# A NUMERICAL METHOD FOR SOLVING SINGULAR DIFFERENTIAL EQUATIONS UTILIZING STEEPEST DESCENT IN WEIGHTED SOBOLEV SPACES 

## DISSERTATION

Presented to the Graduate Council of the University of North Texas in Partial

Fulfillment of the Requirements

For the Degree of DOCTOR OF PHILOSOPHY

## By

William Ted Mahavier, B.S., M.S.
Denton, Texas
August, 1995

# A NUMERICAL METHOD FOR SOLVING SINGULAR DIFFERENTIAL EQUATIONS UTILIZING STEEPEST DESCENT IN WEIGHTED SOBOLEV SPACES 

## DISSERTATION

Presented to the Graduate Council of the University of North Texas in Partial

Fulfillment of the Requirements

For the Degree of DOCTOR OF PHILOSOPHY

## By

William Ted Mahavier, B.S., M.S.
Denton, Texas
August, 1995

Mahavier, William Ted, A Numerical Method for Solving Singular Differential Equations Utilizing Steepest Descent in Weighted Sobolev Spaces. Doctor of Philosophy (Mathematics), August 1995, 67 pp., 18 tables, 3 figures, bibliography, 18 titles.

We develop a numerical method for solving singular differential equations and demonstrate the method on a variety of singular problems including first order ordinary differential equations, second order ordinary differential equations which have variational principles, and one partial differential equation.

The method is a variation of steepest descent in Sobolev spaces which is a variation of descent based on the Euclidean gradient. We cast the differential equation as a least squares problem yielding a functional representing the equation. A weighted Sobolev space for the problem is chosen where the weights are based on the functional. This produces gradients which take into account both the weights and the boundary conditions for the given equation.

Results are presented which demonstrate the improvements obtained by computing based on weighted Sobolev gradients over computing based on either unweighted Sobolev gradients or on the Euclidean gradient.

## ACKNOWLEDGMENTS

The constant support of my wife, my family, and my advisor have made this paper possible. This research was partially funded by a grant from the Texas Advanced Research Project.

## TABLE OF CONTENTS

Page
LIST OF FIGURES ..... vi
LIST OF TABLES ..... vii
Chapter
1 INTRODUCTION ..... 1
1.1 Overview ..... 1
1.2 History ..... 3
2 A MOTIVATING EXAMPLE ..... 5
2.1 Introduction to Spaces ..... 5
2.2 Results ..... 12
2.3 Convergence ..... 13
3 FIRST ORDER PROBLEMS ..... 17
3.1 The Problem ..... 17
3.2 Numerical Method in Detail ..... 17
3.3 Examples ..... 25
4 SECOND ORDER PROBLEMS ..... 33
4.1 A Variational Problem ..... 34
4.2 Legendre's Equation of order $\alpha=1$ ..... 42
5 A PARTIAL DIFFERENTIAL EQUATION ..... 46
5.1 The problem ..... 46
5.2 Notation ..... 46
5.3 Numerics ..... 49
5.4 Results ..... 53
6 MATRICES OF STEEPEST DESCENT ..... 56
6.1 Matrix for First Order Equations ..... 56
6.2 Matrix for Second Order Equations ..... 60
6.3 Matrix for Partial Differential Equations ..... 60
7 CONCLUSIONS ..... 66
BIBLIOGRAPHY ..... 67

## LIST OF FIGURES

2.1 Motivational Problem ..... 15
3.1 Constrained Singular Problem ..... 29
4.1 Variational Problem ..... 40

## LIST OF TABLES

2.1 Motivational Problem ..... 14
2.2 Motivational Problem ..... 14
2.3 Motivational Problem ..... 14
3.1 Unconstrained Singular Problem ..... 27
3.2 Unconstrained Singular Problem ..... 27
3.3 Constrained Singular Problem ..... 30
3.4 Constrained Singular Problem ..... 30
3.5 Partially Constrained Singular Problem ..... 32
3.6 Partially Constrained Singular Problem ..... 32
4.1 Variational Problem ..... 41
4.2 Variational Problem ..... 41
4.3 Variational Problem ..... 41
4.4 Legendre's Equation ..... 45
4.5 Legendre's Equation ..... 45
4.6 Legendre's Equation ..... 45
5.1 Non-Singular Partial Differential Equation ..... 55
5.2 Singular Partial Differential Equation ..... 55
5.3 Singular Partial Differential Equation ..... 55

## CHAPTER 1

## INTRODUCTION

### 1.1 Overview

This paper exhibits a numerical method for the study of differential equations which have linear singularities. The method extends the work pioneered by Neuberger [N5], where he introduces steepest descent based on Sobolev gradients. Our gradients arise from weighted Sobolev spaces such as those introduced by Kufner [KA] and Elschner $[\mathrm{E}]$. The weights (and spaces) depend on the singularity of the differential equation we wish to consider and on the functional we use to represent the equation. We refer throughout the paper to three types of steepest descent: $L^{2}$ steepest descent, Sobolev steepest descent, and weighted Sobolev steepest descent. We show that for the problems studied, weighted Sobolev descent outperforms Sobolev steepest descent which in turn outperforms $L^{2}$ descent. The value of Sobolev steepest descent has already been established for difficult problems, a few of which will be referenced in the next section. This paper demonstrates not a fast solution to one applied problem, but rather a systematic approach for using weights to improve an already established method. The numerical method is applied throughout the paper on first order problems, variational problems, and one partial differential equation in order to demonstrate the adaptability of the method. Methods traditionally used to improve
$L^{2}$ descent such as conjugate gradient or multi-step methods are also applicable to descent based on Sobolev gradients and weighted Sobolev gradients.

We first develop the theory for the following general class of singular first order ordinary differential equations.

$$
\begin{align*}
& q(t) y^{\prime}(t)=f(t, y(t))  \tag{1.1}\\
& k_{1} y(a)+k_{2} y(b)=k_{3}
\end{align*}
$$

where $a, b \in I, f \in C_{I}^{2}, q \in C_{I}^{1}$, and $q(t)=0$ for some $t \in I=[0,1]$. Here we walk the reader through the numerical method in great detail, outlining first the descent methods without constraints and then adding the constraints. The method is then extended to certain second order differential equations which have variational properties such as Legendre's equation. Finally, we apply our method to the first order partial differential equation on $\Omega=I \times I$,

$$
\begin{align*}
& x u_{1}(x, y)+y u_{2}(x, y)=0  \tag{1.2}\\
& u(x, 0)=1=u(0, y) \forall(x, y) \in \Omega
\end{align*}
$$

which has linear singularities in each variable.
Throughout the paper we exhibit the continuous ideas which motivated the numerical algorithms, posing questions which we wish to answer in the continuous case, then consider the discretized numerical algorithms, followed by comparisons of the three methods. Chapter 3 is a prerequisite for Chapters 4 and 5 because we outline certain information on projections and linear systems solvers in Chapter 3 which are assumed in the later chapters.

The advantages of the algorithm will be expanded upon in the conclusions, but briefly they are as follows: The algorithm we put forth outperforms $L^{2}$ and Sobolev steepest descent in every case considered. The algorithm is flexible and can be easily adapted to consider a wide variety of different types of differential equations with varying boundary conditions. The success of the numerical algorithm points the way for further study of the continuous theory which motivates the numerics.

### 1.2 History

Steepest descent goes back as far as Cauchy and Sobolev steepest descent was first introduced by Neuberger [N6]. Neuberger and his students have demonstrated the power of Sobolev steepest descent for many specific problems. See [K], [G], $\{\mathrm{DM}]$ for examples of Sobolev steepest descent on non-singular problems and see [N2] for an example of a nonlinear second order differential equation with nonlinear singularity arising from a problem concerning transonic flow. Existence and uniqueness arguments for singular problems in Sobolev spaces have been given by Schuchman in $[\mathrm{S}]$ and by Canic and Keyfitz in [CK]. For a specific paper concerning Sobolev gradients which are constructed specifically based on the problem at hand (as ours are), consider the paper [RN].

While mathematicians and scientists have sought solutions to differential equations using steepest descent based on the discretized $L^{2}$ gradient (or Euclidean gradient), we demonstrate that the choice of the underlying space (and thus the underlying
gradient) is all important to developing efficient numerical methods. Support for this belief has been offered before for non-singular problems by Neuberger. In [N6], Neuberger addresses this point, comparing $L_{I}^{2}$ steepest descent to Sobolev steepest descent for the differential equation $y^{\prime}=y, y(0)=1$. His result indicates that using the standard gradient one may expect a larger number of divisions of the interval to result in smaller descent step sizes and thus slower convergence rates. On the other hand, for the Sobolev gradient the step size remains large regardless of the number of divisions. Indeed, Neuberger points out that solving the problem with machine accuracy and 100 divisions on the interval $I$ requires 500,000 iterations using the standard gradient and only 7 using the Sobolev gradient.

We show similar results for singular problems where weighted Sobolev steepest descent outperforms Sobolev steepest descent. We use throughout this paper the phrases $L^{2}$ steepest descent, Sobolev steepest descent, and weighted Sobolev steepest descent to indicate whether the gradient flow arises from an $L^{2}$ space, a Sobolev space without weight, or a weighted Sobolev space.

## CHAPTER 2

## A MOTIVATING EXAMPLE

This chapter introduces the spaces and techniques used throughout the paper as well as illustrating our first example. This chapter also puts forth here the necessary continuous theory for the remainder of the paper. The example which follows marks the departure from Neuberger's work by adding the weighted spaces to the descent process while at the same time supporting his belief that one observes vastly improved numerical results by choosing a space which is well suited to the problem.

### 2.1 Introduction to Spaces

We introduce the spaces here in the continuous case although for the numerical work we will be using discretized versions of these spaces. The discrete spaces are defined in Chapter 3. Consider the simple problem, $2 t y^{\prime}=y$ on $I=[0,1]$ with final condition $y(1)=1$ and cast the equation as a least squares minimization problem, setting

$$
J(u)=\int_{I}\left(2 j u^{\prime}-u\right)^{2}
$$

for every $u \in C_{I}^{1}$ where $j$ denotes the identity on $I$. If there existed a $C^{1}$ solution, then a zero of $J$ would indicate this solution. The fact that there is no $C^{1}$ solution motivated the development of the spaces which follow. We note here that the results
we obtain for this problem are not as precise as results we obtain in later examples. This is due to the fact that our motivating example is singular in two senses. First, the problem is singular in the traditional sense that we cannot solve for $y^{\prime}$ on the interval and, second, the solution has infinite derivative. Later problems, such as $t y^{\prime}=y$, which have solutions with finite derivatives yield machine precision results.

We proceed to consider three Hilbert spaces and seek solutions via steepest descent based on gradients arising from each of the spaces. These spaces and notations remain constant throughout the paper. Let $L=L_{I}^{2}$ and $\langle\cdot, \cdot\rangle_{L}$ denote the $L$ inner product. Define

$$
\begin{aligned}
& D_{1}={\left.\overline{\left\{\binom{u}{u^{\prime}}: u \in C_{I}^{I}\right.}\right\}^{L \times L}}^{L} \\
& H=\pi_{1} D_{1} \\
& \langle u, v\rangle_{H}=\langle u, v\rangle_{L}+\left\langle D_{1}(u), D_{1}(v)\right\rangle_{L} .
\end{aligned}
$$

Of course, if $u \in C_{I}^{1}$ then $D_{1}(u)=u^{\prime}$ and for any $u \in H, D_{1}(u)=u^{\prime}$ where ' denotes the generalized derivative. Using the same technique to develop the weighted Sobolev space, we define for any $w \in C_{I}^{1}$ which is positive almost everywhere,

$$
\begin{aligned}
D_{1}^{w} & ={\overline{\left\{\binom{u}{w u^{\prime}}: u \in C_{I}^{1}\right.}}^{-L \times L} \\
H_{w} & =\pi_{1} D_{1}^{w}
\end{aligned}
$$

$$
\langle u, v\rangle_{H_{w}}=\langle u, v\rangle_{L}+\left\langle D_{1}^{w}(u), D_{1}^{w}(v)\right\rangle_{L} .
$$

If $w=1$ then $H_{w}=H_{1}=H$ and if $w=j$ and $u \in C_{I}^{1}$ then $D_{1}^{w}(u)=D_{1}^{j}(u)=j u^{\prime}$.
Neuberger has shown [ N 4$]$ that $D_{1}$ is a function in the sense that no two elements of $D_{1}$ have the same first coordinates and distinct second coordinates. We now show that $D_{1}^{j}$ is a function. Thanks to John M. Neuberger for his ideas on this proof.

Theorem $1 I f j(t)=t$ on I then $D_{1}^{j}$ is a function in the sense that no two elements of $D_{1}^{j}$ have the same first coordinates and distinct second coordinates.

Proof. Let

$$
A=\left\{\binom{u}{j u^{\prime}}: u \in C_{I}^{1}\right\}
$$

so that $\bar{A}=D_{1}^{j}$. Let $\binom{f}{g},\binom{f}{h}$ be elements of $\bar{A}$. Let $\binom{a_{n}}{j a_{n}^{\prime}}$ be a sequence in $A$ converging to $\binom{f}{g}$ in $L^{2} \times L^{2}$ and let $\binom{b_{n}}{j b_{n}}$ be a sequence in $A$ converging to $\binom{f}{h}$ in $L^{2} \times L^{2}$. We prove four lemmas, the last resulting in the proof of the theorem.

Lemma $1\left(j a_{n}\right)_{\mathrm{N}}$ is uniformly Cauchy on $I$.

Proof. $a_{n} \rightarrow f$ in $L^{2}$ implies $a_{n} \rightarrow f$ in $L^{1}$, thus $\left(a_{n}\right)_{\mathrm{N}}$ is $L^{1}$ Cauchy. Similarly, $\left(j a_{n}^{\prime}\right)_{\mathrm{N}}$ is $L^{1}$ Cauchy. Let $\epsilon>0$. Let $N_{1}$ in $\mathbf{N}$ such that for all $n, m \geq N_{1}, \int_{I}\left|a_{n}-a_{m}\right|<$ $\frac{\epsilon}{2}$. Let $N_{2}$ in N such that for all $n, m \geq N_{2}, \int_{I}\left|j a_{n}^{\prime}-j a_{m}^{\prime}\right|<\frac{\epsilon}{2}$. Let $N=\max \left\{N_{1}, N_{2}\right\}$, $t \in I$, and $n, m \geq N .\left|t a_{n}(t)-t a_{m}(t)\right|=\left|\int_{0}^{t}\left(j a_{n}\right)^{\prime}-\left(j a_{m}\right)^{\prime}\right|=\left|\int_{0}^{t} j a_{n}^{\prime}-j a_{m}^{\prime}+a_{n}-a_{m}\right| \leq$ $\int_{0}^{t}\left|j a_{n}^{\prime}-j a_{m}^{\prime}\right|+\int_{0}^{t}\left|a_{n}-a_{m}\right|<\frac{\epsilon}{2}+\frac{\epsilon}{2}=\epsilon$.

Lemma $2\left(j a_{n}-j b_{n}\right)_{\mathrm{N}}$ converges to zero uniformly on $I$.

Proof. Since $\left(j a_{n}\right)_{\mathrm{N}}$ is uniformly Cauchy, $j a_{n} \rightarrow j f$ uniformly and since $\left(j b_{n}\right)_{\mathrm{N}}$ is uniformly Cauchy, $j b_{n} \rightarrow j f$ uniformly. Hence, $j a_{n}-j b_{n} \rightarrow j f-j f=0$ uniformly.

Lemma $3\left|\int_{s}^{t}\left(j a_{n}^{\prime}-j b_{n}^{\prime}\right)\right|$ converges to zero uniformly on $I$.

