\mathcal{H}^1$

$$\int_a^b (v'^2 + q_{\varphi(n)}v^2)\,dx \geq \lambda_{-M}\int_a^b v^2\,dx, \tag{3.4.52}$$

where λ_{-M} denotes the smallest eigenvalue of the operator A_{-M}. As $q_{\varphi(n)}$ converges weakly to $\tilde{q}$ in $\mathcal{L}^2$, it follows that (3.4.52) holds with $\tilde{q}$ replacing $q_{\varphi(n)}$ for all v in $\mathcal{H}^1$. Consequently, $\tilde{q} \in \mathcal{D}$. Let $\tilde{g} = \nabla_{\mathcal{H}^1}H(\tilde{q})$ and note that $\{g_{\psi(n)}\}$ tends strongly to zero in $\mathcal{H}^1$.

We now claim that $\{g_{\psi(n)}\}$ converges strongly to $\tilde{g}$ in $\mathcal{H}^1$. To see this, first note that by an argument similar to the one used to obtain (3.1.10) it can be shown that $\{u_{\psi(n)}\}$ tends to $u_{\tilde{q}}$ in $\mathcal{H}^1$, hence in $\mathcal{L}^2$ as well. So setting $\tilde{w}_n = g_n - \tilde{g}$ and proceeding along the line of Lemma 3.4.2, we have that $\|\tilde{w}_{\psi(n)}\|_\infty \leq \|\tilde{K}\|_\infty \cdot 2k_1\|u^2_{q_{\psi(n)}} - u^2_{\tilde{q}}\|_1$, where k_1 is the right hand side of (3.3.35) and $\tilde{K}$ is the Green's function associated with the Neumann problem

$$-v'' + v = u^2_{\tilde{q}} - u^2_{q_n}$$
$$v'(a) = v'(b) = 0.$$

Again appealing to the proof of Lemma 3.4.2 it can easily be shown that $\{\tilde{w}_{\psi(n)}\}$ tends strongly to zero in $\mathcal{H}^1$. In other words, $\{g_{\psi(n)}\}$ is strongly convergent to $\tilde{g}$ in $\mathcal{H}^1$.

Now $\{g_{\psi(n)}\}$ tends to $\tilde{g}$ and to zero in $\mathcal{H}^1$, hence $\tilde{g} = 0$. As H is strictly convex, there can be at most one stationary point; it follows that $\tilde{q} = Q$. Also as H is twice Fréchet differentiable in $\mathcal{L}^2$ it must be weakly sequentially continuous in $\mathcal{L}^2$ [52, p. 236, Corollary 41.9]; hence $\{H(q_{\psi(n)})\}$ must converge to $H(Q)$, which is zero. Finally, as $\{H(q_n)\}$ is strictly decreasing, it must tend to zero. ∎

For numerical purposes $||g_n||^2_{\mathcal{H}^1}$ can be easily computed via an integration by parts, on noting that $-g_n'' + g_n = \nabla H(q_n)$:

$$||g_n||^2_{\mathcal{H}^1} = \int_a^b (g_n'^2 + g_n^2)\, dx = \int_a^b g_n(u^2 - u_{q_n}^2)\, dx.$$

Also, (3.4.50) is a condition similar in spirit to one originally suggested by Goldstein [25] ensuring convergence of gradients to zero.

Chapter 4

Implementation and Numerical Results

A program to test the numerical viability of the method was written in FORTRAN 77 and run on a SUN 4 work-station. The function $f(x) = \cos(x)$ was differentiated under two sets of circumstances. The first experiment was designed to study the effect of roundoff error when a small stepsize, h, is used in the discretization. The other experiment was designed to examine the stability of the method when the given function contains a random error. All computations were done in single precision. The results were then compared with derivative values obtained using central differences.

All boundary value problems were solved by the method of invariant embedding (see, for example, [20, p. 117]). This allowed us to use initial value solvers, which tend to be more accurate than standard boundary value solvers in this situation. This was especially important in the first experiment as standard boundary solvers for these equations make use of formulae that involve factors of h^2 which would be quite disastrous in this context. Also, invariant embedding seems to handle nicely the stiffness that typically occurs in disconjugate Sturm-Liouville boundary value problems. The initial value problems that resulted were solved using a fourth order Runge-Kutta routine (presumably, one might do better with more accurate predictor-corrector routines, but we have not tried this as yet). Our initial estimate for Q was always taken to be $q_0 = 0$.

The effect of taking a small stepsize is known to severely amplify the effects of roundoff error especially when the function to be differentiated is large compared to the value of the derivative [44, p. 145]. In addition, the computed value of the derivative, in the case of sufficiently small stepsize, will be zero even when the correct value of the derivative is relatively large; for example, see [44, Table 5.1]. Therefore, it is instructive to examine the computed derivative of f at a value of x close, but not equal, to the stationary point $x = 0$. The results of such an experiment are given in Table 4.0.1. The function f is

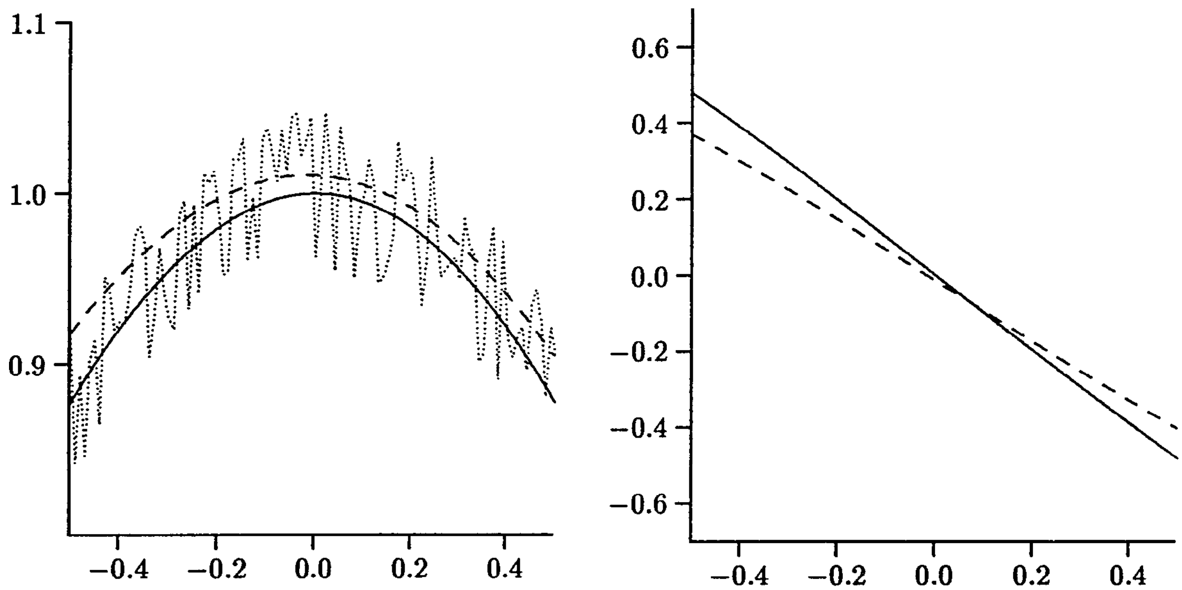
Table 4.0.1: Relative error in computation of $f'(0.001)$, $f(x) = \cos x$

h	central difference	variational derivative	# subintervals	iterations
10^{-3}	0.0133	0.0004	100	3
10^{-4}	0.106	0.0013	100	6
10^{-5}	1.98	0.0729	500	1
10^{-6}	1.00	0.2847	500	2
10^{-7}	1.00	0.404	1000	1

discretized on an interval with left endpoint $x = 0.001$, with the grid points evenly spaced at a distance h apart, and using the number of subintervals shown in the table. The derivative was calculated using the variational method discussed here and compared with values obtained using the standard central difference formula. The large relative errors for the central difference method with $h \leq 10^{-5}$ indicate a complete breakdown of this method. At present, the optimization routine being used (steepest descent with exact line search) is not overly sophisticated. Nonetheless it can be seen from the table that the performance is close to optimal (in some sense), as the machine on which the computations were performed makes use of a 23 digit binary mantissa in single precision mode. As $2^{-23} \approx 1.2 \times 10^{-7}$ the poor (but not disastrous) results for $h \leq 10^{-6}$ are probably as good as one could expect. The method failed completely when $h = 0.9 \times 10^{-7}$, giving $f' = 0$. In general the results were marginally better when $x = 0.001$ was the left hand end-point of the interval compared to being the right hand end-point, and as h decreased, it was necessary to use more subintervals to retain optimal accuracy. Finally, it should be noted that for $h \leq 10^{-6}$ it was sometimes necessary to replace the factor α_n obtained via the exact linesearch formula of §3.2 by $\alpha_n/2$, or even $\alpha_n/10$ in some cases, to guarantee that $H(q_n) - H(q_{n+1}) > 0$ at that step. Presumably, one might expect some improvements from a more detailed study of the optimization involving, say, the use of conjugate gradient, or DFP, or BFGS techniques, possibly with an appropriate approximate line search algorithm, but this has not been done as yet.