Proof. Let $\epsilon>0$. From Lemma 2 let $N_{1} \in \mathrm{~N}$ such that for all $n \geq N_{1}, \mid x a_{n}(x)-$ $x b_{n}(x) \left\lvert\,<\frac{\epsilon}{3}\right.$ for every $x \in I$. Let $N_{2} \in \mathbf{N}$ such that $n \geq N_{2},\left\|a_{n}-b_{n}\right\|<\frac{\epsilon}{3}$. Let $n \geq \max \left\{N_{1}, N_{2}\right\}$ and $s, t \in I .\left|\int_{s}^{t} j a_{n}^{\prime}-j b_{n}^{\prime}\right|=\left|\int_{s}^{t}\left(j b_{n}^{\prime}+b_{n}-b_{n}+a_{n}-a_{n}-j a_{n}^{\prime}\right)\right|=$ $\left|\int_{s}^{t}\left(\left(j b_{n}\right)^{\prime}-\left(j a_{n}\right)^{\prime}\right)+\int_{s}^{t}\left(a_{n}-b_{n}\right)\right| \leq\left|\int_{s}^{t}\left(\left(j b_{n}\right)^{\prime}-\left(j a_{n}\right)^{\prime}\right)\right|+\int_{s}^{t}\left|a_{n}-b_{n}\right| \leq \mid t b_{n}(t)-t a_{n}(t)-$ $s b_{n}(s)+s a_{n}(s)\left|+\left\|a_{n}-b_{n}\right\| \leq\left|t b_{n}(t)-t a_{n}(t)\right|+\left|s b_{n}(s)-s a_{n}(s)\right|+\left\|a_{n}-b_{n}\right\|<\frac{\epsilon}{3}+\frac{\epsilon}{3}+\frac{\epsilon}{3}=\right.$ $\epsilon$.

Lemma $4\left|\int_{s}^{t}(g-h)\right|=0$ for every $s, t \in I$.

Proof. Recall, $j a_{n}^{\prime} \rightarrow g$ in $L^{2}$ implies $j a_{n}^{\prime} \rightarrow g$ in $L^{1}$ and $j b_{n}^{\prime} \rightarrow h$ in $L^{2}$ implies $j b_{n}^{\prime} \rightarrow h$ in $L^{1}$. Pick $s, t \in I$ and we have $\left|\int_{s}^{t} g-\int_{s}^{t} j a_{n}^{\prime}\right| \leq \int_{s}^{t}\left|g-j a_{n}^{\prime}\right| \rightarrow$ 0 and $\left|\int_{s}^{t} h-\int_{s}^{t} j b_{n}^{\prime}\right| \leq \int_{s}^{t}\left|h-j b_{n}^{\prime}\right| \rightarrow 0$. Therefore, $\lim _{n \rightarrow \infty} \int_{s}^{t} j a_{n}^{\prime}=\int_{s}^{t} g$ and $\lim _{n \rightarrow \infty} \int_{s}^{t} j b_{n}^{\prime}=\int_{s}^{t} h$. We conclude $\left|\int_{s}^{t}(g-h)\right|=\left|\int_{s}^{t} g-\int_{s}^{t} h\right|=\mid \lim _{n \rightarrow \infty} \int_{s}^{t} j a_{n}^{\prime}-$ $\lim _{n \rightarrow \infty} \int_{s}^{t} j b_{n}^{\prime}\left|=\left|\lim _{n \rightarrow \infty}\left(\int_{s}^{t}\left(j a_{n}^{\prime}-j b_{n}^{\prime}\right)\right)\right|=\lim _{n \rightarrow \infty}\right| \int_{s}^{t}\left(j a_{n}^{\prime}-j b_{n}^{\prime}\right) \mid$. Thus, $\mid \int_{s}^{t}(g-$ $h)\left|=\lim _{n \rightarrow \infty}\right| \int_{s}^{t}\left(j a_{n}^{\prime}-j b_{n}^{\prime}\right) \mid$. Let $\epsilon>0$ and from Lemma 3 choose $N \in \mathbf{N}$ such that for all $n \geq N,\left|\int_{s}^{t}\left(j a_{n}^{\prime}-j b_{n}^{\prime}\right)\right|<\epsilon$ for every $s, t \in I$. Then, $\lim _{n \rightarrow \infty}\left|\int_{s}^{t}\left(j a_{n}^{\prime}-j b_{n}^{\prime}\right)\right| \leq \epsilon$ for every $s, t \in I$. Hence for every $\epsilon>0$ and for every $s, t \in I$ we have $\left|\int_{s}^{t}(g-h)\right| \leq \epsilon$ thus $\int_{s}^{t}(g-h)=0$. This implies that $g-h=0$ almost everywhere. For if $g-h \neq 0$
almost everywhere then there exists $x$ and $\epsilon>0$ such that, without loss of generality, $g-f>0$ on $(x-\epsilon, x+\epsilon)$. Therefore, $\int_{x-\epsilon}^{x+\epsilon} f-g>0$, a contradiction. Conclude, $\left|\int_{s}^{t}(g-h)\right|=0$ in $L^{2}$ and $g=h$ almost everywhere. q.e.d.

Theorem $2 D_{1}^{j}$ is a non-expansive closed bounded densely defined linear operator.

Proof. The fact that $D_{1}^{j}$ is closed follows from its definition. Recall that $D_{1}^{j}$ is densely defined iff the domain of $D_{1}^{j}$ is dense in $L$. But the polynomials on I are dense in $L$ and $H_{j}=\operatorname{dom}\left(D_{1}^{j}\right)$ is a superset of the polynomials on $I$ and a subset of $L$, thus, $D_{1}^{j}$ is densely defined. Since we have for any $u \in H_{j}$,

$$
\frac{\left\|D_{1}^{j} u\right\|_{L}}{\|u\|_{H_{j}}}=\frac{\left\|D_{1}^{j} u\right\|_{L}}{\|u\|_{L}+\left\|D_{1}^{j} u\right\|_{L}} \leq 1
$$

we see that $D_{1}^{j}$ is bounded and non-expansive as an operator from $H_{j}$ to $L$. q.e.d.
We demonstrate that $H_{w}$ is a larger space by showing that the solution to our motivating example is in $H_{w}$ but not in $H$. We follow this with a proof that $H_{w}$ is a Hilbert space.

Theorem $3 H$ is a proper subset of $H_{j}$.
Proof. Suppose $u \in H$. There exists, $\left(\begin{array}{c}u_{n} \\ \\ u_{n}^{\prime}\end{array}\right)$ such that $\binom{u_{n}}{u_{n}^{\prime}} \rightarrow\binom{u}{u^{\prime}}$ in $L \times L$, hence, $\binom{u_{n}}{w u_{n}^{\prime}} \rightarrow\binom{u}{w u^{\prime}}$ in $L \times L$. We conclude $H \subset H_{w}$.

We claim that $z=\sqrt{j} \in H_{j} \backslash H$. We first show that $z \in H_{j}$. We must show that $z=\pi_{1}\binom{\alpha}{\beta}$ where $\binom{\alpha}{\beta}=\lim _{n \rightarrow \infty}\binom{u_{n}}{j u_{n}^{\prime}}$ for some sequence $\left(u_{n}\right) \in C_{I}^{1}$.

Put

$$
u_{n}(x)= \begin{cases}-\frac{1}{2} n^{3 / 2} x^{2}+\frac{3}{2} n^{1 / 2} x & \text { if } x \in\left[0, \frac{1}{n}\right]  \tag{2.1}\\ \sqrt{x} & \text { if } x \in\left[\frac{1}{n}, 1\right]\end{cases}
$$

$u_{n} \in C_{I}^{0}$ since $u_{n}\left(\frac{1}{n}-\right)=\frac{1}{\sqrt{n}}=u_{n}\left(\frac{1}{n}+\right) . u_{n} \in C_{I}^{1}$ since $u_{n}^{\prime}\left(\frac{1}{n}-\right)=\frac{\sqrt{n}}{2}=u_{n}^{\prime}\left(\frac{1}{n}+\right)$.
Since $u_{n} \rightarrow \sqrt{j}=u$ pointwise almost everywhere, $u_{n} \xrightarrow{L} \sqrt{j}=u$. Since $j u_{n}^{\prime} \rightarrow$ $\frac{\sqrt{3}}{2}=j u^{\prime}$ pointwise almost everywhere $j u_{n}^{\prime} \xrightarrow{L} \frac{\sqrt{3}}{2}=j u^{\prime}$. We conclude $u \in H_{j}$.

To demonstrate that $z \notin H$ we show $z \neq \pi_{1}\binom{\alpha}{\beta}$ for any pair $\binom{\alpha}{\beta}$ satisfying $\binom{\alpha}{\beta}=\lim _{n \rightarrow \infty}\binom{v_{n}}{v_{n}^{\prime}}$ where $v_{n} \in C_{I}^{1}$. If there exists such a sequence converging in $L \times L$ then $\left(v_{n}^{\prime}\right)$ is a Cauchy sequence in the Hilbert space $L$ converging to $z^{\prime} \notin L$, a contradiction. q.e.d.

Neuberger has shown [ N 4$]$ that the space $H$ is the often studied space $H_{I}^{1,2}$ and we will show that $H_{w}$ is an alternate definition of a Hilbert space defined in Kufner.

Theorem $4 H_{w}$ is a Hilbert space.

Proof. Certainly, $\langle u, v\rangle_{H_{w}}=\langle u, v\rangle_{L}+\left\langle D_{1}^{w} u, D_{1}^{w} v\right\rangle_{L}$ is an inner product, so we show only that $H_{w}$ is complete. If $\left(u_{n}\right)$ is a Cauchy sequence in $H_{w}$ then there exists $\binom{u_{n}}{v_{n}} \in D_{w}$ such that $u_{n}=\pi_{1}\binom{u_{n}}{v_{n}}$. Since $\left(u_{n}\right)$ is Cauchy in $H_{w}$ we have $\left\|u_{n}-u\right\|_{L} \rightarrow 0$ and $\left\|v_{n}-v\right\|_{L} \rightarrow 0$ for some $\binom{u}{v} \in L \times L$. We conclude that $\left\|u_{n}-u\right\|_{H_{w}}=\left\|u_{n}-u\right\|_{L}+\left\|v_{n}-v\right\|_{L} \rightarrow 0$ and thus $\left\|u_{n}-u\right\|_{H_{w}} \rightarrow 0$ in $H_{w}$. q.e.d.

A discussion of the history of generalized derivatives would be too lengthy to include here and we refer the reader to [A]. We have already stated that $H$ is the
often studied Hilbert space, $H_{I}^{1,2}$. We now observe that $H_{w}$ is a special case of the spaces defined by Kufner [KA]. Let $D^{\alpha}$ be the $\alpha^{\text {th }}$ generalized derivative, where $\alpha$ is a multi-index, $\alpha=\left(\alpha_{1}, \ldots, \alpha_{k}\right)$ and $\sum_{i}\left|\alpha_{i}\right|=k$, where $k$ denotes the number of derivatives, $p$ indicates the overlying $L^{p}$ space, $\Omega \subset \Re^{n}$, and $\sigma=\left\{\sigma_{\alpha}:|\alpha| \leq k\right\}$. We now have the space

$$
W=\left(W^{k, p}(\Omega, \sigma),\|\cdot\|_{W}\right)
$$

where the norm is given by

$$
\|u\|=\left\{\sum_{|\alpha| \leq k} \int_{\Omega}\left|D^{\alpha} u(x)\right|^{p} \sigma_{\alpha}(x) d x\right\}^{1 / p}
$$

Setting $\sigma=\left(1, t^{2}\right)$ we have the space,

$$
W=\left(W^{k, p}(\Omega, \sigma),\|\cdot\|\right)
$$

where if $u \in C_{I}^{1}$ then

$$
\|u\|_{W}=\int u^{2}+w^{2} u^{\prime 2}=\|u\|_{H_{w}} .
$$

Theorem $5 H_{w}=W$

Proof. Let $P$ denote the polynomials with rational coefficients. Each of $H$ and $W$ are subsets of $L$ containing $P$ and $P$ is dense in $L$, thus each of $H$ and $W$ are dense in $L$. Furthermore, $\|p\|_{H_{w}}=\|p\|_{W}$ for every $p \in P$. Let $x \in W$. There exists a sequence $\left(p_{n}\right) \in P$ such that $p_{n} \xrightarrow{W} x$. Also, $\left\|p_{n}-x\right\|_{L} \leq\left\|p_{n}-x\right\|_{W}$ implies $p_{n} \xrightarrow{L} x$. $p_{n} \xrightarrow{W} x$ implies $\left(p_{n}\right)$ is a $W$-Cauchy sequence. Since $\|p\|_{H_{w}}=\|p\|_{W}$ for every $p \in P$, $\left(p_{n}\right)$ is a $H_{w}$-Cauchy sequence also. Thus, there exists $\hat{x}$ such that $p_{n} \xrightarrow{H_{\psi}} \hat{x}$. Finally,
$\left\|p_{n}-\hat{x}\right\|_{L} \leq\left\|p_{n}-\hat{x}\right\|_{W}$ implies $p_{n} \xrightarrow{L} \hat{x}$. But limits are unique in $L$ thus $x=\hat{x}$. We conclude that $W \subseteq H_{w}$ and a symmetric argument shows that $H_{w} \subseteq W$. q.e.d.

With the notation just defined, we redefine $J$. As defined above, the domain of $J$ was all $C_{I}^{1}$ functions but this is not acceptable, as the solution to our problem is $y(t)=\sqrt{t}$ is not $C^{1}$. Thus we redefine

$$
J(u)=\int_{I}\left(2 D_{1}^{w}(u)-u\right)^{2}
$$

for every $u \in H_{w}$ and seek a solution in $H_{w}$.

### 2.2 Results

Having demonstrated the continuous properties of the spaces, we list some results for this problem. Table 2.1 illustrates our claim that the number of iterations and the time required to solve the problem both decrease while the accuracy increases as we consider $L, H$, and $H_{w}$ descent respectively. Table 2.2 demonstrates how drastically results change as we increase the number of divisions. While the $H_{w}$ descent still performs well, $L$ descent was unable to obtain the desired accuracy. When in future examples we use a small number of divisions or a large stopping criteria it is for precisely the reasons demonstrated here. In many cases $L$ and $H$ will simply not converge for too tight a stopping criteria or for too large a number of divisions. We have observed that the order of magnitude of the gradient for $H_{w}$ descent is usually on the order of the stopping criteria. This is certainly a desirable quality for steepest descent which is not shared by $L$ or $H$ descent for singular problems.

Figure 2.1 shows the difference after three iterations between the two descent processes. The graph shows four lines. They are shaded from light to dark and vary from thick to thin. Respectively they represent the initial estimate, the Sobolev approximation to the solution after 100 iterations, the weighted Sobolev approximation to the solution after three iterations, and the solution itself. The advantage of the weight near the origin (where the singularity occurs) is clear from the graph. The results in Figure 2.1 are supported by the following reasoning. $L$ descent does not take the derivative of the function into consideration, hence $H$ descent outperforms $L$ descent. While $H$ descent considers the derivative, the solution does not belong to $H$. Since $\sqrt{t} \in H_{w} \backslash H, H_{w}$ descent outperforms $H$ descent. These difficulties in the continuous setting carry over to the numerical settings. We will see examples in later chapters where we are unable to get acceptable accuracies using $L$ or $H$ descent methods in any amount of time, while we obtain machine accuracy results using $H_{w}$ descent.

### 2.3 Convergence

The next theorem, due to Neuberger [ N 4 ], guarantees convergence in the continuous case for the problem stated and for the problems follow for each of the Sobolev spaces considered.

Theorem 6 Suppose $\mathcal{H}$ and $\mathcal{K}$ are Hilbert spaces and $G \in L(\mathcal{H}, \mathcal{K})$. Suppose $g \in \mathcal{K}$, $v \in \mathcal{H}, G v=g$, and $\phi(u)=\frac{1}{2}\|G u-g\|^{2}$ for every $u \in \mathcal{H}$. If $x \in \mathcal{H}$ and $z$ is the

Table 2.1: Motivational Problem

| 2 ty $^{\prime}-\mathbf{y}=\mathbf{0}$ |  |  | $\mathbf{y}(\mathbf{1})=\mathbf{2}$ |  | $\mathbf{N}=100$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $L$ | 21518 | 208 | $10^{-5}$ | $10^{-2}$ | $1.3 \times 10^{-1}$ |
| $H$ | 830 | 10 | $10^{-5}$ | $10^{-3}$ | $7.1 \times 10^{-2}$ |
| $H_{w}$ | 11 | 1 | $10^{-5}$ | $10^{-5}$ | $2.7 \times 10^{-3}$ |

Table 2.2: Motivational Problem

| 2 ty $^{\prime}-\mathbf{y}=\mathbf{0}$ |  |  | $\mathbf{y}(\mathbf{1})=\mathbf{2}$ | $\mathbf{N}=\mathbf{1 , 0 0 0}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H$ | 10,000 | 1290 | $10^{-10}$ | $10^{-4}$ | $4.1 \times 10^{-3}$ |
| $H_{w}$ | 26 | 3 | $10^{-10}$ | $10^{-6}$ | $8.7 \times 10^{-4}$ |

Table 2.3: Motivational Problem

| $2 \mathbf{t y}^{\prime}-\mathbf{y}=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{1})=\mathbf{2}$ |  | $\mathbf{N}=\mathbf{1 0}, \mathbf{0 0 0}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H_{w}$ | 41 | 53 | $10^{-10}$ | $10^{-8}$ | $2.8 \times 10^{-4}$ |



Figure 2.1: Motivational Problem
function on $[0, \infty)$ so that

$$
z(0)=x, z^{\prime}(t)=-(\nabla \phi)(z(t)), t \geq 0
$$

then $u=\lim _{n \rightarrow \infty} z(t)$ exists and $G u=g$.

If we consider $D_{1}$ as an operator on $H$, then $D_{1}$ is bounded, [N4], and we have shown in Section 2 that $D_{1}^{j}$ is bounded as an operator on $H_{j}$. Thus we may apply the theorem in either space putting $v(t)=\sqrt{t}$ and $g=0$. In $H$, we have $\mathcal{H}=H$, $\mathcal{K}=\Re, G u=2 j D_{1} u-u$, and $\nabla=\nabla_{H}$. In $H_{w}$, we have $\mathcal{H}=H_{w}, \mathcal{K}=\Re$, and $G u=2 D_{\mathbf{1}}^{w}(u)-u$ and $\nabla=\nabla_{H_{w}}$.

## CHAPTER 3

## FIRST ORDER PROBLEMS

This chapter presents a numerical method for solving a general class of singular first order differential equations with various constraints. After presenting the discrete weighted steepest descent, we demonstrate the distinction between descent in the three Hilbert spaces. Finally, we offer examples of the algorithm by considering unconstrained, constrained, and partially constrained problems.

### 3.1 The Problem

Let $k_{1}, k_{2}, k_{3}, a, b \in \Re$ with $a<b$ and $q \in C_{I}^{1}$. Let $f:[a, b] \times \Re \rightarrow \Re$ be differentiable with respect to the second variable. The problem is

$$
\begin{align*}
& q(t) y^{\prime}(t)=f(t, y(t))  \tag{3.1}\\
& k_{1} y(a)+k_{2} y(b)=k_{3}
\end{align*}
$$

### 3.2 Numerical Method in Detail

We break the problem into two cases, one describing the solution to the problem without boundary conditions and the other incorporating boundary conditions. The former of these two cases may be considered entirely a subset of the latter for both coding and theoretical considerations and is presented here to introduce the method in its simplest form.

### 3.2.1 $q(t) y^{\prime}(t)=f(t, y(t))$ without Boundary Conditions

We first introduce the necessary notation, much of which was motivated by [N1] and [N6].

Throughout the paper $\|\cdot\|_{L}$ denotes the Euclidean norm and we denote $x \in \Re^{m}$ by $x=\left(x_{1}, \ldots, x_{m}\right)$. Let $k_{1}, k_{2}, k_{3}, a, b \in \Re$ with $a<b$ and $q \in C_{I}^{1}$. Let $I=[a, b]$ and $f: I \times \Re \rightarrow \Re$ be differentiable with respect to the second variable. Suppose $n$ is the number of divisions into which we partition the interval $[a, b]$, and $\delta=(b-a) / n$. Let $\epsilon$ be the stopping criteria for our algorithm; we stop when $\left\|y^{\text {new }}-y\right\|_{L}<\epsilon$ where $y$ and $y^{\text {new }}$ denote successive approximations to the solution.

To simplify our notation, we define discretized versions of the identity and derivative operators. Let $D_{0}: \Re^{n+1} \rightarrow \Re^{n}, D_{1}^{w}: \Re^{n+1} \rightarrow \Re^{n}$, and $D_{w}: \Re^{n+1} \rightarrow \Re^{2 n}$ be defined by

$$
D_{0}(x)=\left(\begin{array}{c}
\frac{x_{1}+x_{2}}{2} \\
\vdots \\
\frac{x_{n}+x_{n+1}}{2}
\end{array}\right), D_{1}^{w}(x)=\left(\begin{array}{c}
\left(\frac{w_{2}+w_{2}}{2}\right)\left(\frac{x_{2}-x_{1}}{\delta}\right) \\
\vdots \\
\left(\frac{w_{n}+w_{n+1}}{2}\right)\left(\frac{x_{n+1}-x_{n}}{\delta}\right)
\end{array}\right), D_{w}(x)=\binom{D_{0}(x)}{D_{1}^{w}(x)} .
$$

The three discretized versions of the spaces $L, H$, and $H_{w}$ from the introduction are now $\left(\Re^{n+1},\langle\cdot, \cdot\rangle_{L}\right),\left(\Re^{n+1},\langle\cdot, \cdot\rangle_{H}\right)$, and $\left(\Re^{n+1},\langle\cdot, \cdot\rangle_{H_{w}}\right)$ where

$$
\begin{gathered}
\langle u, v\rangle_{H_{\omega}}=\left\langle D_{0}(u), D_{0}(v)\right\rangle+\left\langle D_{1}^{w}(u), D_{1}^{w}(v)\right\rangle= \\
\sum_{k=1}^{n}\left(\frac{u_{k+1}+u_{k}}{2}\right)\left(\frac{v_{k+1}+v_{k}}{2}\right)+\left(\frac{w_{k+1}+w_{k}}{2}\right)^{2}\left(\frac{u_{k+1}-u_{k}}{\delta}\right)\left(\frac{v_{k+1}-v_{k}}{\delta}\right)
\end{gathered}
$$

for all $u, v \in \Re^{n+1}$.

Observe that $D_{w}$ relates the Euclidean and Sobolev norms by $\|\cdot\|_{H_{w}}=\left\|D_{w}(\cdot)\right\|_{L}$. Let $y \in \Re^{n+1}$ and for all $k=1,2, \ldots, n+1$, let $t_{k}=a+(k-1) \delta$ and $f_{k}=f\left(t_{k}, y_{k}\right)$. Define $J:\left(\Re^{n+1},\|\cdot\|_{H_{w}}\right) \rightarrow \Re$ by

$$
\begin{aligned}
J(y) & =\frac{1}{2}\left\|D_{1}^{w} y-D_{0} f\right\|_{L}^{2} \\
& =\frac{1}{2} \sum_{k=1}^{n}\left(\left(\frac{q_{k+1}+q_{k}}{2}\right)\left(\frac{y_{k+1}-y_{k}}{\delta}\right)-\frac{f_{k}+f_{k+1}}{2}\right)^{2} .
\end{aligned}
$$

We will minimize $J$ via successive approximations, since a zero of $J$ corresponds to a solution of our differential equation. We specialize one well-known theorem to our case before outlining the method. To verify the non-singular nature of the matrix $A_{w}$ we refer the reader to $[\mathrm{RSN}]$ and for a more complete description of the matrix, we refer the reader to Chapter 6 .

Theorem 7 If $\langle\cdot,\rangle_{H_{w}}$ denotes the discretized Sobolev inner product on $\Re^{n+1}$ and $\langle\cdot, \cdot\rangle_{L}$ represents the standard inner product on $\Re^{n+1}$ then there exists a matrix $A_{w}$ in $L\left(\Re^{n+1}, \Re^{n+1}\right)$ such that $\langle x, y\rangle_{H_{w}}=\left\langle A_{w} x, y\right\rangle=\left\langle x, A_{w} y\right\rangle$ for every $x, y \in \Re^{n+1}$. Moreover, $A_{w}=D_{w}^{t} D_{w}$ and $A_{w}\left(\nabla_{H_{w}} J\right)(x)=\left(\nabla_{L} J\right)(x)$ for every $x \in \Re^{n+1}$.

Proof. Let $\left(e_{i}\right)_{i=1}^{n+1}$ represent the standard basis on $\Re^{n+1}$. If $A_{w}$ exists and $a_{i, j}$ denotes the $i, j$ - th component of $A_{w}$ then

$$
\begin{aligned}
\left\langle e_{j}, e_{i}\right\rangle_{H_{w}} & =\left\langle A_{w} e_{j}, e_{i}\right\rangle \\
& =\sum_{k}\left(A_{w} e_{j}\right)_{k}\left(e_{i}\right)_{k} \\
& =\left(A_{w} e_{j}\right)_{i} \\
& =a_{i, j}
\end{aligned}
$$

Hence, if $A_{w}$ exists then $A_{w}$ is unique and $a_{i, j}=\left\langle e_{j}, e_{i}\right\rangle_{H_{w}}$. Let $a_{i, j}=\left\langle e_{j}, e_{i}\right\rangle_{H_{w}}$. If $x, y \in \Re^{n+1}$, then

$$
\begin{aligned}
\left\langle A_{w} x, y\right\rangle & =\sum_{i}\left(A_{w} x\right)_{i} y_{i} \\
& =\sum_{i} \sum_{j}\left(a_{i, j} x_{j}\right) y_{i} \\
& =\sum_{i} \sum_{j}\left(e_{j}, e_{i}\right\rangle_{H_{w}} x_{j} y_{i} \\
& =\sum_{i}\left\langle x, e_{i}\right\rangle_{H_{w}} y_{i} \\
& =\langle x, y\rangle_{H_{w}}
\end{aligned}
$$

A similar argument shows $\langle x, y\rangle_{H_{w}}=\left\langle x, A_{w} y\right\rangle$.
Since $\langle x, y\rangle_{H_{\omega}}=\left\langle D_{0} x, D_{0} y\right\rangle+\left\langle D_{1}^{\omega} x, D_{1}^{w} y\right\rangle=\left\langle D_{0}^{t} D_{0} x, y\right\rangle+\left\langle\left(D_{1}^{w}\right)^{t} D_{1}^{w} x, y\right\rangle=$ $\left\langle\left(D_{0}^{t} D_{0}+\left(D_{1}^{w}\right)^{t} D_{1}^{w}\right) x, y\right\rangle$ for every $x, y \in \Re^{n+1}$ we have $A_{w}=D_{w}^{t} D_{w}$. Given $u \in \Re^{n+1}$ we have $\left\langle\left(\nabla_{L} J\right)(u), h\right\rangle=J^{\prime}(u)(h)=\left\langle\left(\nabla_{H_{w}} J\right)(u), h\right\rangle_{H_{w}}=\left\langle A_{w}\left(\nabla_{H_{w}} J\right)(u), h\right\rangle$ for every $h \in \Re^{n+1}$. Consequently, $A_{w}\left(\nabla_{H_{w}} J\right)(x)=\left(\nabla_{L} J\right)(x)$ for every $x \in \Re^{n+1}$. q.e.d.

Before outlining the method, we define 'optimal' step size. We say that $h$ is the optimal step size if $h$ minimizes $J\left(y-h\left(\nabla_{H_{w}} J\right)(y)\right)$. If $f$ is linear, then the optimal step size is easily computed as

$$
h=\frac{\left\|\left(\nabla_{H_{w}} J\right)(y)\right\|_{H_{w}}^{2}}{\left\langle\left(\nabla_{H_{w}} J\right)^{2}(y),\left(\nabla_{H_{w}} J\right)(y)\right\rangle_{H_{w}}}
$$

by minimizing $\alpha(h)=J\left(y-h\left(\nabla_{H_{w}} J\right)(y)\right)$. We now outline the method. Compute the matrix, $A_{w}$, and choose an initial guess $y$. Compute the standard gradient, $\left(\nabla_{L} J\right)(y)$, and solve the linear system from Theorem $7, A_{w}\left(\nabla_{H_{w}} J\right)(y)=\left(\nabla_{L} J\right)(y)$,
for the Sobolev gradient, $\left(\nabla_{H_{w}} J\right)(y)$. Follow the negative of this direction an 'optimal' distance, $h$. Consider the distance between the new point, $y^{n e w}$, and $y$. If this distance is less than $\epsilon$, consider $y^{\text {new }}$ a solution, else repeat the process with $y^{n e w}$ as an initial guess.

## Algorithm:

1. Compute the matrix, $A_{w}$.
2. Choose $y \in \Re^{n+1}$.
3. Compute the gradient of $J$ at $y,\left(\nabla_{L} J\right)(y)$.
4. Solve $A_{w}\left(\nabla_{H_{w}} J\right)(y)=\left(\nabla_{L} J\right)(y)$ for $\left(\nabla_{H_{w}} J\right)(y)$.
5. Determine $h$ which minimizes $J\left(y-h\left(\nabla_{H_{w}} J\right)(y)\right)$.
6. Let $y^{\text {new }}=y-h\left(\nabla_{H_{\omega}} J\right)(y)$.
7. If $\left\|y^{\text {new }}-y\right\|_{L}<\epsilon$, we have a solution; else, put $y=y^{\text {new }}$ and repeat steps 3 through 7.
3.2.2 $q(t) y^{\prime}(t)=f(t, y(t))$ with Boundary Conditions.

Recall that the boundary conditions were $k_{1} y(a)+k_{2} y(b)=k_{3}$ and retain the notation from Section 3.2.1. In addition, consider the perturbation space, $\Re_{0}^{n+1}=$ $\left\{x \in \Re^{n+1}: k_{1} x_{1}+k_{2} x_{n+1}=0\right\}$ and let $\pi_{H_{w}}$ denote the orthogonal projection of $\Re^{n+1}$ onto $\Re_{0}^{n+1}$ under the Sobolev inner product.

It is noteworthy that in each of the following algorithms, $\pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$ is in the
linear subspace, $\Re_{0}^{n+1}$. This assures that each successive approximation also lies in $\Re_{0}^{n+1}$, thus the boundary conditions are exactly maintained for each approximation.

A numerically naive generalization of the case without boundary conditions follows. Start with an initial guess, $y$. Compute the Sobolev gradient, $\left(\nabla_{H_{w}} J\right)(y)$, and the projection, $\pi_{H_{w}}$. Project $\left(\nabla_{H_{w}} J\right)(y)$ onto $\Re_{0}^{n+1}$ under $\pi_{H_{w}}$ and follow the negative of this direction an optimal distance, $h$. Consider the distance between the new point, $y^{\text {new }}$, and $y$. If this distance is less than $\epsilon$, consider $y^{\text {new }}$ a solution, else repeat the process with $y^{\text {new }}$ as our initial guess.