Next we examined the effect that a small random perturbation has on the variationally computed derivative. Letting $g(x) = \cos(x) + \varepsilon(x)$ for x in $[-0.5, 0.5]$, where the values of $\varepsilon(x)$ were randomly chosen such that $|\varepsilon(x)| \leq 0.05$, we used the variational algorithm to numerically differentiate g. As before, the grid points are evenly spaced. In this case, we took 100 subintervals, hence $h = 10^{-2}$. The results are summarized in Figure 4.0.1. Here, the solid curves represent the function $\cos x$ and its derivative, and the dashed curves

Figure 4.0.1: Numerical differentiation of $\cos x + \epsilon(x)$, $|\epsilon(x)| \leq 0.05$



represent the variational approximations to the function g and its derivative. It can be seen from this figure that the error in the variationally computed derivative is of the same order of magnitude as the error in the given function. Work done in [46] suggests that is somewhat optimal.

In practice, the computed value of u_q seems to be a good $\mathcal{L}^2$ approximation for g, which has bounded curvature. This suggests that u_q should be a reasonable estimate for the unperturbed function f. The data summarized in Figure 4.0.1 supports this idea.

An observation we have made through the course of many such experiments is that this variational method seems to work best on a small interval. This is in agreement with the theory in that an examination of (3.1.6) shows that the estimated function q_n is more likely to be in the set $\mathcal{D}$ (and to satisfy (3.1.5)) when $b - a$ is small. This is a potentially useful fact, as the variational method is highly parallelizable in that the problem of computing u'

on the interval $[a, b]$ may usually be replaced by that of numerically differentiating u on a number of small subintervals.

It is worth noting also that the addition of a larger constant to the function seems to have an effect similar to that of making the interval small. Taking the constant too large, however, causes information about the function to be lost if computer precision is limited.

Finally, as one should perhaps expect, if the random error is made larger, problems with convergence and/or instability tend to occur. This happened only occasionally with a 10% relative error and more so at higher relative error levels. In some of the more delicate cases, we resorted to using "continuous" steepest descent in which, if the α_n obtained from "exact" line-search did not cause a decrease in H, then α_n was replaced with $c\alpha_n$ for some c, $0 < c < 1$ (usually, $c = 0.5$, or 0.1). In general, the convergence was poor for functions with more than one or two "bends" in the curve, or for functions defined at too few points. The latter deficiency might possibly be improved with a more careful study of which quadrature formula is appropriate for use in computing the factors α_n (we used Simpson's formula), or by using an inexact linesearch, but this remains to be seen. The former deficiency is probably not a problem in practice, as one can usually divide the problem into sub-problems over smaller intervals, as observed above. It may also be less of an issue if one uses a higher order version of the method as described below. In any event, in these extreme cases, the methods used in [9], appear to be more accurate, at least when the exact form of the "base" function and the perturbation function ($\cos x$ and $\epsilon(x)$, respectively, in our case) are known.

Chapter 5

Generalizations

5.1 The Fourth Order Version of the Variational Method

In this section we outline how the method we have discussed might be adjusted to improve convergence. As most of the proofs are exactly the same as their corresponding second order counterparts, we restrict this discussion to the most important differences and similarities between the second order and the fourth order methods. In addition, we borrow most of the notation used earlier to help emphasize the similarities.

As before, u is the function that we wish to differentiate numerically. We assume that u is four times differentiable on $[a, b]$ with $u^{(4)}$ in $\mathcal{L}^\infty$. We assume the existence of a constant $c > 0$ such that $u(x) \geq c > 0$ for $a \leq x \leq b$, and set $Q = -u^{(4)}/u$. As discussed in §§1.3 and 3.1, we have that the equation $v^{(4)} + Qv = 0$ is disconjugate on $[a, b]$ and we define $\mathcal{D}$ to be the set of all functions q in $\mathcal{L}^1$ such that

$$v^{(4)} + qv = 0 \tag{5.1.1}$$

is disconjugate on $[a, b]$. As in the earlier case, $\mathcal{D}$ is convex and open in $\mathcal{L}^1$ and in $\mathcal{L}^2$. Set

$$A = u(a),\ B = u(b),\ A' = u'(a),\ B' = u'(b). \tag{5.1.2}$$

Then, for q in $\mathcal{D}$, there exists a unique solution to (5.1.1) subject to (5.1.2). We denote this solution by u_q. The functional is now given by

$$H(q) = \int_a^b (u''^2 + qu^2)\,dx - \int_a^b (u_q''^2 + qu_q^2)\,dx$$

for q in $\mathcal{D}$. As before, we have that $H(q) \geq 0$ for all q in $\mathcal{D}$ with equality if and only if $q = Q$.

Now let A_q be the operator given by $A_q v = v^{(4)} + qv$ for functions v in $\mathcal{H}^4$ satisfying

$$v(a) = v(b) = v'(a) = v'(b) = 0.$$

Noting that $q = 0$ is in $\mathcal{D}$, let λ_0 be the smallest eigenvalue of A_0, and take $0 < M < \lambda_0$. Observe that all functions q satisfying

$$\|q\|_\infty \leq M \tag{5.1.3}$$

must be in $\mathcal{D}$. Further we can assume without loss of generality that Q satisfies (5.1.3) by adding an appropriate constant to u, if necessary.

It turns out that the derivative results are similar to those obtained for the second order case. The Fréchet differential of H is given by equation (3.1.13) and the second Fréchet differential of H is given by equation (3.1.16) for each q in $\mathcal{D}$ and h in $\mathcal{L}^1$ where u_q is the solution to (5.1.1) subject to (5.1.2). Also the $\mathcal{L}^2$ gradient of H is given by (3.1.14). The main difference is that the function u_q is now the solution of a fourth order equation.

Nearly the same technique for steepest descent outlined in §3.2 can be applied to the fourth order case. The only real difference is that a fourth order equation must be solved at each step, and this requires that one have at hand the derivatives $u'(a)$ and $u'(b)$. These can be obtained by running the second order method first, in exact analogy with the procedure for using a lower order method to start a typical ODE multistep method. The expected benefit from this approach is indicated by the following theorem, which predicts that convergence for this method should be superior to convergence obtained by the second order method.

Theorem 5.1.1 *Under the assumptions (3.3.34) and (3.4.50), we have that $\{q_n\}$ tends weakly to Q in $\mathcal{L}^2$ and $\{u_{q_n}\}$ tends strongly to u in $\mathcal{H}^2$.*

Thus, if one is interested in applications to something like missile tracking, in which one is given (noisy) radar data on the position (and possibly the velocity) and one wishes to find the acceleration, then the computation of the second derivative becomes important. In theory, one can obtain the second derivative from the second order method by noting that, if $u_{q_n} \approx u$, then $u'' \approx u''_{q_n} = q_n\, u_{q_n}$. However, as the sequence $\{q_n\}$ converges only weakly in $\mathcal{L}^2$ (Theorem 3.4.1), this is not expected to be a good approximation, and our numerical experiments confirmed this to be so. On the implementation side, it is worth noting that in [3, p. 119] one can obtain a fourth order version of the invariant embedding equations that proved so effective in the second order case.