## Algorithm:

1. Compute the matrix, $A_{w}$.
2. Choose $y \in \Re_{0}^{n+1}$.
3. Compute the gradient of $J$ at $y,\left(\nabla_{L} J\right)(y)$.
4. Solve $A_{w}\left(\nabla_{H_{w}} J\right)(y)=\left(\nabla_{L} J\right)(y)$ for $\left(\nabla_{H_{w}} J\right)(y)$.
5. Compute the Sobolev projection, $\pi_{H_{w}}$, and project $\left(\nabla_{H_{w}} J\right)(y)$ onto $\Re_{0}^{n+1}$.
6. Determine $h$ which minimizes $J\left(y-h \pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)\right)$.
7. Let $y^{n e w}=y-h \pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$.
8. If $\left\|y^{\text {new }}-y\right\|_{L}<\epsilon$ we have a solution; else, put $y=y^{\text {new }}$ and repeat steps 3 through 7.

In fact, this is not the method we use, although it is equivalent [N5]. The difference between this algorithm and the one which follows is the combination of steps 4 and

5 in order to solve for the quantity $\pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$ by solving simultaneously one system of $n+1$ equations. This allows us to avoid computing the projection $\pi_{H_{w}}$ directly. Depending on the boundary conditions associated with a problem, we may or may not know the projection, $\pi_{H_{w}}$. For example, if we are considering a problem with perturbation space $\left\{u: \int u=0\right\}$, then we have $\pi_{H_{w}} u=u-\int u$ if $\int 1=1$. For our case determining $\pi_{H_{w}}$ directly (to our knowledge) would require solving an additional $(n+1) \times(n+1)$ system and thus almost doubling the computations in the algorithm.

We now exhibit the method used to compute $\pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$. Let $u=\left(\nabla_{H_{w}} J\right)(y)$ and define $\gamma: \Re_{0}^{n+1} \rightarrow \Re$ by $\gamma(x)=\frac{1}{2}\|x-u\|_{H_{w}}^{2}$. Minimizing $\gamma$ over $\Re_{0}^{n+1}$ corresponds to determining $x \in \Re_{0}^{n+1}$ such that $x=\pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$. Let $\pi_{e}$ denote the orthogonal Euclidean projection onto $\Re_{0}^{n+1}$.

$$
\begin{gathered}
\gamma(x)=\frac{1}{2}\|x-u\|_{H_{w}}^{2}=\frac{1}{2}\left\|D_{w}(x-u)\right\|_{L}^{2} \\
\gamma^{\prime}(x)(g)=\left\langle D_{w}(x-u), D_{w}(g)\right\rangle=\left\langle\left(D_{w}\right)^{t} D_{w}(x-u), g\right\rangle \\
=\left\langle\left(D_{w}\right)^{t} D_{w}(x-u), \pi_{e}(g)\right\rangle=\left\langle\pi_{e}\left(D_{w}\right)^{t} D_{w}(x-u), g\right\rangle .
\end{gathered}
$$

Therefore $\gamma^{\prime}(x)(g)=0$ for every $g \in \Re_{0}^{n+1}$ if and only if $\pi_{e} D_{w}^{t} D_{w}(u)=\pi_{e} D_{w}^{t} D_{w}(x)$. Substituting $A_{w}=D_{w}^{t} D_{w}, u=\left(\nabla_{H_{w}} J\right)(y)$, and $A_{w}\left(\nabla_{H_{w}} J\right)(y)=\left(\nabla_{L} J\right)(y)$ into this equation yields $\pi_{e} A_{w} x=\pi_{e}\left(\nabla_{L} J\right)(y)$. The solution to this equation is the desired quantity, $x=\pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$.

Having avoided the direct computation of the projection $\pi_{H_{w}}$, we must still determine the projection $\pi_{e}$. Yet, we may compute $\pi_{e}$ easily by defining $\psi(x)=\|x-u\|_{L}{ }^{2} / 2$
and minimizing $\psi$ over $\Re_{0}^{n+1}$ to obtain

$$
\pi_{e}(x)=\left(\frac{k_{2}\left(k_{2} x_{1}-k_{1} x_{n+1}\right)}{k_{1}^{2}+k_{2}^{2}}, x_{2}, x_{3}, \ldots, x_{n}, \frac{-k_{1}\left(k_{2} x_{1}-k_{1} x_{n+1}\right)}{k_{1}^{2}+k_{2}^{2}}\right)
$$

Certain values of $k_{1}$ and $k_{2}$ cause numerical difficulties. For example, $\pi_{e}$ is not defined if $k_{1}=k_{2}=0$. If $k_{1}$ and $k_{2}$ are not both zero, $\pi_{e}$ is well defined, yet when we apply the projection to the matrix $A_{w}$, the first and last rows of the projected matrix are linearly dependent. Let us look at each of four possible cases. If $k_{1}=k_{2}=0$, we do not wish to use the projection as we have no boundary conditions; we are actually solving the problem outlined in Section 3.2.1 where $\pi_{e}$ was neither defined nor needed. If $k_{1}=0$ and $k_{2} \neq 0$ we are considering the final value problem, $y(b)=k_{3} / k_{2}$. Here, $\pi_{e}$ zeroes out the last row of the matrix $A_{w}$. Therefore we replace this row by the data $\left(0, \ldots, 0, k_{2}\right)$, and zero out the final entry of the gradient vector $\left(\nabla_{L} J\right)(y)$, before solving the system. If $k_{1} \neq 0$ and $k_{2}=0$, we are considering the initial value problem $y(a)=k_{3} / k_{1}$. In this case $\pi_{e}$ zeroes out the first row of the matrix $A_{w}$ and we replace this row by the data $\left(k_{1}, 0, \ldots, 0\right)$ and zero out the first entry of the gradient vector, $\left(\nabla_{L} J\right)(y)$, before solving the system. Finally if both $k_{1}$ and $k_{2}$ are non-zero then we replace the last row by the boundary data, $\left(k_{1}, 0, \ldots, 0, k_{2}\right)$ and zero out the last entry of the gradient vector, $\left(\nabla_{L} J\right)(y)$, before solving the system.

A revised algorithm follows.

1. Compute the matrix, $A_{w}$, and the projection, $\pi_{e}$.
2. Choose $y \in \Re_{0}^{n+1}$.
3. Compute the gradient of $J$ at $y,\left(\nabla_{L} J\right)(y)$.
4. Apply $\pi_{e}$ to the matrix, $A_{w}$, and the gradient, $\left(\nabla_{L} J\right)(y)$.
5. Make $A_{w}$ nonsingular by replacing the necessary rows.
6. Solve $\pi_{e} A_{w}\left(\pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)\right)=\pi_{e}\left(\nabla_{L} J\right)(y)$ for $\pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$.
7. Determine $h$ which minimizes $J\left(y-h \pi_{H_{\omega}}\left(\nabla_{H_{\omega}} J\right)(y)\right)$.
8. Let $y^{n e w}=y-h \pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$.
9. If $\left\|y^{\text {new }}-y\right\|_{L}<\epsilon$ then we have a solution; else, put $y=y^{\text {new }}$ and repeat steps 3 through 8 .

### 3.3 Examples

In this section we apply our algorithm to several problems. Our first example shows the reader how we predict the solution depending on the initial function estimate and choice of gradient in the case where boundary conditions are not sufficient to assure uniqueness. The second example adds boundary conditions to assure uniqueness. The third example demonstrates the effectiveness of the algorithm when the singularity is on the interior of the domain and our final example shows that even in cases where there is no singularity we can expect the weighted Sobolev descent to outperform Sobolev descent. We conclude this section and the chapter with a singular non-linear differential equation.

### 3.3.1 Unconstrained Singular Problem

Consider $t y^{\prime}=y$ on $I$ with no boundary conditions. An initial condition $y(0)=0$ is forced by the singularity, and using separation of variables we see that the one parameter family of solutions is given by $z(t)=k t$. Our functional is given by $J(u)=$ $\int_{I}\left(D_{j}^{1} u-u\right)^{2}$ for every $u \in H_{w}$.

We begin this section by predicting the solution to which the algorithm will converge based on the choice of the initial function and the gradient when boundary conditions are not sufficient to guarantee uniqueness.

Theorem 8 If $y_{0}$ is our initial estimate, steepest descent will converge to $z(t)=k t$ where $k_{L}=3 \int_{I} j y_{0}, k_{H}=\frac{1}{3} \int_{I}\left(j y_{0}+y_{0}^{\prime}\right)$, and $k_{H_{j}}=\frac{3}{2} \int_{I} j\left(y_{0}+j y_{0}^{\prime}\right)$ are determined by the choice of the gradient used in the descent process.

Proof. We prove only the statement associated with weighted descent. Suppose $J$ is as stated above and $\alpha(z)=\left\|y_{0}-z\right\|_{H_{w}}^{2}$. Observe that $\alpha(z)=\left\|y_{0}-z\right\|_{H_{w}}^{2}=$ $\left\|y_{0}\right\|_{H_{w}}^{2}+\|z\|_{H_{w}}^{2}-2\left\langle z, y_{0}\right\rangle_{H_{w}}$. Minimizing $\alpha$ over $S=\{z: z(t)=k t\}$ will yield the closest element in $H_{w} \cap S$. This is a quadratic equation yielding $k$ as stated. q.e.d.

To illustrate, we choose the initial function $y_{0}(t)=t^{2}$. We obtain the resulting solutions $z_{L}(t)=\frac{3}{4} t, z_{H}(t)=\frac{15}{16} t$, and $z_{H_{w}}(t)=\frac{9}{8} t$. The numerical results are in Tables 3.1 and 3.2. The number of divisions is small so that we may compare the Sobolev descent results with the $L$ results. After we have made our point that $L$ descent is clearly outperformed by Sobolev descent, we will omit the $L$ and $H$ results so that we can increase the number of divisions and accuracy desired.

Table 3.1: Unconstrained Singular Problem

| ty $^{\prime}-\mathbf{y}=\mathbf{0}$ |  | $y_{0}(t)=t^{2}$ | No Boundary Conditions |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |$\quad N=100$

Table 3.2: Unconstrained Singular Problem

| ty $-\mathbf{y}=\mathbf{0}$ |  | $y_{0}(t)=t^{2}$ | No Boundary Conditions |  |  |  | $N=10,000$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |  |  |
| $H_{w}$ | 36 | 45 | $10^{-15}$ | $10^{-10}$ | $8.7 \times 10^{-10}$ |  |  |

### 3.3.2 Constrained Singular Problem

If we add the boundary condition $y(1)=1$, guaranteeing the unique solution $y(t)=t$, we see similar results (see Tables 3.3 and 3.4 ). We use this example to demonstrate graphically the advantage of the weights near the origin. Figure 3.1 shows four lines shaded from light to dark and varying from thick to thin. Respectively they represent the intial estimate, the Sobolev approximation to the solution after three iterations, the weighted Sobolev approximation to the solution after three iterations, and the solution itself. The advantage of the weight near the singularity is clear from the graph.

### 3.3.3 Partially Constrained Singular Problem

Consider $\left(t-\frac{1}{2}\right) y^{\prime}=y$ with $y(0)=-\frac{1}{2}$. Demonstrating the versatility of the algorithm we consider a partially constrained example. Solutions are given by,

$$
z(t)= \begin{cases}k_{1}\left(t-\frac{1}{2}\right) & \text { if } x \in\left[0, \frac{1}{2}\right] \\ k_{2}\left(t-\frac{1}{2}\right) & \text { if } x \in\left[\frac{1}{2}, 1\right]\end{cases}
$$

Therefore since we have specified only an initial condition, the value for $k_{2}$ is not unique. As in the first example, we may determine the solution. If $y_{0}$ is our initial guess, our solution, $z$, will be the function which minimizes $\left\|y_{0}-z\right\|$ in whichever norm is chosen for steepest descent. Having previously made our point that $L$ descent is outperformed by both $H$, and $H_{w}$ descent, we omit these results here. This allows us to increase the number of divisions and the accuracy considerably. $H$ descent will


Figure 3.1: Constrained Singular Problem

Table 3.3: Constrained Singular Problem

| ty $^{\prime}-\mathbf{y}=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{1})=\mathbf{1}$ |  | $\mathbf{y}_{\mathbf{0}}(\mathbf{t})=\mathbf{1}$ | $\mathbf{N}=\mathbf{1 0 0}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $L$ | 20056 | 232 | $10^{-5}$ | $10^{-2}$ | $3.2 \times 10^{-1}$ |
| $H$ | 1346 | 18 | $10^{-5}$ | $10^{-4}$ | $6.7 \times 10^{-3}$ |
| $H_{w}$ | 8 | 1 | $10^{-5}$ | $10^{-7}$ | $2.2 \times 10^{-6}$ |

Table 3.4: Constrained Singular Problem

| ty $^{\prime}-\mathbf{y}=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{1})=\mathbf{1}$ | $\mathbf{y}_{0}(\mathbf{t})=\mathbf{1}$ | $\mathbf{N}=\mathbf{1 0}, \mathbf{0 0 0}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H_{w}$ | 24 | 49 | $10^{-15}$ | $10^{-16}$ | $4.2 \times 10^{-14}$ |

yield the solution above with $k_{1}=1$ and $k_{2}=\frac{3}{2}$ while $H_{w}$ descent will yield $k_{1}=1$ and $k_{2}=\frac{15}{4}$.

We make two points with this example using Tables 3.5 and 3.6. $H_{w}$ descent outperforms $H$ descent by a factor of 66 in time and by $10^{5}$ in accuracy. After increasing the number of divisions, we still have the time factor of 66 . However, we have an increase of $10^{7}$ in accuracy. This trend persists in all examples we have considered: as we increase the number of divisions we observe an increase in the differential between the obtainable accuracy.

It is worth observing that in the Table 3.6 we have a less strict stopping criteria for $H$ descent than for $H_{w}$ descent. This is the 'best' result obtainable for the $H$ descent. We are using optimal step size and we are unable obtain superior results to the ones listed since the order of magnitude of $\left(\nabla_{H} J\right)$ is $10^{-16}$ or machine precision. We end this section merely by pointing out that we have obtained similar results for problems with multiple singularities over the interval such as $\left(t-\frac{1}{4}\right)\left(t-\frac{3}{4}\right) y^{\prime}=y$ with an initial condition at any one of the interior points $t=0, t=\frac{1}{4}, t=\frac{3}{4}$, or $t=1$.

Table 3.5: Partially Constrained Singular Problem

| $\left(\mathbf{t}-\frac{\mathbf{1}}{2}\right) \mathbf{y}^{\prime}-\mathbf{y}=\mathbf{0}$ |  |  | $\mathbf{y}(\mathbf{0})=-\frac{\mathbf{1}}{\mathbf{2}}$ |  | $\mathbf{N}=\mathbf{1 0 0 0}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H$ | 526 | 66 | $10^{-5}$ | $10^{-2}$ | $1.5 \times 10^{-1}$ |
| $H_{w}$ | 7 | 1 | $10^{-5}$ | $10^{-7}$ | $4.3 \times 10^{-6}$ |

Table 3.6: Partially Constrained Singular Problem

| $\left(t-\frac{1}{2}\right) y^{\prime}-\mathrm{y}=0$ |  |  | $y(0)=-\frac{1}{2} \quad N$ |  | 10,000 |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| H | 764 | 497 | $10^{-6}$ | $10^{-3}$ | $3.4 \times 10^{-2}$ |
| $H_{w}$ | 13 | 17 | $10^{-10}$ | $10^{-10}$ | $1.8 \times 10^{-9}$ |

## CHAPTER 4

## SECOND ORDER PROBLEMS

We consider two second order problems which reduce to first order via variational principles thus allowing us to use the spaces constructed in Chapter 3.

$$
\begin{gather*}
t^{2} u^{\prime \prime}+2 t u^{\prime}-u=0 \text { on } I  \tag{4.1}\\
u(0)=0 \text { and } u(1)=1 \\
\left(1-t^{2}\right) u^{\prime \prime}-2 t u^{\prime}+u=0 \text { on } I  \tag{4.2}\\
u(0)=0 \text { and } u(1)=1
\end{gather*}
$$

In the section on Legendre's equation we obtain machine precision results using weighted descent which we are unable to obtain without the weights. This illustrates again the phenomenon shown in Chapter 3: Upgrading from $L$ descent to $H$ descent to $H_{w}$ descent not only results in a decrease in both time and the number of iterations required to solve the problem, but also in accuracies unobtainable by the previous methods.