5.2 Data Smoothing

A large number of books and articles have been written on the subject of data smoothing (or curve/surface fitting). In contrast with the problem of interpolation in which one seeks a curve or surface passing through given data points, the problem of data smoothing dictates that one construct a curve or surface having certain properties (e.g. smoothness or bounded curvature) that in some way preserve the information contained in a cloud of data points.

An extensive range of methods have been suggested for solving such a problem. As data smoothing is clearly related to solving an overdetermined system of equations. One might use something like a non-linear least squares fit to a parametric model giving up much objectively in the process. At the other extreme, one could invoke "Draftsman license" and try to draw a smooth curve through the cloud of data points.

Alternately, one could also proceed as follows. Run the variational algorithm discussed in §3.2. Referring to Figure 4.0.1 again, the dashed curve represents the solution u_{q_n} of the Sturm-Liouville equation from which the numerical derivative is obtained. This function may be considered as a suitable candidate for a smoothed version of the original function. In effect, one is choosing one function (the one that minimizes 3.0.1) from a large class of smooth functions (i.e., the class of all solutions of Sturm-Liouville boundary value problems (1.2.7) subject to the boundary conditions $v(a) = A$ and $v(b) = B$ for which q is in $\mathcal{D}$).

In some sense this is comparable to the method mentioned above of fitting to a parametric model, but is probably more objective in that the effective number of parameters is equal to the number of data points, which is in general quite large.

The value of such an approach might lie in providing a reasonably objective comparison mechanism in a situation in which one wishes to use certain experimental data to support a particular theoretical viewpoint. In particular, one might fit to a model with a small number of parameters in line with the theory, and then compare this with the result obtained with the above variational approach. As a minimizing variational method will in general try to "settle down" to the minimum in an unbiased way, and only as far as the roughness in the data permits, reasonable agreement could then be interpreted as confirmation of the validity of the model. It is likely that this type of smoothing could also be carried out by means of higher order techniques discussed in §5.1.

Data smoothing can most likely be accomplished in higher dimensions using a multi-dimensional version of the variational algorithm discussed in §3.2. To illustrate how this might work we proceed along lines parallel to the one dimensional case.

Let Ω be a bounded open region in $\mathbf{R}^n$, $n \geq 2$, with a smooth boundary, and let u be in $C^2(\Omega)$ be a given positive function. Set $Q = \Delta u/u$ and $g = u|_{\partial\Omega}$. As before, one can think of u as the solution of the following Schrödinger-type boundary value problem

$$-\Delta v + qv = 0, \quad x \in \Omega \tag{5.2.4}$$

$$v|_{\partial\Omega} = g, \tag{5.2.5}$$

when $q = Q$. The inverse problem "given u, find Q " is, as before, equivalent to effecting differentiation on u. This, however, is not our main focus here, although it will arise naturally later.

We consider first the general framework needed for the later development. The analogue for the space $\mathcal{D}$ defined earlier may be obtained by observing that in the ODE case the function q is in the disconjugacy if and only if the associated Dirichlet operator A_q is positive in $\mathcal{L}^2$. So, in the present environment it is natural to define $\mathcal{D}$ to be the set of all

q in $\mathcal{L}^1(\Omega)$ for which the Dirichlet operator $A_q = -\Delta + q$ defined on the domain

$$\mathcal{D}(A_q) = \{u \in \mathcal{H}^2(\Omega) : u|_{\partial\Omega} = 0\} = \mathcal{H}_0^2(\Omega),$$

is a positive operator in $\mathcal{L}^2(\Omega)$. The set $\mathcal{D}$ is convex and open. The question arises then as to whether or not Q lies in $\mathcal{D}$. From [39, §C8] it is at least close to being true. However the question is effectively moot in that one can always add a suitably large constant to the function u to guarantee that Q is in $\mathcal{D}$. Finally notice that, under these conditions, the boundary value problem (5.2.4)–(5.2.5) has a unique solution, which we denote by u_q. It follows from the Maximum Principle for equation (5.2.4) that, for $||q||_\infty$ small enough, u_q is a positive function.

Define the functional H on $\mathcal{D}$ by

$$H(q) = \int_\Omega (|\nabla u|^2 + q(x)u^2)\,dx - \int_\Omega (|\nabla u_q|^2 + q(x)u_q^2)\,dx. \tag{5.2.6}$$

There is an exact analogue of Theorem 3.1.5 that holds for this new functional, with essentially identical proofs. Thus H is a convex functional on $\mathcal{D}$ with a unique global minimizer Q and $\mathcal{L}^2$ gradient $\nabla\ H(q) = u^2 - u_q^2$, and $\nabla\ H(q) = 0$ if and only if $q = Q$.

The theory indicates that the computation of this global minimizer should be a reasonable task. The boundary value problems of the type (5.2.4)–(5.2.5) that arise could, for example, be solved by an appropriate finite element procedure. In situations where it is inadvisable to allow the presence of squares of the stepsizes, or where one needs the first order derivatives explicitly, one would be forced to convert the second order equation into a first order system. Invariant embedding techniques of the type employed in the ODE case, which involve [20, p. 124] the factorizing of the second order equation into a product of two first order equations, appear not to be available in the PDE environment.

5.3 The Inverse Problem for Aquifer Transmissivity

Before we look at the aquifer transmissivity problem it is useful to discuss some of the technical details of the mathematical problem that is used to model it.

5.3.1 A Related Inverse Problem

With Ω as above, let u in $C^2(\Omega)$ be given and consider the problem of finding a function $P > 0$ in $C^{1,1}(\bar{\Omega})$ such that,

$$\operatorname{div}(P(x)\nabla u) = 0, \quad x \in \Omega. \tag{5.3.7}$$

With no further information on P, this is clearly a losing battle as all multiples of a given P also satisfy (5.3.7). In fact, equation (5.3.7) may be rewritten in the form

$$\sum_{i=1}^{n} \frac{\partial P}{\partial x_i}\frac{\partial u}{\partial x_i} + P(x)\frac{\partial^2 u}{\partial x_i^2} = 0, \quad x \in \Omega,$$

and it is clear from this equation that one must specify P at least on an initial hypersurface cut by all the flow lines defined in Ω by the vector field ∇u. With this addition, the inverse problem "given u, find P" has a unique solution, albeit an ill-posed one.

Corresponding to the function u given above, an appropriate functional for this case may be constructed as follows. For functions $p > 0$ in $C^{1,1}(\bar{\Omega})$ let u_p denote the solution of the boundary problem

$$\operatorname{div}(p(x)\nabla v) = 0, \quad x \in \Omega \tag{5.3.8}$$

$$v|_{\partial\Omega} = g, \tag{5.3.9}$$

where $g = u|_{\partial\Omega}$. Observe that, by the Friedrichs inequality [21, p. 211], the Dirichlet operator A_p defined by $A_p\, v = \operatorname{div}(p(x)\nabla\, v)$ on functions v in the set $\mathcal{D}(A_p) = \mathcal{H}_0^2(\Omega)$, is positive in $\mathcal{L}^2(\Omega)$. Consequently, for functions p in the (convex) set

$$\mathcal{D} = \{p : p > 0 \text{ and } p \in C^{1,1}(\bar{\Omega})\}$$

we set

$$H(p) = \int_\Omega p(x)(|\nabla u|^2 - |\nabla u_p|^2)\, dx. \tag{5.3.10}$$

One can show that this functional has gradient in $\mathcal{L}^2(\Omega)$ given by $\nabla H(p) = |\nabla u|^2 - |\nabla u_p|^2$. It is clear that P is a critical point for this functional. One can also see from the definition of the solution u_p that the function u_P is "shared" by any function P that satisfies equation (5.3.7) which may be thought of as a first order equation in the function P. It follows that any function P satisfying (5.3.7) is also a critical point for H.

One possible way to circumvent this difficulty is to make use of the following well-known transformation between (5.3.7) and the Schrödinger equation

$$-\triangle w + Qw = 0. \tag{5.3.11}$$

The substitution

$$u = P^{-1/2} w \tag{5.3.12}$$

into (5.3.7) gives a function w that satisfies (5.3.11) with

$$Q = P^{-1/2} \triangle P^{1/2}. \tag{5.3.13}$$

5.3.2 The Aquifer Problem

We now consider how one might apply these ideas to the the aquifer transmissivity problem. Mathematical models of groundwater flow have long been used for studying groundwater as an essential resource, and for determining the effect of human activity on this resource. A confined aquifer is usually assumed to be described by the diffusion equation

$$\nabla \cdot (P(x)\nabla u) = S(x)\frac{\partial u}{\partial t} + q(x,t),$$

for x in some bounded region Ω in $\mathbf{R}^2$ and $t > 0$, where $u(x,t)$ represents the piezometric head (easy to observe – look at the water levels in various wells at various times), $P(x)$ is the "transmissivity" (roughly, the reciprocal of the resistance to movement of the water), $S(x)$ is the storage coefficient (usually assumed known) and $q(x,t)$ is the discharge-recharge term representing gain or loss of water from the aquifer.

The inverse aquifer identification problem consists in estimating the values of the transmissivity $P(x)$ on the basis of noisy data on $u(x,t)$, given that the functions S and q are known (or ignored). While much work has been done on this problem [5,10,17,22,30,34,36, 37], the ill-posedness of the problem combined with the considerable noise present in the data seem to have led to a lack of reliability in field applications with a consequence that the techniques are not much used in practical water management tasks [50]. In the typical texts (see e.g. [19]), P is usually assumed to be constant; in other situations, an expensive trial and error process called 'model calibration' is employed.

Given the encouraging behaviour that the algorithms discussed above exhibited in the ODE case, it is reasonable to hope that similar effects can be observed in the present situation, especially in light of the fact that a similar sensitivity analysis to that represented by Theorem 3.3.3 is probably possible. This would appear mandatory in the light of the observations in [50].

For simplicity assume that there is no loss or gain to the aquifer from the outside (i.e., q=0) and that the aquifer flow has reached a steady state (i.e., $\partial\, u/\partial\, t = 0$). One thus arrives at equation (5.3.7) considered in inverse problem 2 above. An algorithm to compute P could proceed as follows. Let u be given and choose a starting approximation, p_0, for P. From the transformation (5.3.12) the function $w_0 = p_0^{1/2}u$ solves equation (5.2.4); i.e., $-\triangle\, w_0 + qw_0 = 0$. By minimizing the functional H defined in (5.2.6) (with u replaced by w_0) one obtains a function q_0. Notice that this stage of the algorithm is precisely where noise in the data could be a problem, as it is involves the operation of differentiation. In our case, if the noise is not too excessive we suspect from the one variable case that the method should be effective. Next, one uses the equation (5.3.13) to compute a new approximation, p_1, for P. This step requires solving the Schrödinger equation

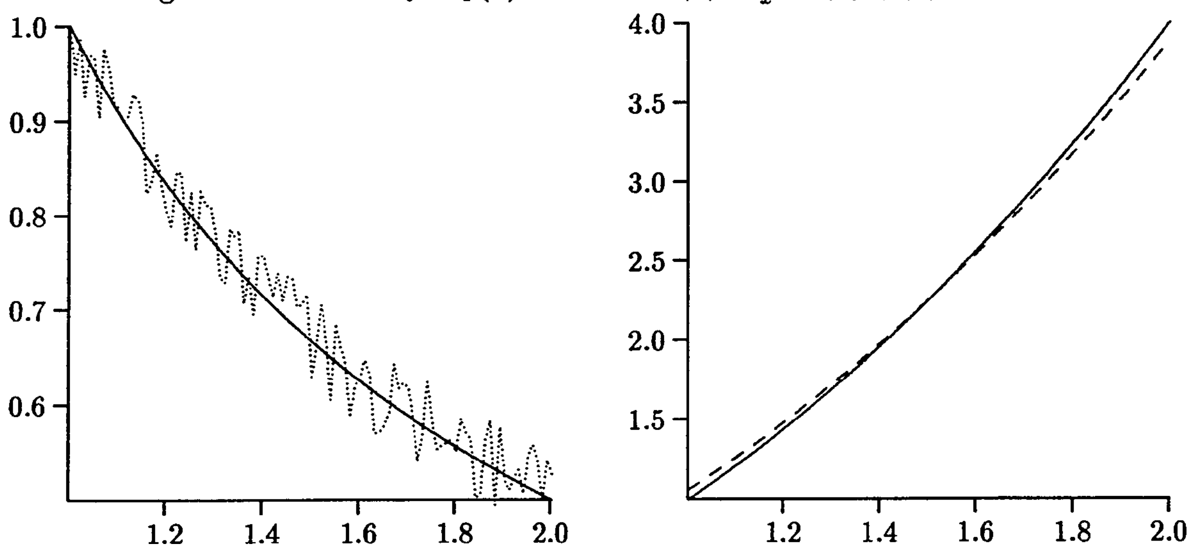
$$-\triangle p^{1/2} + q_0(x)p^{1/2} = 0, \quad x \in \Omega,$$

for $p^{1/2}$. In order to accomplish this, one must know some boundary data for the unknown function P. In theory this can be obtained from well experiments.

In any event, it is clear from the discussion in §5.3.1 that some readings of this nature are mandatory, so we are proposing for the sake of stability that they be done all around the boundary of the region Ω. With this information, a new approximation, p_1, can be obtained from the equation above. The cycle can now be repeated until an iterate p_n, when substituted into the model equation (5.3.7), gives hydraulic head values that are sufficiently close to the measured ones.

To test the numerical viability of this algorithm (in the one dimensional case), we attempted to recover the function $p(x) = x^2$ defined for x in $[0, 1]$, given $\tilde{u}(x) = u(x) + \varepsilon(x)$, where $u(x) = 1/x$ and the values of $\varepsilon(x)$ were chosen randomly from the interval $[-.05, .05]$. Here we used 101 grid points evenly spaced, hence there were 100 subintervals of equal length. The results are summarized in Figure 5.3.1. The solid curves on the left and right

Figure 5.3.1: Recovery of $p(x) = x^2$ from $\tilde{u}(x) = \frac{1}{x} + \epsilon(x)$, $|\epsilon(x)| \leq 0.05$



represent the correct values of u and p, respectively. The dotted curve on the left represents $u + \varepsilon(x)$, and the dashed curve on the right represents the variational approximation to p. As in the variational computation of the derivative (see Figure 4.0.1), the magnitude of the error in the computed function p, is on the same order of magnitude as the error in the given function $u + \varepsilon(x)$.

It is worth mentioning that this proposed method, if it proves practicable in general, will have one significant advantage over most, if not all, previous methods; namely, that the method avoids explicit computation of the derivatives of u. This is a significant source of instability in essentially all of these other approaches.

Bibliography

[1] R. S. Anderssen and P. Bloomfield. On the numerical differentiation of data. Technical Report No. 13, Series 2, Department of Statistics, Fine Hall, Princeton University, Princeton, New Jersey 08540, U.S.A., April 1972.

[2] R. S. Anderssen and P. Bloomfield. Numerical differentiation procedures for non-exact data. *Numer. Math.*, 22:157–182, 1973/74.

[3] I. Babuška, M. Práger, and E. Vitásek. *Numerical Processes in Differential Equations.* Interscience Publishers, New York, 1966.

[4] N. I. Belaya and V. V. Ivanov. Algorithms for reconstruction of derivatives of functions of certain classes, which are optimal with respect to accuracy. *Zh. vȳchisl. Mat. i mat. Fiz.*, 25(3):456–461, 1985.

[5] W. H. Chen and J. H. Seinfeld. Estimation of spacially varying parameters in partial differential equations. *Int. J. Control*, 15:487–495, 1972.

[6] E. A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations.* McGraw-Hill, New York, 1955.