We considered another numerical approach to this problem suggested in [N1]. The method used was to apply steepest descent directly to $J$, hoping that we did not 'fall off' the critical point. The alternative approach was to form the functional $\phi(u)=\frac{1}{2}\left\|\left(\nabla_{H_{w}} J\right)(u)\right\|^{2}$ whose zeroes are clearly critical points of $J$. We had successful results from each method, but the latter requires solving two systems of equations per iteration. Since neither had superior accuracy results and the alternative approach
was computationally inferior, we present only the former approach. It is noteworthy that for problems where the first method tends to 'fall off' the critical points, the latter method is appropriate and, surprisingly, requires minimal alteration (about 3 lines) of the original code. More information on this method can be found in [N2].

### 4.1 A Variational Problem

Consider solving $K u=0$ where $K$ is defined by $K u=\left(t^{2} u^{\prime}\right)^{\prime}-u$ and suppose we seek $u \in C_{I}^{0} \cap C_{(0,1\}}^{2}$ such that $u(0)=0$ and $u(1)=1$. Seeking out series solutions yields $u(t)=c_{1} t^{\frac{-1+\sqrt{5}}{2}}+c_{2} t^{\frac{-1-\sqrt{5}}{2}}$ where only the first summand satisfies the equation, boundary conditions, and space limitations; thus, we seek only this solution. Once again we are seeking a solution $t^{\frac{-1+\sqrt{5}}{2}} \in H_{w} \backslash H$, and consider descent based on subspaces of $L, H$, and $H_{w}$. The three subspaces based on our boundary conditions are $L^{0}:=\{h \in L: h(0)=0=h(1)\}, H^{0}=H \cap L^{0}$, and $H_{w}^{0}=H_{w} \cap L^{0}$. As all three functionals will agree on the space $C:=C_{I}^{0} \cap C_{(0,1]}^{2}$, we abuse the notation and label them all $J$, letting

$$
J(u)=\frac{1}{2} \int_{I} j^{2}\left(u^{\prime}\right)^{2}+u^{2} .
$$

The motivation can be summarized in one sentence if we ignore the boundary conditions for the moment and assume $u \in H_{w}$ where $w(t)=t$.

Since

$$
J^{\prime}(u)(h)=\int_{I} j^{2} u^{\prime} h^{t}+u h=\langle u, h\rangle_{H_{j}}
$$

it seems natural to seek a critical point of $J$ in the space $\left.\left(H_{j},\langle\cdot,\rangle\right\rangle_{H_{j}}\right)$.
In fact, we apply steepest descent based on a gradient which takes into consideration both the weight and the boundary conditions simultaneously as we outlined in Chapter 3. Because $L^{0}$ is a Hilbert space and $J$ is a bounded linear operator, the Reisz Representation Theorem guarantees existence of $\left(\nabla_{L^{0}} J\right)$ such that for $u \in C \subset L, h \in L^{0}$,

$$
\begin{aligned}
\left\langle\left(\nabla_{L^{0}} J\right)(u), h\right\rangle_{L^{0}} & =J^{\prime}(u)(h) \\
& =\int_{I} j^{2} u^{\prime} h^{\prime}+u h \\
& =\int_{I}\left(\left(-j^{2} u^{\prime}\right)^{\prime}+u\right) h \\
& =-\int_{I} h K u \\
& =\left\langle h,-P_{L} K u\right\rangle_{L^{0}}
\end{aligned}
$$

where $P_{L}: L \rightarrow L^{0}$ is the orthogonal projection.

The parallel in the Hilbert space $H^{0}$ is given by

$$
\begin{equation*}
J(u)=\frac{1}{2} \int_{I} j^{2}\left(D_{1} u\right)^{2}+u^{2} \tag{4.3}
\end{equation*}
$$

and for $u \in C, h \in H^{0}$ we have

$$
\begin{aligned}
\left\langle\left(\nabla_{H^{\circ}} J\right)(u), h\right\rangle_{H^{0}} & =J^{\prime}(u)(h) \\
& =\int_{I} j^{2} D_{1} u D_{1} h+u h \\
& =\left\langle\left(\begin{array}{c}
h \\
\\
D_{1} h
\end{array}\right),\binom{u}{j^{2} D_{1} u}\right\rangle_{L \times L} \\
& =\left\langle\binom{ h}{D_{1} h}, P_{H}\binom{u}{j^{2} D_{1} u}\right\rangle_{L \times L}
\end{aligned}
$$

$$
=\left\langle h, \pi_{1} P_{H}\binom{u}{j^{2} D_{1} u}\right\rangle_{H^{0}}
$$

where

$$
P_{H}: L \times L \rightarrow\left\{\binom{u}{D_{1} u}: u \in H^{0}\right\}
$$

is the orthogonal projection and $\pi_{1}: \Re \times \Re \rightarrow \Re$ such that $\pi_{1}\binom{\alpha}{\beta}=\alpha$.
As in Chapter 3, the weight we choose is the square root of the function in the functional which results from the singularity in the differential equation. In this case the singularity in the differential equation is $t^{2}$ which appears again in the functional. The parallel in the Hilbert space $H_{j}^{0}$ is given by

$$
\begin{equation*}
J(u)=\frac{1}{2} \int_{I}\left(D_{1}^{j} u\right)^{2}+u^{2} \tag{4.4}
\end{equation*}
$$

and for $u \in C, h \in L^{0}$ we have

$$
\begin{aligned}
\left\langle\left(\nabla_{H_{j}^{0}} J\right)(u), h\right\rangle_{H_{j}} & =J^{\prime}(u)(h) \\
& =\int_{I} D_{1}^{j} u D_{1}^{j} h+u h \\
& =\left\langle\binom{ h}{D_{1}^{j} h},\binom{u}{D_{1}^{j} u}\right\rangle_{L \times L} \\
& =\left\langle\binom{ h}{D_{1}^{j} h}, P_{H_{j}}\binom{u}{D_{1}^{j} u}\right\rangle_{L \times L} \\
& =\left\langle h, \pi_{1} P_{H_{j}}\binom{u}{D_{1}^{j} u}\right\rangle_{H_{j}^{0}}
\end{aligned}
$$

where

$$
P_{H_{j}}: L \times L \rightarrow\left\{\binom{u}{D_{1}^{j} u}: u \in H_{j}^{0}\right\}
$$

is the orthogonal projection.
We summarize this exposition with the following theorem.

Theorem 9 For all $u \in C$, the gradients with respect to the Hilbert spaces $L^{0}, H^{0}$, and $H_{j}^{0}$ are given by $\left(\nabla_{L^{\circ}} J\right)(u)=-P_{L} K u,\left(\nabla_{H^{0}} J\right)(u)=\pi_{1} P_{H}\binom{u}{j^{2} D_{1} u}$, and $\left(\nabla_{H_{w}^{0}} J\right)(u)=\pi_{1} P_{H_{j}}\binom{u}{D_{1}^{j} u}$.

Having put forth the continuous theory, we now discretize the problem. The question is: Which of the following equations $\left(\nabla_{L} J\right)(u)=0$ (Euler's equation), $\left(\nabla_{H} J\right)(u)=0$, or $\left(\nabla_{H_{w}} J\right)(u)=0$ is the appropriate equation to consider for computing on variational problems concerning singular differential equations?

Discretizing the functional,

$$
J(u)=\frac{1}{2} \sum_{k=1}^{n}\left(\frac{t_{k+1}+t_{k}}{2}\right)^{2}\left(\frac{u_{k+1}-u_{k}}{\delta}\right)^{2}+\left(\frac{u_{k+1}+u_{k}}{2}\right)^{2} .
$$

Since $\pi_{e}(x)=\left(0, x_{1}, \ldots, x_{n}, 0\right)$ and $\left(\nabla_{L} J\right)(u)=-K u$, we have

$$
\left(\nabla_{L^{\mathrm{o}}} J\right)(u)=\left(0, \ldots,-t_{k}^{2} \frac{u_{k-1}-2 u_{k}+u_{k+1}}{\delta^{2}}-2 t_{k} \frac{u_{k+1}-u_{k-1}}{\delta}+u_{k}, \ldots, 0\right)
$$

The following algorithm outlines both $H$ and $H_{w}$ steepest descent since if $w=1$, then $H_{w}=H$. We have not discussed solving the linear system here; the matrix itself is outlined in Chapter 6 and the boundary conditions are handled as in Chapter 3.

1. Compute the matrix, $A_{w}$, and the projection, $\pi_{e}$.
2. Choose $y \in \Re_{0}^{n+1}$.
3. Compute the gradient of $J$ at $y,\left(\nabla_{L} J\right)(y)$.
4. Apply $\pi_{e}$ to the matrix, $A_{w}$, and the gradient, $\left(\nabla_{L} J\right)(y)$.
5. Make $A_{w}$ nonsingular by replacing the necessary rows.
6. Solve $\pi_{e} A_{w} x=\pi_{e}\left(\nabla_{L} J\right)(y)$ for $x=\pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$.
7. Determine $h$ which minimizes $J\left(y-h \pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)\right)$.
8. Let $y^{\text {new }}=y-h \pi_{H_{w}}\left(\nabla_{H_{w}} J\right)(y)$.
9. If $\left\|y^{\text {new }}-y\right\|_{L}<\epsilon$ then we have a solution; else, put $y=y^{\text {new }}$ and repeat steps 3 through 8 .

We exhibit the difference between the weighted and non-weighted descent processes via Figure 4.1. The graph shows four curves. They are shaded from light to dark and vary from thick to thin. Respectively they represent the intial estimate, the Sobolev approximation to the solution after three iterations, the weighted Sobolev approximation to the solution after three iterations, and the solution itself. The advantage of the weight near the singularity is clear from the graph. The solution and the weighted Sobolev approximation to the solution are already virtually indistinguishable by three iterations. Tables 4.1, 4.2, and 4.3 represent the numerical results obtained using each of the above methods. First observe the decrease on both time and iterations required and the increase in both average absolute accuracy and maximum absolute accuracy. The time required to solve each problem is valid for comparison between these algorithms only. We state this because every attempt was made to make a fair
comparison between the three methods, with the author going so far as to use one code with flags for all three methods. For these reasons, these numbers should not be used for comparison to other methods, as no attempt was made by the author to make the codes efficient for solving any particular of problem. Considerable improvements in the time required to solve each problem could be made merely by separating the code into three codes and writing for efficiency as opposed to comparability and readability. We also note that traditional methods such as conjugate gradient methods or multi-step methods are applicable to all three descent processes. The algorithms were written to support two facts: First, weighted Sobolev steepest descent is a noteworthy numerical method and efficient codes using this methodology could be developed and, second, computing based on an alternative to the traditional Euler's equation should be considered for singular variational problems.

Our improved numerical results from the various processes were expected and a defense of the intuitive reasoning follows. Necessary conditions are given in [CH] in order that satisfying Euler's equation be a necessary condition for existence of an extremal point; however, our problem does not satisfy these conditions. This difficulty in the continuous case translates over to the poor numerical performance in solving $\left(\nabla_{L} J\right)=0$. Similarly, seeking the solution, $t^{\frac{-1+\sqrt{5}}{2}}$ which does not belong to the space $H$, makes solving $\left(\nabla_{H} J\right)=0$ an unpromising task. This leaves us with the equation $\left(\nabla_{H_{w}} J\right)=0$ which indeed performed the best.


Figure 4.1: Variational Problem

Table 4.1: Variational Problem

| $\mathbf{t}^{2} \mathbf{y}^{\prime \prime}+2 \mathbf{t y} \mathbf{y}^{\prime}-\mathbf{y}=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{0})=\mathbf{0}$ | $\mathbf{y}(\mathbf{1})=\mathbf{1}$ | $\mathbf{N}=\mathbf{1 0 0}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $L$ | 10,000 | 41 | $10^{-6}$ | $10^{-2}$ | $1.4 \times 10^{-1}$ |
| $H$ | 538 | 2 | $10^{-6}$ | $10^{-4}$ | $1.8 \times 10^{-2}$ |
| $H_{w}$ | 1 | 1 | $10^{-6}$ | $10^{-5}$ | $1.9 \times 10^{-3}$ |

Table 4.2: Variational Problem

| $\mathbf{t}^{2} \mathbf{y}^{\prime \prime}+2$ ty $^{\prime}-\mathbf{y}=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{0})=\mathbf{0}$ | $\mathbf{y}(\mathbf{1})=\mathbf{1}$ | $\mathbf{N}=\mathbf{1 , 0 0 0}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H$ | 151 | 4000 | $10^{-8}$ | $10^{-4}$ | $1.3 \times 10^{-2}$ |
| $H_{w}$ | 1 | 2 | $10^{-8}$ | $10^{-6}$ | $4.8 \times 10^{-4}$ |

Table 4.3: Variational Problem

| $\mathfrak{t}^{2} \mathbf{y}^{\prime \prime}+\mathbf{2 t y} \mathbf{y}^{\prime}-\mathbf{y}=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{0})=\mathbf{0}$ | $\mathbf{y}(\mathbf{1})=\mathbf{1}$ | $\mathrm{N}=\mathbf{1 0 0}, \mathbf{0 0 0}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H_{w}$ | 14 | 3 | $10^{-16}$ | $10^{-9}$ | $2.8 \times 10^{-6}$ |

### 4.2 Legendre's Equation of order $\alpha=1$

Consider the problem $K u=0$ where $K u=\left(\left(1-t^{2}\right) u^{\prime}\right)^{\prime}+2 u$ on $I$ with $u(0)=0$ (forced initial condition), $u(1)=1$, and $u \in C_{I}^{2}$. General solutions are $u(t)=c_{1} t+$ $\frac{c_{2}}{2} t \ln \left(\frac{1+t}{1-t}\right)$ and only $u(t)=t$ satisfies the boundary conditions. To obtain this solution, consider the functional

$$
J(u)=\frac{1}{2} \int_{I}\left(1-j^{2}\right)\left(u^{\prime}\right)^{2}+u^{2}
$$

and define the three distinct functionals which parallel those from the previous section. Let $L^{0}:=\{h \in L: h(0)=0=h(1)\}, H^{0}=H \cap L^{0}$, and $H_{w}^{0}=H_{w} \cap L^{0}$. For $u \in C_{I}^{2}$ and $h \in L^{0}$ we have

$$
\begin{aligned}
\left\langle\left(\nabla_{L^{0}} J\right)(u), h\right\rangle_{L^{0}} & =J^{\prime}(u)(h) \\
& =\int_{I}\left(1-j^{2}\right) u^{\prime} h^{\prime}+u h \\
& =\int_{I}\left(\left(-\left(1-j^{2}\right) u^{\prime}\right)^{\prime}+u\right) h \\
& =-\int_{I} h K u i \\
& =\left\langle h,-P_{L} K u\right\rangle_{L^{0}},
\end{aligned}
$$

where $P_{L}: L \rightarrow L^{0}$ is the orthogonal projection.
The parallel in $H$ is given by

$$
J(u)=\frac{1}{2} \int_{I}\left(1-j^{2}\right)\left(D_{1} u\right)^{2}+u^{2}
$$

and for $u \in C_{I}^{2}, h \in H^{0}$ we have

$$
\left\langle\left(\nabla_{H^{0}} J\right)(u), h\right\rangle_{H^{0}}=J^{\prime}(u)(h)
$$

$$
\begin{aligned}
& =\int_{I}\left(1-j^{2}\right) D_{1} u D_{1} h+u h \\
& =\left\langle\left(\begin{array}{c}
h \\
\\
D_{1} h
\end{array}\right),\binom{u}{\left(1-j^{2}\right) D_{1} u}\right\rangle_{L \times L} \\
& =\left\langle\binom{ h}{D_{1} h}, P_{H}\binom{u}{\left(1-j^{2}\right) D_{1} u}\right\rangle_{L \times L} \\
& =\left\langle h, \pi_{1} P_{H}\binom{u}{\left(1-j^{2}\right) D_{1} u}\right\rangle_{H^{0}}
\end{aligned}
$$

where

$$
P_{I I}: L \times L \rightarrow\left\{\binom{u}{D_{1} u}: u \in H^{0}\right\}
$$

is the orthogonal projection.
As in the previous section the weight we choose is the square root of the function in the functional which results from the singularity in the differential equation. In this case $w(t)=\sqrt{1-t^{2}}$.