[7] W.A. Coppel. *Disconjugacy.* Springer-Verlag, Berlin, 1971.

[8] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience, 1962.

[9] J. Cullum. Numerical differentiation and regularization. *SIAM J. Numer. Analysis*, 8(2):254–265, 1971.

[10] R. G. Cummings and J. W. McFarland. Groundwater management and salinity control. *Water Resources Research*, 10(5):909–916, 1974.

[11] H.B. Curry. The method of steepest descent for non-linear minimization problems. *Quarterly App. Math*, 2:258–261, 1944.

[12] V. B. Demidovich. The restoration of a function and its derivative on the basis of experimental data. *Vȳchisl. metody i programmirovanie*, 8:96–102, 1967.

[13] T.F. Dolgopolova. Finite dimensional regularization in the case of numerical differentiation of periodic functions. *Matem. zap. Ural'skii un-t*, 7(4):27–33, 1970.

[14] T.F. Dolgopolova and V.K. Ivanov. Numerical differentiation. *Comp. Math and Math. Physics*, 6(3):570–576, 1966.

[15] I. F. Dorofeev. *On stable algorithms concerning numerical differentiation and the numerical solution of some inverse problems for equations with deviating argument*, volume 15 of *Colloq. Math. Soc. János Bolyai*, pages 143–154. North-Holland, Amsterdam, 1977.

[16] N. Dunford and J. T. Schwartz. *Linear Operators*, volume II of *Pure and Applied Mathematics.* Interscience/Wiley, New York, 1958.

[17] Y. Emsellem and G. de Marsily. An automatic solution for the inverse problem. *Water Resources Research*, 7(5):1264–1283, 1971.

[18] H. Engels. Über ein numerisches Differentiationsverfahren mit ableitungsfreien Fehlerschranken. *Z. Angew. Math. Mech.*, 53:T184–T185, 1973.

[19] C. W. Fetter. *Applied Hydrology.* Macmillan, New York, 1988.

[20] L. Fox and D. F. Mayers. *Numerical Solution of Ordinary Differential Equations.* Chapman & Hall, London, 1987.

[21] K. O. Friedrichs. Die randwert- und eigenwertprobleme aus der theorie der elastischen platten. *Math. Ann.*, 98:205–247, 1928.

[22] E. O. Frind and G. F. Pinder. Galerkin solution of the inverse problem for aquifer transmissivity. *Water Resources Research*, 9(5):1397–1410, 1973.

[23] E. Frătilă. A statistical method for numerical differentiation. *Stud. Cerc. Mat.*, 22:443–446, 1970.

[24] W. Gerlach. Nachkorrektur bei numerischer Differentiation. *Z. Angew. Math. Mech.*, 54:193, 1974.

[25] A. A. Goldstein. On steepest descent. *SIAM J. Control*, 3:147–151, 1965.

[26] L. P. Grabar. Numerical differentiation by means of Chebyshev polynomials orthonormalized on a system of equidistant points. *Zh. vy̆chisl. Mat. i mat. Fiz.*, 7(6):1375–1379, 1967.

[27] J. Hadamard. Sur les problèmes dérivées partielles et leur signification physique. *Bull. Univ. Princeton*, pages 49–52, 1902.

[28] P. Hartman. *Ordinary Differential Equations.* John Wiley & Sons, New York, 1964.

[29] A. F. Kalaĭda and V. A Chirikalov. Approximation of the zero rank of a superposition of functions, and matrix algorithms of numerical differentiation. *Vychisl. Prikl. Mat. (Kiev)*, (52):10–17, 1984.

[30] D. Kleinecke. Use of linear programming for estimating geohydrologic parameters of groundwater basins. *Water Resources Research*, 7(2):367–375, 1971.

[31] È. V. Kolpakova. Numerical solution of the problem of reconstructing the derivative. *Differnecial'nye Uravnenija i Vyčisl. Mat. Vyp.*, 6:137–143, 1976.

[32] S. Markov. Computation of the derivatives of a function on the basis of its experimentally measured values. *Automat. i Izčisl. Tehn.*, 9(4):5–11, 1975.

[33] G. Miel and R. Mooney. On the condition number of Lagrangian numerical differentiation. *Appl. Math. Comput.*, 16(3):241–252, 1985.

[34] W. R. Nelson. In-place determination of permeability distribution for heterogeneous porous media through analysis of energy dissipation. *Soc. Pet. Eng. J.*, 8(1):32–42, 1968.

[35] J. Neuberger. Steepest descent for general systems of linear differential equations in Hilbert spaces. *J. Math. Soc. Japan*, 37, 1985.

[36] S. P. Neuman and S. Yakowitz. A statistical approach to the inverse problem of aquifer hydrology. *Water Resources Research*, 15(4):845–860, 1979.

[37] G. Ponzini. An approximate solution to the inverse problem in hydrology. *Energ. Electrica*, 49(8):3–16, 1972.

[38] È. I. Sakalauskas. The Galerkin-Tikhonov method of regularization in the problem of numerical differentiation. *Zh. vȳchisl. Mat. i mat. Fiz*, (11):1742–1747, 1984.

[39] B. Simon. Schrödinger semigroups. *Bull. Amer. Math. Soc.*, 7(3):447–526, 1982.

[40] N. S. Surova. An investigation of the problem of reconstructing a derivative by using an optimal regularizing integral operator. *Numerical methods and programming*, (1):30–34, 1977. Saratov. Gos. Univ., Saratov.

[41] A. N. Tikhonov. On the regularization of ill-posed problems. *Doklady Akad. Nauk SSSR*, 153(1):49–52, 1963.

[42] A. N. Tikhonov. The solution of ill-posed problems. *Doklady Akad. Nauk SSSR*, 151(3):501–504, 1963.

[43] A.E. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems.* John Wiley & Sons, New York, 1977.

[44] J. Vandergraft. *Introduction to Numerical Computation.* Academic Press, New York, 1983.

[45] V.V. Vasin. Regularization of the problem of numerical differentiation. *Matem. zap. Ural'skii un-t*, 7(2):29–33, 1969.

[46] V.V. Vasin. The stable evaluation of a derivative in $C(-\infty,\infty)$. *Zh. vȳchisl. Mat. mat. Fiz.*, 13(6):1383–1389, 1973.

[47] V. V. Vershinin and N. N. Pavlov. Approximation of derivatives by smoothing splines. *Vychisl. Sistemy*, (98):83–91, 1983.

[48] E. Ja. Voronenok. A numerical method of inverse Laplace transformation. *Zh. vȳchisl. Mat. i mat. Fiz.*, 12:1308–1312, 1972.

[49] H. F. Weinberger. *A First Course in Partial Differential Equations.* John Wiley & Sons, Inc., New York, 1965.

[50] S. Yakowitz and L. Duckstein. Instability in aquifer identification: theory and case studies. *Water Resources Research*, 16(6):1045–1064, 1980.

[51] E. Zeidler. *Fixed Point Theorems*, volume I of *Nonlinear Functional Analysis and its Applications.* Springer-Verlag, New York, 1986.

[52] E. Zeidler. *Variational Methods and Optimization*, volume III of *Nonlinear Functional Analysis and its Applications.* Springer-Verlag, New York, 1986.

Appendix

FORTRAN Code for the Implementation of the Variational Algorithm

```
c       This program computes the derivative of the function u(x)
c       by locating the function Q that minimizes H(q) and then
c       computing u and u' as the solution of the system
c       u'=v
c       v'=Qv.

        real q,q1,q2,qmin,gL2,gH1,h,maxgL2,maxgH1
        integer skip,ws,it
        dimension q(1001),q1(1001),q2(1001),u(1001),du(1001),it(3),
     8  uq1(1001),duq1(1001),uq2(1001),duq2(1001),f(1001),e1(1001),
     8  gL2(1001),gH1(1001),dgH1(1001),x(1001),fdu(1001),temp(1001),
     8  e2(1001)
        ws=2
        print*,'Enter xa,xb,n,eps'
        read*,xa,xb,n,eps
        h=(xb-xa)/(n-1)

        print*
        qmin=-(2/(xb-xa))**2
        d2ua=d2uu(xa)
        d2ub=d2uu(xb)
        s=(d2ub-d2ua)/(xb-xa)
        do 10 i=1,n
        x1=xa+(i-1)*h
        q1(i)=ua+s*(x1-xa)
        q1(i)=0.0
10      continue
        do 15 i=1,n
        x1=xa+(i-1)*h
```

```
      x(i)=x1
      u(i)=uu(x1)+eps*(2*ran3(int(10000*ran3(int(10000*
    8 ran3(i+1000)))))-1)
      du(i)=duu(x1)
15    continue
      fdu(1)=(u(2)-u(1))/h
      fdu(n)=(u(n)-u(n-1))/h
      do 17 i=2,n-1
      fdu(i)=(u(i+1)-u(i-1))/(2*h)
17    continue
      do 20 i=1,n
      f(i)=0
20    continue
      do 30 i=1,n
      q(i)=1
30    continue
1492  call dsolve(xa,xb,n,q1,f,0,u(1),u(n),uq1,duq1,ws)
      do 40 i=1,n
      gL2(i)=u(i)*u(i)-uq1(i)*uq1(i)+eps2*q1(i)
40    continue
      print*
      call nsolve(xa,xb,n,q,gL2,0,0,0,gH1,dgH1)
      print*
      print*,'Choices:'
      print*,'Compute alpha --------------- (1)'
      print*,'Enter new alpha ------------- (2)'
      print*,'View data ------------------- (3)'
      print*,'Send data to a file --------- (4)'
      print*,'New q ----------------------- (5)'
      print*,'Stop ------------------------ (6)'
      print*
      read*,l
      if (l.eq.1) then
      alph=alpha(xa,xb,n,q1,uq1,gL2,gH1)
      endif
      if (l.eq.2) then
      print*,'Enter new alpha'
      print*
      read*, alph
      print*
      endif
      do 60 i=1,n
      q2(i)=q1(i)-alph*gH1(i)
60    continue
      if ((l.eq.1).or.(l.eq.2)) then
      do 61 i=1,n
```

```
        temp(i)=abs(gL2(i))
61      continue
        maxgL2=vmax(temp,n)
        do 62 i=1,n
        temp(i)=abs(gH1(i))
62      continue
        maxgH1=vmax(temp,n)
        print*
        print*,'H(old)-H(new) = ',diffh(xa,xb,n,q1,q2,u,uq1,uq2)
        print*,'infnorm(gradL2) = ',maxgL2
        print*,'infnorm(gradH1) = ',maxgH1
        print*,'Need ',qmin,' <= q'
        print*,'q2min = ',vmin(q2,n)
        print*,'h = ',h
        print*
        endif
        call dsolve(xa,xb,n,q2,f,0,u(1),u(n),uq2,duq2,ws)
        if (l.eq.3) then
        skip=int((n+1)/10)
        print*,' x du duq fdu'
        do 70 i=1,n,skip
        x1=xa+(i-1)*h
        write(*,71) x(i),du(i),duq2(i),fdu(i)
70      continue
71      format(f10.7,2x,f10.7,2x,f10.7,2x,f10.7,2x,f10.7)
        endif
        if (l.eq.4) then
        do 1963 i=1,n
        e1(i)=abs(du(i)-duq2(i))
        e2(i)=abs(du(i)-fdu(i))
1963    continue
        if (abs(eps).gt.0.000000001) then
        open(unit=7,file='funct')
        do 954 i=1,n
        uuu=uu(x(i))+.01*(2*rand(i)-1)
        write(7,955) x(i),uu(x(i)),uq2(i),u(i)
954     continue
955     format(f10.7,f10.7,f10.7,f10.7)
        open(unit=3,file='rerror')
        do 9641 i=1,n
        write(3,9651) x(i),e1(i),e2(i)
9641    continue
9651    format(f10.7,f10.7,f10.7)
        open(unit=4,file='rderiv')
        do 751 i=1,n
        write(4,721) x(i),du(i),duq2(i),fdu(i)
```

```
751     continue
721     format(f10.7,f10.7,f10.7,f10.7)
        endif
        if (abs(eps).lt.0.000000001) then
        open(unit=2,file='serror')
        do 1964 i=1,n
        write(2,1965) x(i),e1(i)/abs(du(i)),e2(i)/abs(du(i))
1964    continue
1965    format(f10.7,f10.7,f10.7)
        open(unit=1,file='sderiv')
        do 75 i=1,n
        write(1,72) x(i),du(i),duq2(i),fdu(i)
75      continue
72      format(f10.7,f10.7,f10.7,f10.7)
        endif
        endif
        if (l.eq.5) then
        do 80 i=1,n
        q1(i)=q2(i)
80      continue
        endif
        if (l.eq.6) then
        stop
        endif
        goto 1492
        end
```

```
c       This subroutine solves the following Dirichlet BVP:
c       -y''+qy=f
c       y(a)+alpha*y'(a)=beta
c       y(b)=gamma.
c       This is accomplished by converting the BVP to the following
c       pair of initial value problems:
c       t'=qtt-1
c       w'=qtw-ft
c       t(a)=alpha
c       w(a)=beta
c       and
c       r'=q-r*r-f/y
c       y'=r*y
c       r(b)=(w(b)-gamma)/(t(b)*gamma)
c       y(b)=gamma
c       using the transformation y+ty'=w.
c       The IVP's are solved using 4th order Runge'-Kutta.
c       y and y' are returned as n-dimensional vectors.

        subroutine dsolve(xa,xb,n,q,f,alpha,beta,gamma,uq,uq1,j)
        real h,q,f,uq,uq1,y1,y2,k1,k2,k3,k4,k5,k6,h1,h2,h3,h4,h5,h6,r
        dimension q(1001),f(1001),y1(1001),y2(1001),uq(1001),
     8  uq1(1001),r(1001)
        h=(xb-xa)/(n-1)

c       Note:  Here y1=t and y2=w.
c       This the forward sweep which solves the first IVP.

        y1(1)=alpha
        y2(1)=beta
        do 110 i=1,n-1
        x=xa+(i-1)*h
        k1=h*f1(x,y1(i),y2(i),xa,h,n,q)
        h1=h*f2(x,y1(i),y2(i),xa,h,n,q,f)
        z1=x+h/4.
        z2=y1(i)+k1/4.
        z3=y2(i)+h1/4.
        k2=h*f1(z1,z2,z3,xa,h,n,q)
        h2=h*f2(z1,z2,z3,xa,h,n,q,f)
        z1=x+3*h/8.
        z2=y1(i)+3*k1/32+9*k2/32
        z3=y2(i)+3*h1/32+9*h2/32
        k3=h*f1(z1,z2,z3,xa,h,n,q)
        h3=h*f2(z1,z2,z3,xa,h,n,q,f)