The parallel in $H_{\sqrt{1-j^{2}}}$ is given by

$$
\begin{equation*}
J(u)=\frac{1}{2} \int_{I}\left(D_{1}^{\sqrt{1-j^{2}}} u\right)^{2}+u^{2} \tag{4.5}
\end{equation*}
$$

and for $u \in H_{\sqrt{1-j^{2}}}, h \in H_{\sqrt{1-j^{2}}}^{0}$, we have

$$
\begin{aligned}
\left\langle\left(\nabla_{H_{W}^{0}} J\right)(u), h\right\rangle & =J^{\prime}(u)(h) \\
& =\int_{I} D_{1}^{\sqrt{1-j^{2}}} u D_{1}^{\sqrt{1-j^{2}}} h+u h \\
& =\left\langle\binom{ h}{D_{1}^{\sqrt{1-j^{2}}} h},\binom{u}{D_{1}^{\sqrt{1-j^{2}}} u}\right\rangle_{L \times L} \\
& =\left\langle\binom{ h}{D_{1}^{\sqrt{1-j^{2}}} h}, P_{H}\binom{u}{D_{1}^{\sqrt{1-j^{2}}} u}\right\rangle_{L \times L}
\end{aligned}
$$

$$
=\left\langle h, P_{H \sqrt{1-j^{2}}} \pi_{1}\binom{u}{D_{1}^{\sqrt{1-j^{2}}} u}\right\rangle_{H_{\sqrt{1-j^{2}}}^{0}}
$$

where

$$
P_{H_{\sqrt{1-j^{2}}}}: L \times L \rightarrow\left\{\binom{u}{D_{1}^{\sqrt{1-j^{2}}} u}: u \in H_{\sqrt{1-j^{2}}}^{0}\right\}
$$

is the orthogonal projection. Discretizing the functional,

$$
J(u)=\frac{1}{2} \sum_{k=1}^{n}\left(1-\left(\frac{t_{k+1}+t_{k}}{2}\right)^{2}\right)\left(\frac{u_{k+1}-u_{k}}{\delta}\right)^{2}+\left(\frac{u_{k+1}+u_{k}}{2}\right)^{2}
$$

Since $\pi_{e}(x)=\left(0, x_{1}, \ldots, x_{n}, 0\right)$ and $\left(\nabla_{L} J\right)(u)=-K u$, we have

$$
\begin{aligned}
& \left(\nabla_{L^{\circ}} J\right)(u)= \\
& \quad\left(0, \ldots-\left(1-t_{k}^{2}\right) \frac{u_{k-1}-2 u_{k}+u_{k+1}}{\delta^{2}}+2 t_{k} \frac{u_{k+1}-u_{k-1}}{\delta}-2 u_{k}, \ldots, 0\right) .
\end{aligned}
$$

Tables 4.4, 4.5, and 4.6 demonstrate the success associated with these problems. The algorithm is parallel to the one from the preceding section.

Table 4.4: Legendre's Equation

| $\left(\mathbf{1}-\mathbf{t}^{2}\right) \mathbf{y}^{\prime \prime}-\mathbf{2 t y}+\mathbf{2 y}=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{0})=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{1})=\mathbf{1}$ | $\mathrm{N}=100$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $L$ | 5948 | 24 | $10^{-6}$ | $10^{-1}$ | $6.6 \times 10^{-1}$ |
| $H$ | 1998 | 7 | $10^{-6}$ | $10^{-6}$ | $3.7 \times 10^{-5}$ |
| $H$ | 64 | 1 | $10^{-6}$ | $10^{-7}$ | $8.0 \times 10^{-6}$ |

Table 4.5: Legendre's Equation

| $\left(\mathbf{1}-\mathbf{t}^{2}\right) \mathbf{y}^{\prime \prime}-\mathbf{2 t y}+\mathbf{2 y}=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{0})=\mathbf{0}$ | $\mathbf{y}(\mathbf{1})=\mathbf{1}$ | $\mathbf{N}=\mathbf{1 0}, \mathbf{0 0 0}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H$ | 2142 | 82 | $10^{-6}$ | $10^{-6}$ | $3.4 \times 10^{-5}$ |
| $H_{w}$ | 85 | 3 | $10^{-6}$ | $10^{-6}$ | $1.2 \times 10^{-5}$ |

Table 4.6: Legendre's Equation

| $\left(\mathbf{1}-\mathbf{t}^{2}\right) \mathbf{y}^{\prime \prime}-\mathbf{2 t y}+\mathbf{2}=\mathbf{y}=\mathbf{y}(\mathbf{0})=\mathbf{0}$ |  | $\mathbf{y}(\mathbf{1})=\mathbf{1}$ | $\mathrm{N}=\mathbf{1 0 0}, \mathbf{0 0 0}$ |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H_{w}$ | 325 | 125 | $10^{-15}$ | $10^{-14}$ | $1.7 \times 10^{-14}$ |

## CHAPTER 5

## A PARTIAL DIFFERENTIAL EQUATION

In this chapter we demonstrate that our method extends to partial differential equations by considering a partial differential equation with linear singularity in each variable.

### 5.1 The problem

Let $\Omega=I \times I$ and consider the problem of finding $u \in C^{1}$ satisfying $K u=0$ on $\Omega$ where $K u=x u_{1}+y u_{2}$. The boundary conditions are given by $u(x, 0)=1=u(0, y)$ for every $(x, y) \in \Omega$. Our approach will be much the same as in the previous chapters. We will consider three spaces $L, H$ and $H_{w}$ which have both continuous and discrete definitions on the square disk $\Omega$. Again we will consider descent based on each of the three spaces which parallel the spaces from Chapter 4.

### 5.2 Notation

We take a direct approach, considering the functional

$$
J(u)=\frac{1}{2} \int_{\Omega}\left(x u_{1}+y u_{2}\right)^{2}
$$

whose zeroes are solutions of the equation.
In previous chapters we defined a generalized derivative and a generalized weighted
derivative. Here we define a generalized gradient and a generalized weighted gradient. Let $L=L_{\Omega}^{2}$, and define $\pi_{1}, \pi_{2}: \Re \times \Re \rightarrow \Re$ such that $\pi_{1}\binom{\alpha}{\beta}=\alpha$ and $\pi_{2}\binom{\alpha}{\beta}=\beta$. If $\nabla=\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$, then define

$$
\begin{gathered}
\left.G=\left\{\begin{array}{c}
u \\
\nabla u
\end{array}\right): u \in C_{\Omega}^{\mathrm{L}}\right\}^{2} \\
H=\pi_{1} G \\
E_{0}(u)=u \\
E_{1}(u)=\pi_{1} G u \\
E_{2}(u)=\pi_{2} G u \\
\langle u, v\rangle_{H}=\langle u, v\rangle_{L}+\left\langle E_{1}(u), E_{1}(v)\right\rangle_{L}+\left\langle E_{2}(u), E_{2}(v)\right\rangle_{L}
\end{gathered}
$$

Thus, $G u=\left(E_{1} u, E_{2} u\right)$ denotes the generalized gradient and we will let $G_{w}$ denote the generalized weighted gradient. Define $\nabla_{w}=\left(j \frac{\partial}{\partial x}, k \frac{\partial}{\partial y}\right)$ where $j(x, y)=x$ and $k(x, y)=y$ and

$$
\begin{gathered}
G_{w}={\left.\overline{\left\{\binom{u}{\nabla_{w} u}: u \in C_{\Omega}^{1}\right.}\right\}^{L \times(L \times L)}}^{H_{w}=\pi_{1} G_{w}} \\
E_{0}(u)=u \\
E_{1}^{x}(u)=\pi_{1} G_{w} u
\end{gathered}
$$

$$
\begin{gathered}
E_{2}^{y}(u)=\pi_{2} G_{w} u . \\
\langle u, v\rangle_{H}=\langle u, v\rangle_{L}+\left\langle E_{1}^{x}(u), E_{1}^{x}(v)\right\rangle_{L}+\left\langle E_{2}^{y}(u), E_{2}^{y}(v)\right\rangle_{L}
\end{gathered}
$$

Thus, $G_{w} u=\left(E_{1}^{x} u, E_{2}^{y} u\right)$ is the generalized weighted gradient. In order to develop a gradient which depends on our constraints, put

$$
L^{0}=\{u \in L: u(x, 0)=0=u(0, y) \forall(x, y) \in \Omega\},
$$

$H^{0}=H \cap L^{0}$, and $H_{w}^{0}=H_{w} \cap L^{0}$. In the Hilbert space, $H^{0}, J$ is given by

$$
\begin{equation*}
J(u)=\frac{1}{2} \int_{\Omega}\left(x E_{1} u+y E_{2} u\right)^{2} \tag{5.1}
\end{equation*}
$$

and for $u, h \in H^{0}$ we have

$$
\begin{aligned}
\left\langle\left(\nabla_{H^{0}} J\right)(u), h\right\rangle_{H^{0}} & =J^{\prime}(u)(h) \\
& =\int_{\Omega}\left(x E_{1} u+y E_{2} u\right)\left(x E_{1} h+y E_{2} h\right) \\
& =\left\langle\left(\begin{array}{c}
0 \\
x^{2} E_{1} u+x y E_{2} u \\
y^{2} E_{2} u+x y E_{1} u
\end{array}\right),\left(\begin{array}{c}
h \\
E_{1} h \\
E_{2} h
\end{array}\right)\right\rangle_{L \times(L \times L)} \\
& =\left\langle h, \pi_{1} P_{H}\left(\begin{array}{c}
0 \\
x^{2} E_{1} u+x y E_{2} u \\
y^{2} E_{2} u+x y E_{1} u
\end{array}\right)_{H^{0}}\right.
\end{aligned}
$$

where

$$
P_{H}: L \times(L \times L) \rightarrow\left\{\binom{u}{G u}: u \in H^{0}\right\}
$$

is the orthogonal projection.
In the Hilbert space, $H_{w}^{0}, J$ is given by

$$
\begin{equation*}
J(u)=\frac{1}{2} \int_{\Omega}\left(E_{1}^{x} u+E_{2}^{y} u\right)^{2} \tag{5.2}
\end{equation*}
$$

and for $u, h \in H_{w}^{0}$ we have

$$
\begin{aligned}
\left\langle\left(\nabla_{H_{w}^{0}} J\right)(u), h\right\rangle_{H_{w}^{0}} & =J^{\prime}(u)(h) \\
& =\int_{\Omega}\left(E_{1}^{x} u+E_{2}^{y} u\right)\left(E_{1}^{x} h+E_{2}^{y} h\right) \\
& =\left\langle\left(\begin{array}{c}
0 \\
E_{1}^{x} u+E_{2}^{y} u \\
E_{1}^{x} u+E_{2}^{y} u
\end{array}\right),\left(\begin{array}{c}
h \\
E_{1}^{x} h \\
E_{2}^{y} h
\end{array}\right)\right\rangle \\
& =\langle h \times(L \times L) \\
& \left.=\left\langle\begin{array}{c}
0 \\
K u \\
K \\
K u
\end{array}\right)\right\rangle_{H_{\omega}^{0}}
\end{aligned}
$$

where

$$
P_{H_{w}}: L \times(L \times L) \rightarrow\left\{\binom{u}{G_{w} u}: u \in H_{w}^{0}\right\}
$$

is the orthogonal projection.

### 5.3 Numerics

For simplicity, we subdivide $\Omega$ into $n$ pieces along each axis. We order our grid starting in the lower left hand corner at $(0,0)$ so that the value of $u_{i, j}=u\left(\frac{1}{n}(i-\right.$ 1), $\left.\frac{1}{n}(j-1)\right)$.

For the discrete case, $E_{0}, E_{1}, E_{2}, E_{1}^{x}$, and $E_{2}^{y}$ are defined below.

$$
\begin{align*}
& \left\{E_{0}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{1}{4}\left(u_{i, j}+u_{i+1, j}+u_{i, j+1}+u_{i+1, j+1}\right) \\
& \left\{E_{1}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{n}{2}\left(-u_{i, j}+u_{i+1, j}-u_{i, j+1}+u_{i+1, j+1}\right) \\
& \left\{E_{2}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{n}{2}\left(-u_{i, j}-u_{i+1, j}+u_{i, j+1}+u_{i+1, j+1}\right) \\
& \left\{E_{1}^{x}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{n}{4}\left(x_{i}+x_{i+1}\right)\left(-u_{i, j}+u_{i+1, j}-u_{i, j+1}+u_{i+1, j+1}\right) \\
& \left\{E_{2}^{y}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{n}{4}\left(y_{j}+y_{j+1}\right)\left(-u_{i, j}-u_{i+1, j}+u_{i, j+1}+u_{i+1, j+1}\right) \\
& E(u)=\left(\begin{array}{c}
E_{0}(u) \\
E_{1}(u) \\
\\
E_{2}(u)
\end{array}\right) \quad E_{w}(u)=\left(\begin{array}{c}
E_{0}(u) \\
E_{1}^{x}(u) \\
E_{2}^{y}(u)
\end{array}\right) \tag{5.3}
\end{align*}
$$

We caution the reader that while we are subscripting each of $E_{0}(u), E_{1}^{x}(u), E_{1}^{x}(u)$, and $u$ as though they were matrices, we are treating each as a vector. For example,

$$
u=\left(\begin{array}{c}
u_{1,1}  \tag{5.4}\\
u_{2,1} \\
\vdots \\
u_{n+1, n+1}
\end{array}\right) \& \quad E_{0}(u)=\left(\begin{array}{c}
\left\{E_{0}(u)\right\}_{1,1} \\
\left\{E_{0}(u)\right\}_{2,1} \\
\vdots \\
\\
\left\{E_{0}(u)\right\}_{n+1, n+1}
\end{array}\right)
$$

Discretizing our functional,

$$
\begin{aligned}
J(u) & =\frac{1}{2} \int_{\Omega}\left(x u_{1}+y u_{2}\right)^{2} \\
& =\frac{1}{2} \int_{\Omega}\left(E_{1}^{x} u+E_{2}^{y} u\right)^{2} \\
& \cong \frac{1}{2 n^{2}} \sum_{i, j}^{n}\left(\left\{E_{1}^{x} u\right\}_{i, j}+\left\{E_{2}^{y} u\right\}_{i, j}\right)^{2}
\end{aligned}
$$

As before we compute $\nabla J$, the discrete analog to $\left(\nabla_{L} J\right)$.

$$
\nabla J(u)=\left(\begin{array}{c}
\frac{\partial J}{\partial u_{1,1}} \\
\frac{\partial J}{\partial u_{2,1}} \\
\vdots \\
\frac{\partial J}{\partial u_{n+1, n+1}}
\end{array}\right)
$$

Looking closely at the definition for $J$, only four summands contribute to $\frac{\partial J}{\partial u_{p, q}}$.