        z1=x+12*h/13
        z2=y1(i)+1932*k1/2197-7200*k2/2197+7296*k3/2197
```

```
      z3=y2(i)+1932*h1/2197-7200*h2/2197+7296*h3/2197
      k4=h*f1(z1,z2,z3,xa,h,n,q)
      h4=h*f2(z1,z2,z3,xa,h,n,q,f)
      z1=x+h
      z2=y1(i)+439*k1/216-8*k2+3680*k3/513-845*k4/4104
      z3=y2(i)+439*h1/216-8*h2+3680*h3/513-845*h4/4104
      k5=h*f1(z1,z2,z3,xa,h,n,q)
      h5=h*f2(z1,z2,z3,xa,h,n,q,f)
      z1=x+h/2.
      z2=y1(i)-8*k1/27+2*k2-3544*k3/2565+1859*k4/4104-11*k5/40
      z3=y2(i)-8*h1/27+2*h2-3544*h3/2565+1859*h4/4104-11*h5/40
      k6=h*f1(z1,z2,z3,xa,h,n,q)
      h6=h*f2(z1,z2,z3,xa,h,n,q,f)
      y1(i+1)=y1(i)+16*k1/135+6656*k3/12825
     8 +28561*k4/56430-9*k5/50+2*k6/55
      y2(i+1)=y2(i)+16*h1/135+6656*h3/12825
     8 +28561*h4/56430-9*h5/50+2*h6/55
110   continue

c     Here we solve the second IVP.
c     This section does Riccotti
      if (j.eq.1) then
      uq(n)=gamma
      r(n)=(y2(n)-gamma)/(y1(n)*gamma)
      do 210 i=n,2,-1
      x=xa+(i-1)*h
      k1=-h*f3(x,uq(i),r(i))
      h1=-h*f4(x,uq(i),r(i),xa,h,n,q,f,y2,y1)
      z1=x-h/4.
      z2=uq(i)+k1/4
      z3=r(i)+h1/4
      k2=-h*f3(z1,z2,z3)
      h2=-h*f4(z1,z2,z3,xa,h,n,q,f,y2,y1)
      z1=x-3*h/8
      z2=uq(i)+3*k1/32+9*k2/32
      z3=r(i)+3*h1/32+9*h2/32
      k3=-h*f3(z1,z2,z3)
      h3=-h*f4(z1,z2,z3,xa,h,n,q,f,y2,y1)
      z1=x-12*h/13
      z2=uq(i)+1932*k1/2197-7200*k2/2197+7296*k3/2197
      z3=r(i)+1932*h1/2197-7200*h2/2197+7296*h3/2197
      k4=-h*f3(z1,z2,z3)
      h4=-h*f4(z1,z2,z3,xa,h,n,q,f,y2,y1)
      z1=x-h
      z2=uq(i)+439*k1/216-8*k2+3680*k3/513-845*k4/4104
      z3=r(i)+439*h1/216-8*h2+3680*h3/513-845*h4/4104
```

```
      k5=-h*f3(z1,z2,z3)
      h5=-h*f4(z1,z2,z3,xa,h,n,q,f,y2,y1)
      z1=x-h/2
      z2=uq(i)-8*k1/27+2*k2-3544*k3/2565+1859*k4/4104-11*k5/40
      z3=r(i)-8*h1/27+2*h2-3544*h3/2565+1859*h4/4104-11*h5/40
      k6=-h*f3(z1,z2,z3)
      h6=-h*f4(z1,z2,z3,xa,h,n,q,f,y2,y1)
      uq(i-1)=uq(i)+16*k1/135+6656*k3/12825
     8 +28561*k4/56430-9*k5/50+2*k6/55
      r(i-1)=r(i)+16*h1/135+6656*h3/12825
     8 +28561*h4/56430-9*h5/50+2*h6/55
210   continue
      do 30 i=1,n
      uq1(i)=r(i)*uq(i)
30    continue
      endif
c     This section does not do Riccotti
      if (j.eq.2) then
      if (abs(y1(n)).lt.1.0E-28) then
      print*,'t(b) too small in dsolve'
      print*,'t(b) = ',y1(n)
      print*,'t(b) set to .000000001'
      y1(n)=.000000001
      endif
      r(n)=(y2(n)-gamma)/y1(n)
      do 2102 i=n,2,-1
      x=xa+(i-1)*h
      h1=-h*f5(x,r(i),xa,h,n,q,f,y2,y1)
      z1=x-h/4.
      z3=r(i)+h1/4
      h2=-h*f5(z1,z3,xa,h,n,q,f,y2,y1)
      z1=x-3*h/8
      z3=r(i)+3*h1/32+9*h2/32
      h3=-h*f5(z1,z3,xa,h,n,q,f,y2,y1)
      z1=x-12*h/13
      z3=r(i)+1932*h1/2197-7200*h2/2197+7296*h3/2197
      h4=-h*f5(z1,z3,xa,h,n,q,f,y2,y1)
      z1=x-h
      z3=r(i)+439*h1/216-8*h2+3680*h3/513-845*h4/4104
      h5=-h*f5(z1,z3,xa,h,n,q,f,y2,y1)
      z1=x-h/2
      z3=r(i)-8*h1/27+2*h2-3544*h3/2565+1859*h4/4104-11*h5/40
      h6=-h*f5(z1,z3,xa,h,n,q,f,y2,y1)
      r(i-1)=r(i)+16*h1/135+6656*h3/12825
     8 +28561*h4/56430-9*h5/50+2*h6/55
2102  continue
```

```
      do 47 i=1,n
      uq1(i)=r(i)
      uq(i)=y2(i)-y1(i)*r(i)
47    continue
      endif
      return
      end
```

```
c       This subroutine solves the following Neumann BVP:
c       -y''+qy=f
c       y'(a)+alpha*y(a)=beta
c       y'(b)=gamma.
c       This is accomplished by converting the BVP to the following
c       pair of initial value problems:
c       t'=tt-q
c       w'=tw-f
c       t(a)=alpha
c       w(a)=beta
c       and
c       y'=w-ty
c       y(b)=(w(b)-gamma)/t(b)
c       using the transformation y'+ty=w.
c       The IVP's are solved using 4th order Runge'-Kutta.
c       y and y' are returned as n dimensional vectors.

        subroutine nsolve(xa,xb,n,q,f,alpha,beta,gamma,uq,uq1)
        real h,q,f,uq,uq1,y1,y2,k1,k2,k3,k4,k5,k6,h1,h2,h3,h4,h5,h6
        dimension q(1001),f(1001),y1(1001),y2(1001),uq(1001),
     8  uq1(1001)
        h=(xb-xa)/(n-1)

c       Here we solve the first IVP.
c       Note:  y1=t and y2=w

        y1(1)=alpha
        y2(1)=beta
        do 110 i=1,n-1
        x=xa+(i-1)*h
        k1=h*g1(x,y1(i),y2(i),xa,h,n,q)
        h1=h*g2(x,y1(i),y2(i),xa,h,n,f)
        z1=x+h/4.
        z2=y1(i)+k1/4.
        z3=y2(i)+h1/4.
        k2=h*g1(z1,z2,z3,xa,h,n,q)
        h2=h*g2(z1,z2,z3,xa,h,n,f)
        z1=x+3*h/8.
        z2=y1(i)+3*k1/32+9*k2/32
        z3=y2(i)+3*h1/32+9*h2/32
        k3=h*g1(z1,z2,z3,xa,h,n,q)
        h3=h*g2(z1,z2,z3,xa,h,n,f)
        z1=x+12*h/13
        z2=y1(i)+1932*k1/2197-7200*k2/2197+7296*k3/2197
        z3=y2(i)+1932*h1/2197-7200*h2/2197+7296*h3/2197
        k4=h*g1(z1,z2,z3,xa,h,n,q)
```

```
      h4=h*g2(z1,z2,z3,xa,h,n,f)
      z1=x+h
      z2=y1(i)+439*k1/216-8*k2+3680*k3/513-845*k4/4104
      z3=y2(i)+439*h1/216-8*h2+3680*h3/513-845*h4/4104
      k5=h*g1(z1,z2,z3,xa,h,n,q)
      h5=h*g2(z1,z2,z3,xa,h,n,f)
      z1=x+h/2.
      z2=y1(i)-8*k1/27+2*k2-3544*k3/2565 +1859*k4/4104-11*k5/40
      z3=y2(i)-8*h1/27+2*h2-3544*h3/2565 +1859*h4/4104-11*h5/40
      k6=h*g1(z1,z2,z3,xa,h,n,q)
      h6=h*g2(z1,z2,z3,xa,h,n,f)
      y1(i+1)=y1(i)+16*k1/135+6656*k3/12825
     8 +28561*k4/56430-9*k5/50+2*k6/55
      y2(i+1)=y2(i)+16*h1/135+6656*h3/12825
     8 +28561*h4/56430-9*h5/50+2*h6/55
110   continue

c     Here we solve the second IVP.

      if (abs(y1(n)).lt.1.0E-28) then
      print*,'y1(b) too small in nsolve'
      print*,'y1(b) = ',y1(n)
      print*,'y1(b) set to .00001'
      y1(n)=.00001
      endif
      uq(n)=(y2(n)-gamma)/y1(n)
      do 210 i=n,2,-1
      x=xa+(i-1)*h
      h1=-h*g4(x,uq(i),xa,h,n,y2,y1)
      z1=x-h/4.
      z3=uq(i)+h1/4
      h2=-h*g4(z1,z3,xa,h,n,y2,y1)
      z1=x-3*h/8
      z3=uq(i)+3*h1/32+9*h2/32
      h3=-h*g4(z1,z3,xa,h,n,y2,y1)
      z1=x-12*h/13
      z3=uq(i)+1932*h1/2197-7200*h2/2197+7296*h3/2197
      h4=-h*g4(z1,z3,xa,h,n,y2,y1)
      z1=x-h
      z3=uq(i)+439*h1/216-8*h2+3680*h3/513-845*h4/4104
      h5=-h*g4(z1,z3,xa,h,n,y2,y1)
      z1=x-h/2
      z3=uq(i)-8*h1/27+2*h2-3544*h3/2565 +1859*h4/4104-11*h5/40
      h6=-h*g4(z1,z3,xa,h,n,y2,y1)
      uq(i-1)=uq(i)+16*h1/135+6656*h3/12825
     8 +28561*h4/56430-9*h5/50+2*h6/55
```

```
210     continue
        do 215 i=1,n
        uq1(i)=y2(i)-y1(i)*uq(i)
215     continue
        return
        end
```

```
      function alpha(xa,xb,n,q1,uq1,gL2,gH1)
      real h,q1
      dimension q1(1001),gL2(1001),gH1(1001),temp1(1001),
     8 w(1001),dw(1001),uq1(1001)
      h=(xb-xa)/(n-1)

c     Compute the denominator of alpha.
      do 40 i=1,n
      temp1(i)=uq1(i)*gH1(i)
40    continue
      call dsolve(xa,xb,n,q1,temp1,0,0,0,w,dw,2)
      do 50 i=1,n
      temp1(i)=uq1(i)*gH1(i)*w(i)
50    continue
      s2=2*ss(temp1,xa,h,n)
      print*,'denominator = ',s2
      if (abs(s2).lt.1.0E-50) then
      print*
      print*,'Zero or negative denominator'
      print*,'alpha set to zero'
      alpha=0
      print*
      goto 70
      endif

c     Compute the numerator of alpha.
      do 60 i=1,n
      temp1(i)=gL2(i)*gH1(i)
60    continue
      s1=ss(temp1,xa,h,n)
      print*,'numerator = ',s1
      alpha=s1/s2
      print*,'alpha = ',alpha
70    print*
      return
      end
```

```
      function diffh(xa,xb,n,q1,q2,u,uq1,uq2)
      real h
      dimension q1(1001),q2(1001),u(1001),uq1(1001),uq2(1001),v(1001)
      h=(xb-xa)/(n-1)
      do 20 i=1,n
      v(i)=(q1(i)-q2(i))*(u(i)*u(i)-uq1(i)*uq2(i))
20    continue
      diffh=ss(v,xa,h,n)
      return
      end

      function ran3(idum)
      implicit real*4(m)
      parameter (mbig=4000000.,mseed=1618033.,mz=0.,fac=2.5E-7)
c     parameter (mbig=1000000000,mseed=161803398,mz=0,fac=1.E-9)
      dimension ma(55)
      data iff /0/
      if(idum.lt.0.or.iff.eq.0) then
      iff=1
      mj=mseed-iabs(idum)
      mj=mod(mj,mbig)
      ma(55)=mj
      mk=1
      do 11 i=1,54
      ii=mod(21*i,55)
      ma(ii)=mk
      mk=mj-mk
      if(mk.lt.mz)mk=mk+mbig
      mj=ma(ii)
11    continue
      do 13 k=1,4
      do 12 i=1,55
      ma(i)=ma(i)-ma(1+mod(i+30,55))
      if(ma(i).lt.mz)ma(i)=ma(i)+mbig
12    continue
13    continue
      inext=0
      inextp=31
      idum=1
      endif
      inext=inext+1
      if(inext.eq.56)inext=1
      inextp=inextp+1
      if(inextp.eq.56)inextp=1
      mj=ma(inext)-ma(inextp)
      if(mj.lt.mz)mj=mj+mbig
```

```
      ma(inext)=mj
      ran3=mj*fac
      return
      end

      function c(r,x,xa,h,n)
      real h,r,m
      dimension r(1001)
      do 310 i=2,n-2
      x1=xa+(i-1)*h
      x2=xa+i*h
      if ((x1.le.x).and.(x.le.x2)) then
      m=(r(i+1)-r(i))/h
      c=m*(x-x1)+r(i)
      endif
310   continue
      if (x.le.xa+h) then
      m=(r(2)-r(1))/h
      c=m*(x-xa)+r(1)
      endif
      if (xa+(n-2)*h.le.x) then
      m=(r(n)-r(n-1))/h
      c=m*(x-xa-(n-2)*h)+r(n-1)
      endif
      return
      end

      function f1(x,y1,y2,xa,h,n,q)
      real q,h
      dimension q(1001)
      f1=c(q,x,xa,h,n)*y1**2-1
      return
      end

      function f2(x,y1,y2,xa,h,n,q,f)
      real q,h,f
      dimension q(1001),f(1001)
      f2=c(q,x,xa,h,n)*y1*y2-c(f,x,xa,h,n)*y1
      return
      end

      function f3(x,y1,y2)
      f3=y2*y1
      return
      end
```

```
      function f4(x,y1,y2,xa,h,n,q,f,w,t)
      real q,h
      dimension q(1001),w(1001),t(1001)
      f4=c(q,x,xa,h,n)-y2*y2-c(f,x,xa,h,n)/y1
      return
      end

      function f5(x,y2,xa,h,n,q,f,w,t)
      real q,h
      dimension q(1001),w(1001),t(1001),f(1001)
      f5=-c(q,x,xa,h,n)*c(t,x,xa,h,n)*y2
     8 +c(q,x,xa,h,n)*c(w,x,xa,h,n)-c(f,x,xa,h,n)
      return
      end

      function g1(x,y1,y2,xa,h,n,q)
      real q,h
      dimension q(1001)
      g1=y1**2-c(q,x,xa,h,n)
      return
      end

      function g2(x,y1,y2,xa,h,n,f)
      real h,f
      dimension f(1001)
      g2=y1*y2-c(f,x,xa,h,n)
      return
      end

      function g4(x,r,xa,h,n,w,t)
      real h,r,t,w
      dimension w(1001),t(1001)
      g4=c(w,x,xa,h,n)-c(t,x,xa,h,n)*r
      return
      end

      function ss(t,xa,h,n)
      real t,h
      dimension t(1001)
      sum1=0
      sum2=0
      do 111 i=2,n-1,2
      sum1=sum1+t(i)
111   continue
      do 222 i=3,n-2,2
      sum2=sum2+t(i)
```

```
222     continue
        ss=(n-1)*h*(4*sum1+2*sum2+t(1)+t(n))/(3.0*(n-1))
        return
        end

        function vmax(v,n)
        real v
        dimension v(1001)
        vmax=v(1)
        do 10 i=2,n
        if (v(i).gt.vmax) then
        vmax=v(i)
        endif
10      continue
        return
        end

        function vmin(v,n)
        real v
        dimension v(1001)
        vmin=v(1)
        do 10 i=2,n
        if (v(i).lt.vmin) then
        vmin=v(i)
        endif
10      continue
        return
        end

        function uu(x)
        uu=cos(x)
        return
        end
```

GRADUATE SCHOOL
UNIVERSITY OF ALABAMA AT BIRMINGHAM
DISSERTATION APPROVAL FORM

Name of Candidate Robert Wallace

Major Subject Applied Mathematics

Title of Dissertation A Variational Method for Numerical Differentiation

Dissertation Committee:

Ian Knowles, **Chairman**

Christer Bennewitz

Richard C. Brown

John Neuberger

Richard McNider

John Young

Director of Graduate Program John Mayer

Dean, UAB Graduate School W. A. Sibley

Date 9/27/93

PS-1428