$$
\begin{aligned}
\frac{\partial J}{\partial u_{p, q}}=\frac{1}{2 n^{2}} \frac{\partial}{\partial u_{p, q}}\{ & {\left[\left(E_{1}^{x} u\right)_{p-1, q-1}+\left(E_{2}^{y} u\right)_{p-1, q-1}\right]^{2} } \\
& +\left[\left(E_{1}^{x} u\right)_{p, \quad q-1}+\left(E_{2}^{y} u\right)_{p, \quad},{ }^{2}\right]^{2} \\
& +\left[\left(E_{1}^{x} u\right)_{p-1, q}+\left(E_{2}^{y} u\right)_{p-1, q}\right]^{2} \\
& \left.+\left[\left(E_{1}^{x} u\right)_{p, \quad q}+\left(E_{2}^{y} u\right)_{p, \quad q}\right]^{2}\right\}
\end{aligned}
$$

The necessary partials are computed below.

$$
\begin{array}{ll}
\frac{\partial}{\partial u_{p, q}}\left(E_{1}^{x} u\right)_{p-1, q-1}=\frac{n}{4}\left(x_{p-1}+x_{p}\right) & \frac{\partial}{\partial u_{p, q}}\left(E_{1}^{x} u\right)_{p-1, q}=\frac{n}{4}\left(x_{p-1}+x_{p}\right) \\
\frac{\partial}{\partial u_{p, q}}\left(E_{1}^{x} u\right)_{p, q-1}=-\frac{n}{4}\left(x_{p}+x_{p+1}\right) & \frac{\partial}{\partial u_{p, q}}\left(E_{1}^{x} u\right)_{p, q}=-\frac{n}{4}\left(x_{p}+x_{p+1}\right) \\
\frac{\partial}{\partial u_{p, q}}\left(E_{2}^{y} u\right)_{p-1, q-1}=\frac{n}{4}\left(y_{q-1}+y_{q}\right) & \frac{\partial}{\partial u_{p, q}}\left(E_{2}^{y} u\right)_{p-1, q}=-\frac{n}{4}\left(y_{q}+y_{q+1}\right) \\
\frac{\partial}{\partial u_{p, q}}\left(E_{2}^{y} u\right)_{p, q-1}=\frac{n}{4}\left(y_{q-1}+y_{q}\right) & \frac{\partial}{\partial u_{p, q}}\left(E_{2}^{y} u\right)_{p, q}=-\frac{n}{4}\left(y_{q}+y_{q+1}\right)
\end{array}
$$

Substituting these partial into the previous equation yields:

$$
\begin{aligned}
& \frac{\partial J}{\partial u_{p, q}}= \\
& \frac{1}{n^{2}}\{ {\left[\left(E_{1}^{x} u\right)_{p-1, q-1}+\left(E_{2}^{y} u\right)_{p-1, q-1}\right]\left[\left[\frac{n}{4}\left(x_{p-1}+x_{p}\right)+\frac{n}{4}\left(y_{q-1}+y_{q}\right)\right]+\right.} \\
& {\left[\left(E_{1}^{x} u\right)_{p, \quad q-1}+\left(E_{2}^{y} u\right)_{p,} \quad q_{-1}\right]\left[-\frac{n}{4}\left(x_{p}+x_{p+1}\right)+\frac{n}{4}\left(y_{q-1}+y_{q}\right)\right]+} \\
& {\left[\left(E_{1}^{x} u\right)_{p-1, q}+\left(E_{2}^{y} u\right)_{p-1, q}\right]\left[\left[\frac{n}{4}\left(x_{p-1}+x_{p}\right)-\frac{n}{4}\left(y_{q}+y_{q+1}\right)\right]+\right.} \\
& {\left.\left[\left(E_{1}^{x} u\right)_{p,}+\left(E_{2}^{y} u\right)_{p, \quad}\right]\left[-\frac{n}{4}\left(x_{p}+x_{p+1}\right)-\frac{n}{4}\left(y_{q}+y_{q+1}\right)\right]\right\} }
\end{aligned}
$$

As in Chapter 3 , having computed $\left(\nabla_{L} J\right)(u)$ we proceed to solve the discrete analog to the system $B\left(\nabla_{H} J\right)(u)=\left(\nabla_{L} J\right)(u)$ or the system $B_{w}\left(\nabla_{H_{w}} J\right)(u)=$ $\left(\nabla_{L} J\right)(u)$, depending on whether we choose descent or weighted descent. In Theorem 7 Chapter 3 , we demonstrated the existence of a matrix relating two Euclidean spaces with differing norms. We have the parallel here for our spaces, defining

$$
B=\left(E_{0}\right)^{t} E_{0}+\left(E_{1}\right)^{t} E_{1}+\left(E_{2}\right)^{t} E_{2}
$$

and

$$
B_{w}=\left(E_{0}\right)^{t} E_{0}+\left(E_{1}^{x}\right)^{t} E_{1}^{x}+\left(E_{2}^{y}\right)^{t} E_{2}^{y}
$$

The interested reader may see Chapter 6 for a complete derivation of the matrices $B$ and $B_{w} . B$ is a diagonally dominant, non-singular operator. Furthermore $B$ is band diagonal with nine non-zero diagonals. Rewriting the system in order to solve via a Gaus-Seidel iterative scheme, we see that $u_{i, j}$ receives contributions from eight of the nine non-zero diagonals which correspond to the eight neighbors of $u_{i, j}$. More precisely, although special consideration must be given to the boundary of $\Omega$, for an interior point $u_{i, j}$ we have

$$
\begin{aligned}
& u_{i, j}= \\
& \quad\left(\left\{\left(\nabla_{L} J\right)(u)\right\}_{i, j}-B_{i-1, j-1} u_{i-1, j-1}+B_{i-1, j} u_{i-1, j}+\right. \\
& \quad B_{i-1, j+1} u_{i-1, j+1}+B_{i, j-1} u_{i, j-1}+B_{i, j+1} u_{i, j+1}+ \\
& \left.\quad B_{i+1, j-1} u_{i+1, j-1}+B_{i+1, j} u_{i+1, j}+B_{i+1, j+1} u_{i+1, j+1}\right) / B_{i, j}
\end{aligned}
$$

As there are simple patterns for the elements of $B$, we have an efficient algorithm for solving the band diagonal system. The following section illustrates once again the improvements in time, iterations, and accuracy obtainable by considering the weighted spaces.

### 5.4 Results

The following section offers preliminary results on work in progress. Optimal step size is not implemented in this section, hence the longer run times. We will observe significant improvements in the obtainable accuracy in this section by utilizing the weighted steepest descent at we have in each previous chapter.

We demonstrate in Tables 5.1 and 5.2 the significant increase in computation required for the singular problems by comparing the time required to solve the problems

$$
\begin{align*}
& u_{1}+u_{2}=0  \tag{5.5}\\
& u(x, 0)=1=u(0, y)
\end{align*}
$$

and

$$
\begin{align*}
& x u_{1}+y u_{2}=0  \tag{5.6}\\
& u(x, 0)=1=u(0, y) .
\end{align*}
$$

The results utilize standard Sobolev steepest descent. We observe in Tables 5.1 and 5.2 a factor of five in the length of time and a factor of ten in the number of iterations required to solve the singular problem, despite the fact that each problem has the constant function one as a solution. We use such a small grid to show the reader one of the more valuable attributes of the algorithm. Being able to compute quickly on a small grid and still obtain results which will be representative of the results on a larger grid makes such an algorithm very easy to test and modify quickly.

Table 5.3 offers the comparison between weighted and non-weighted Sobolev steepest descent on the singular problem.

Table 5.1: Non-Singular Partial Differential Equation

| $\mathbf{u}_{\mathbf{1}}+\mathbf{u}_{\mathbf{2}}=\mathbf{0}$ |  | $\mathbf{u}(\mathbf{x}, \mathbf{0})=\mathbf{1}=\mathbf{u}(\mathbf{0}, \mathbf{y})$ |  | Grid $=\mathbf{5} \times \mathbf{5}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H$ | 290 | 5 | $10^{-6}$ | $10^{-5}$ | $3.6 \times 10^{-4}$ |

Table 5.2: Singular Partial Differential Equation

| $\mathrm{xu}_{1}+\mathrm{yu}_{2}=0$ |  | $\mathrm{u}(\mathrm{x}, 0)=1=\mathrm{u}(0, \mathrm{y})$ |  | Grid $=5 \times 5$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Seconds | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| H | 2898 | 27 | $10^{-6}$ | $10^{-4}$ | $1.0 \times 10^{-4}$ |

Table 5.3: Singular Partial Differential Equation

| $\mathbf{x u}_{\mathbf{1}}+\mathbf{y u}_{\mathbf{2}}=\mathbf{0}$ |  | $\mathbf{u}(\mathbf{x}, \mathbf{0})=\mathbf{1}=\mathbf{u}(\mathbf{0}, \mathbf{y})$ |  | Grid $=\mathbf{1 0} \times \mathbf{1 0}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gradient | Iterations | Minutes | Residual | Avg. Abs. Err. | Max. Abs. Err. |
| $H$ | 88,801 | 85 | $10^{-10}$ | $10^{-6}$ | $1.9 \times 10^{-5}$ |
| $H_{w}$ | 20,593 | 66 | $10^{-10}$ | $10^{-6}$ | $5.5 \times 10^{-6}$ |

## CHAPTER 6

## MATRICES OF STEEPEST DESCENT

Throughout this chapter we will show only the case for weighted descent as nonweighted descent is the special case of weighted descent with weight, one. While we view $A$ and $B$ as matrices in this chapter, we have noted that in practice for ordinary differential equations, $A$ is tridiagonal and we store only the three non-zero diagonal vectors while for partial differential equations $B$ is band diagonal (nine nonzero bands) and we compute these non-zero elements as needed in order to solve the systems via a Gaus Seidel iterative process.

### 6.1 Matrix for First Order Equations

Recall the necessary definitions from Chapter 3. If $x, w \in \Re^{n+1}$ then
$D_{0}: \Re^{n+1} \rightarrow \Re^{n}, D_{1}^{w}: \Re^{n+1} \rightarrow \Re^{n}$, and $D_{w}: \Re^{n+1} \rightarrow \Re^{2 n}$ are defined by

$$
D_{0}(x)=\left(\begin{array}{c}
\frac{x_{1}+x_{2}}{2} \\
\vdots \\
\frac{x_{n}+x_{n+1}}{2}
\end{array}\right), D_{1}^{w}(x)=\left(\begin{array}{c}
\left(\frac{w+w_{2}}{2}\right)\left(\frac{x_{2}-x_{1}}{\delta}\right) \\
\vdots \\
\left(\frac{w_{n}+w_{n+1}}{2}\right)\left(\frac{x_{n+1}-x_{n}}{\delta}\right)
\end{array}\right), D_{w}(x)=\binom{D_{0}(x)}{D_{1}^{w}(x)} .
$$

Define $D_{1}=D_{1}^{w}$ where $w(t)=1$ so that $D_{1}$ is the usual discrete differential operator. Define pointwise multiplication $\odot$, by $(x, y, z) \odot(r, s, t)=(x r, y s, z t)$. We define two elementary matrices $M_{0}$ and $M_{1}$ which will be the building blocks for both the matrix $A$ from Chapter 3 and $B$ from Chapter 5 . We now restrict ourselves to
the case $n=3$.

$$
M_{0}=\left(\begin{array}{cccc}
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1
\end{array}\right) \quad \& \quad M_{1}=\left(\begin{array}{cccc}
-1 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 \\
0 & 0 & -1 & 1
\end{array}\right)
$$

Viewing $D_{0}$ and $D_{1}$ as matrices in terms of $M_{0}$ and $M_{1}$

$$
D_{0}=\frac{1}{2} M_{0}, \quad D_{1}=\frac{n}{2} M_{1}
$$

thus the matrix without weights, $A$, may be rewritten in terms of $M_{0}$ and $M_{1}$ as

$$
A=D^{t} D=D_{0}^{t} D_{0}+D_{1}^{t} D_{1}=\frac{1}{4} M_{0}^{t} M_{0}+\frac{n^{2}}{4} M_{1}^{t} M_{1}
$$

We make this observation now because we will refer back to these basic building blocks $M_{0}$ and $M_{1}$ in the section concerning matrices associated with partial differential equation problems. Returning to the weighted case, this allows us to rewrite $D_{1}^{w}$ in terms of $D_{1}, D_{0}$,

$$
D_{1}^{w}(u)=D_{0}(w) \odot D_{1}(u)
$$

Starting with the proof of Theorem 7 in Chapter 3 we sce

$$
\begin{align*}
\left\{A_{w}\right\}_{i, j} & =\left\{e_{j}, e_{i}\right\rangle_{H_{w}} \\
& =\left\langle D_{0} e_{j}, D_{0} e_{i}\right\rangle+\left\langle D_{\mathbf{1}}^{w} e_{j}, D_{1}^{w} e_{i}\right\rangle  \tag{6.1}\\
& =\left\langle D_{0} e_{j}, D_{0} e_{i}\right\rangle+\left\langle D_{0}(w) \odot D_{1} e_{j}, D_{0}(w) \odot D_{1} e_{i}\right\rangle
\end{align*}
$$

If we let $n=3$ this yields

$$
\begin{aligned}
& A_{w}= \\
& \left(\begin{array}{cccc}
\frac{1}{4}+\frac{\left(q_{0}+q_{1}\right)^{2}}{4 \delta^{2}} & \frac{1}{4}-\frac{\left(q_{q}+q_{1}\right)^{2}}{4 \delta^{2}} & 0 & 0 \\
\frac{1}{4}-\frac{\left(q_{0}+q_{1}\right)^{2}}{4 \delta^{2}} & \frac{1}{2}+\frac{\left(q_{0}+q_{1}\right)^{2}}{4 \delta^{2}}+\frac{\left(q_{1}+q_{2}\right)^{2}}{4 \delta^{2}} & \frac{1}{4}-\frac{\left(q_{1}+q_{2}\right)^{2}}{4 \delta^{2}} & 0 \\
0 & \frac{1}{4}-\frac{\left(q_{1}+q_{2}\right)^{2}}{4 \delta^{2}} & \frac{1}{2}+\frac{\left(q_{1}+q_{2}\right)^{2}}{4 \delta^{2}}+\frac{\left(q_{2}+q_{3}\right)^{2}}{4 \delta^{2}} & \frac{1}{4}-\frac{\left(q_{2}+q_{\xi}\right)^{2}}{4 \delta^{2}} \\
0 & 0 & \frac{1}{4}-\frac{\left(q_{2}+q_{5}\right)^{2}}{4 \delta^{2}} & \frac{1}{4}+\frac{\left(q_{2}+q_{3}\right)^{2}}{4 \delta^{2}}
\end{array}\right) .
\end{aligned}
$$

which is easily verified by simple computations with Equation 6.1 or by considering the following Mathematica code.

## MATHEMATICA CODE

```
one}={1,1,1,1
Array[q,4]
q={q0,q1,q2,q3}
od =1/d
Array[x,3]
Array[y,3]
times[x-, y-]:= {x[[1]] y[[1]],x[[2]] y[[2]],x[[3]] y[[3]]}
Array[e,4,4]
e= IdentityMatrix[4]
h=1/2
DO = h**{{1,1,0,0},{0,1,1,0},{0,0,1,1} }
D1= od * {{-1,1,0,0},{0,-1,1,0},{0,0,-1,1}}
A=Table[
Simplify[
                                    D0.e[[j]].D0.e[[i]] +
                                    times[D0.q,D1.e[[j]]].times[D0.q,D1.e[[i]]]
            I
    ,{i,4},{j,4}]
    Print["A=",A]
```


### 6.2 Matrix for Second Order Equations

We make a brief observation concerning second order problems for which there is no variational principle. While none is considered in this paper, we have also studied second order differential equations which do not have a variational approach. We wish to point out how simply the matrices extend to this case. Suppose we are considering the problem $y^{\prime \prime}=f\left(t, y, y^{\prime}\right)$ on $I$ as a system of two first order equations

$$
\binom{u^{\prime}}{v^{\prime}}=\left(\begin{array}{c}
f_{1}(t, u, v) \\
\\
f_{2}(t, u, v)
\end{array}\right) .
$$

The space would then be $\left(\left(\Re^{n+1}\right)^{2},\langle\cdot\rangle,\right)$ and the inner product would be defined by

$$
\begin{aligned}
\left\langle\binom{ u}{v},\binom{x}{y}\right\rangle_{H_{w}} & =\langle u, x\rangle_{H_{w}}+\langle v, y\rangle_{H_{w}} \\
& =\left\langle D_{0} u, D_{0} x\right\rangle_{L}+\left\langle D_{1} u, D_{1} x\right\rangle_{L}+\left\langle D_{0} v, D_{0} y\right\rangle_{L}+\left\langle D_{1} v, D_{1} y\right\rangle_{L} \\
& =\langle u, A x\rangle_{L}+\left\langle v, A_{w} y\right\rangle_{L} \\
& =\left\langle\binom{ u}{v},\binom{A x}{A_{w} y}\right\rangle_{L} \\
& =\left\langle\binom{ u}{v},\left(\begin{array}{cc}
A & 0 \\
0 & A_{w}
\end{array}\right)\binom{x}{y}\right\rangle_{L}
\end{aligned}
$$

Thus once the effort has been made to compute using the weighted gradients for the first order codes, the second order problem is as simple as solving two systems.

### 6.3 Matrix for Partial Differential Equations

The ordering of the grid points is crucial to the following definitions, so recall that we order the grid as a vector starting with the lower left hand corner at $(0,0)$ and
listing $u$ as the vector,

$$
u=\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)
$$

where $u_{1,1}=u(0,0)$ and $u_{i, j}=u\left(\frac{1}{n}(i-1), \frac{1}{n_{2}}(j-1)\right)$.
In the previous section for $n=3$ we defined $M_{0}$ and $M_{1}$ by

$$
M_{0}=\left(\begin{array}{cccc}
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1
\end{array}\right) M_{1}=\left(\begin{array}{cccc}
-1 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 \\
0 & 0 & -1 & 1
\end{array}\right)
$$

In Chapter 5 we defined $E_{0}, E_{1}, E_{2}, E_{1}^{x}, E_{2}^{y}, E$, and $E_{w}$ by

$$
\begin{aligned}
& \left\{E_{0}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{1}{4}\left(u_{i, j}+u_{i+1, j}+u_{i, j+1}+u_{i+1, j+1}\right) \\
& \left\{E_{1}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{n}{2}\left(-u_{i, j}+u_{i+1, j}-u_{i, j+1}+u_{i+1, j+1}\right) \\
& \left\{E_{2}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{n}{2}\left(-u_{i, j}-u_{i+1, j}+u_{i, j+1}+u_{i+1, j+1}\right)
\end{aligned}
$$

$$
\begin{gather*}
\left\{E_{1}^{x}\left(\begin{array}{c}
u_{1,1} \\
u_{2,1} \\
\vdots \\
\\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j}=\frac{n}{4}\left(x_{i}+x_{i+1}\right)\left(-u_{i, j}+u_{i+1, j}-u_{i, j+1}+u_{i+1, j+1}\right) \\
\left\{\begin{array}{c}
u_{1,1} \\
\left.E_{2}^{y}\left(\begin{array}{c}
u_{2,1} \\
\vdots \\
u_{n+1, n+1}
\end{array}\right)\right\}_{i, j} \\
=\frac{n}{4}\left(y_{j}+y_{j+1}\right)\left(-u_{i, j}-u_{i+1, j}+u_{i, j+1}+u_{i+1, j+1}\right)
\end{array}\right. \\
E(u)=\left(\begin{array}{c}
E_{0}(u) \\
E_{1}(u) \\
E_{2}(u)
\end{array}\right) E_{w}(u)=\left(\begin{array}{c}
E_{0}(u) \\
E_{1}^{x}(u) \\
E_{2}^{y}(u)
\end{array}\right) \tag{6.2}
\end{gather*}
$$

Rewrite $E_{0}, E_{1}$, and $E_{2}$ in terms of $M_{0}$ and $M_{1}$ as follows:

$$
\begin{aligned}
& E_{0}=\frac{1}{4}\left(\begin{array}{cccc}
M_{0} & M_{0} & 0 & 0 \\
0 & M_{0} & M_{0} & 0 \\
0 & 0 & M_{0} & M_{0}
\end{array}\right) \\
& E_{\mathbf{1}}=\frac{N}{2}\left(\begin{array}{cccc}
M_{1} & M_{1} & 0 & 0 \\
0 & M_{1} & M_{1} & 0 \\
0 & 0 & M_{1} & M_{1}
\end{array}\right) \\
& E_{2}=\frac{N}{2}\left(\begin{array}{cccc}
-M_{0} & M_{0} & 0 & 0 \\
0 & -M_{0} & M_{0} & 0 \\
0 & 0 & -M_{0} & M_{0}
\end{array}\right)
\end{aligned}
$$

This yields our operator for the non-weighted case

$$
B=E^{t} E=\left(E_{0}\right)^{t} E_{0}+\left(E_{1}\right)^{t} E_{1}+\left(E_{2}\right)^{t} E_{2}
$$

which may be rewritten as a block matrix which is tridiagonal in terms of the elementary building blocks, $M_{0}$ and $M_{1}$.

$$
B=\left(\begin{array}{cccc}
C & D & 0 & 0 \\
D & 2 C & D & 0 \\
0 & D & 2 C & D \\
0 & 0 & C & D
\end{array}\right)
$$

where

$$
D=\frac{1}{16} M_{0}^{t} M_{0}+\frac{n^{2}}{4}\left(M_{1}^{t} M_{1}+M_{0}^{t} M_{0}\right)
$$

and

$$
C=\frac{1}{16} M_{0}^{t} M_{0}+\frac{n^{2}}{4}\left(M_{1}^{t} M_{1}-M_{0}^{t} M_{0}\right)
$$

Unfortunately we now must pay the price for the choice of our ordering of the grid. Because we chose to order from left to right first and then from bottom to top, our following definition for $E_{1}^{x}$ is esthetically pleasing while our definition for $E_{2}^{y}$ is not; however, reordering the choice for the grid will still yield only one esthetically pleasing definition. Define

$$
\begin{aligned}
& M_{1}^{x}=\left(\begin{array}{cccc}
-\left(x_{1}+x_{2}\right) & \left(x_{1}+x_{2}\right) & 0 & 0 \\
0 & -\left(x_{2}+x_{3}\right) & \left(x_{2}+x_{3}\right) & 0 \\
0 & 0 & -\left(w_{3}+w_{4}\right) & \left(w_{3}+w_{4}\right)
\end{array}\right) \\
& E_{1}^{x}
\end{aligned}=\frac{N}{4}\left(\begin{array}{cccc}
-M_{1}^{x} & M_{1}^{x} & 0 & 0 \\
0 & -M_{1}^{x} & M_{1}^{x} & 0 \\
0 & 0 & -M_{1}^{x} & M_{1}^{x}
\end{array}\right) .
$$

$$
E_{2}^{y}=\frac{N}{4}\left(\begin{array}{cccc}
-\left(y_{1}+y_{2}\right) M_{0} & \left(y_{1}+y_{2}\right) M_{0} & 0 & 0 \\
0 & -\left(y_{2}+y_{3}\right) M_{0} & \left(y_{2}+y_{3}\right) M_{0} & 0 \\
0 & 0 & -\left(y_{3}+y_{4}\right) M_{0} & \left(y_{3}+y_{4}\right) M_{0}
\end{array}\right)
$$

Finally, we have $B_{w}: \Re^{(n+1)^{2}} \rightarrow \Re^{(n+1)^{2}}$,

$$
B_{w}=\left(E_{w}\right)^{t} E_{w}=\left(E_{0}\right)^{t} E_{0}+\left(E_{1}^{x}\right)^{t} E_{1}^{x}+\left(E_{2}^{y}\right)^{t} E_{2}^{y}
$$

Once again the simplest way to see the properties of $B$ is to consider a simple Mathematica code. Considering this code with $n=4$ will give the reader a good visual picture of the structure of $B$. We remind the reader that $B$ is band diagonal with nine non-zero bands.

## MATHEMATICA CODE

$$
\begin{aligned}
& n d=2 \\
& \mathrm{~h}=1 / 4 \\
& \mathrm{D} 0=\mathrm{h}^{*}\{\{1,1,0,1,1,0,0,0,0\},\{0,1,1,0,1,1,0,0,0\}, \\
& \{0,0,0,1,1,0,1,1,0\},\{0,0,0,0,1,1,0,1,1\}\} \\
& \mathrm{D} 1=\mathrm{nd} / 2^{*}\left\{\mathrm{rl}^{*}\{-1,1,0,-1,1,0,0,0,0\}, \mathrm{r}^{*}\{0,-1,1,0,-1,1,0,0,0\},\right. \\
& \left.r l^{*}\{0,0,0,-1,1,0,-1,1,0\}, r 2^{*}\{0,0,0,0,-1,1,0,-1,1\}\right\} \\
& \mathrm{D} 2=\mathrm{nd} / 2^{*}\left\{\mathrm{~s} 1^{*}\{-1,-1,0,1,1,0,0,0,0\}, \mathrm{s} 1^{*}\{0,-1,-1,0,1,1,0,0,0\},\right. \\
& \left.s 2^{*}\{0,0,0,-1,-1,0,1,1,0\}, s 2^{*}\{0,0,0,0,-1,-1,0,1,1\}\right\} \\
& \mathrm{A}=\text { Transpose[D0].D0 }+ \text { Transpose[D1].D1 }+ \text { Transpose[D2].D2 } \\
& \mathrm{uv}=\{\mathrm{u}[1][1], \mathrm{u}[2][1], \mathrm{u}[3][1], \mathrm{u}[1][2], \mathrm{u}[2][2], \mathrm{u}[3][2], \mathrm{u}[1][3], \mathrm{u}[2][3], \mathrm{u}[3][3]\} \\
& \mathrm{bv}=\{\mathrm{b}[1][1], \mathrm{b}[2][1], \mathrm{b}[3][1], \mathrm{b}[1][2], \mathrm{b}[2][2], \mathrm{b}[3][2], \mathrm{b}[1][3], \mathrm{b}[2][3], \mathrm{b}[3][3]\} \\
& \mathrm{zv}=\operatorname{Array}[z, 9] \\
& \operatorname{Do}[z[i]=\operatorname{Solve}[(\text { A.uv })[[i]]==\operatorname{bv}[[i]], u v[[i \mathrm{i}]],\{i, 9\}] \\
& \text { Dol Print[ } \\
& z[i][1]][[1]] \\
& \text { ] , }\{i, 9\}] \\
& \text { Do[CForm[ } \\
& \text { Together }[z[i][[1]][1]][[2]]] \\
& ]>\text { Bfile.mth },\{i, 9\}]
\end{aligned}
$$

## CHAPTER 7

## CONCLUSIONS

We have seen throughout the paper that for singular problems our method outperforms the other two methods considered. We emphasize that the weighted descent is an extension of the standard descent so that once the effort has been put forth to implement the non-weighted descent process, little extra effort is required to implement the weighted descent and superior results can be expected.

Perhaps the greatest advantage of the algorithm described is its adaptability. The reader has seen the versatility of the algorithm by viewing its application to several types of equations. We have considered constrained, unconstrained, and partially constrained problems of first order, two variational problems, and one partial differential equation and have touched on the approach to general second order equations. We note that the partial differential equation was approached in a way which is type independent. All of these problems required only a slight adaptation of the underlying theory and spaces from the first order examples. Boundary conditions are maintained at each step of the descent process, guaranteeing exact boundary conditions for the solution.

Given such versatility, one would expect a trade-off in numerical results for some of the problems; however, for every problem considered we have seen that weighted Sobolev descent outperform standard Sobolev descent which in turn outperforms
descent based on the Euclidean gradient. Furthermore, improvements such as multigrid methods and conjugate gradient methods are all applicable to the weighted gradients.

Another characteristic of the method is that the algorithm tends to give good results on a small number of divisions. For example, if one can obtain a given precision on a problem where the interval is subdivided into a large number of divisions then one is able to obtain the same precision if the interval was divided into a small number of divisions. Therefore, quick experiments may be performed using a small number of divisions which will be representative of the results obtainable for a large number of divisions. This also serves as an indication that computing on a course mesh and then using the results as inputs to the code on a finer mesh should lead to vastly improved runtimes.

We believe solving differential equations in this way will continue to lead us to questions such as the following, posed in Chapter 4. Which of the equations $\left(\nabla_{L} J\right)(u)=0$ (Euler's equation), $\left(\nabla_{H} J\right)(u)=0$, or $\left(\nabla_{H_{w}} J\right)(u)=0$ is the appropriate equation to consider for understanding variational approaches concerning singular differential equations?

Finally, we have offered a systematic approach for using weights to improve Sobolev steepest descent.

## BIBLIOGRAPHY

[A] Adams, R. A., Sobolev Spaces, New York : Academic Press (1975).
[CK] Canic, S. and Keyfitz, B. L., An Elliptic Problem Arising From the Unsteady Transonic Small Disturbance Equation, Journal of Differential Equations, to appear.
[C] Conway, John B., A Course in Functional Analysis, Springer-Verlag Graduate Texts in Mathematics, 96 (1985).
[CH] Courant R. and Hilbert D., Methods of Mathematical Physics, Volume I, John Wiley \& Sons (1989).
[DM] Dix, Julio G. and McCabe, Terence W., On finding equilibria for isotropic hyperelastic materials, Nonlinear Analysis 15 (1990) 437-444.
[E] Elschner, Johannes, Singular Ordinary Differential Operators and Pseudodifferential Equations, Springer-Verlag Lecture Notes In Mathematics, 1128 (1980).
[G] Garza, Javier, Using Steepest Descent to Find Energy-Minimizing Maps Satisfying Nonlinear Constraints, Dissertation, University of North Texas (1994).
[K] Kim, Keehwan, Steepest descent for partial differential equations of mixed type, Dissertation, University of North Texas (1992).
[KA] Kufner, Alois, Weighted Sobolev Spaces, John Wiley \& Sons (1985).
[N1] Neuberger, J. W., Constructive variational methods for differential equations, Nonlinear Analysis, Theory, Methods \& Applications, 13 (1988) 413-428.
[N2] Neuberger, J. W., Calculation of Sharp Shocks Using Sobolev Gradients, Contemporary Mathematics 108 (1990) 111-118.
[N3] Neuberger, J. W., Iteration for systems of nonlinear partial differential equations, Nonlinear Equations in Abstract Spaces, Academic Press, Inc. (1978) 253263.
[N4] Neuberger, J. W., "Sobolev Gradients and Differential Equations," in preparation.
[N5] Neuberger, J. W., Steepest descent and differential equations, J. Math. Soc. Japan 37 (1985) 187-195.
[N6] Neuberger, J. W., Steepest descent for general systems of linear differential equations in Hilbert space, Ordinary Differential Equations and Operators, SpringerVerlag Lecture Notes In Mathematics 1032 (1982) 390-406.
[RN] Renka, R. J. and Neuberger, J. W., Minimal Surfaces and Sobolev Gradients, to appear, SIAM Journal of Scientific Computing.
[RSN] Riesz, F. and Sz.-Nagy B., Functional Analysis, Dover Publications (1990).
[S] Schuchman, Vladimir, On Behavior of Nonlinear Differential Equations in Hilbert Space, International Journal of Mathematical Sciences, 11 (1988) 143-165.

