MSc Dissertation

Analysis of Integral Operators from Scattering Problems

Edmund Ridley

August 20, 2006

Acknowledgements

I would like to acknowledge Dr. Marko Lindner and Dr. Stephen Lang-

Contents

List of Figures

1 Introduction

manipulation of the operator A using theory from projection methods in [9] we state why the norms of $||A||$ and $||A^{-1}||$ are crucial in finding out about the behaviour of the error in the solution (i.e. $||\varphi - \varphi_n||$).

This project looks at the single layer potential (2.8), which is a key part of the operator A, and is an important first step into looking at the operator A as a whole. The choice of investigating A instead of A^{-1} is that much work has been done on the inverse operator such as in the paper [1], and less is known about A. In looking at the operator S, we analyse $||S||$ in L^p space (i.e. $||S||_{L^p}$). We establish upper (Section 4) and lower bounds (Section 5) on $||S||_{L^p}$. We start by looking at two special cases where $p = 1$ and $p = \infty$ as these provide us with upper bounds on $||S||_{L^p}$ for $1 < p < \infty$. These two cases fortunately have explicit formulae for $||S||_{L^p}$, unlike for $p = 2$. The explicit formulae are also single intergrals, whereas for the 2-norm we are required to compute a double integral. Although, we also discuss how it is possible to use the more complicated double integrals for the 1-norm and ∞-norm to approximate $||S||_{L^p}$. After this we discuss two upper bounds, one of which is found by manipulation of $||S||_{L^p}$ and another which is found by use of the *Riesz-Thorin interpolation* formula. With these upper bounds, and other arguments that give us $||S||_{L^p}$ being symmetric about $p = 2$ and also $p = 2$ being a global minimum, we are able to combine these upper bounds to provide us with the figures towards the end of Section 4..

The next stage is to look at the lower bounds of $||S||_{L^p}$, which is done in Section 5. In particular in this section we look at $||S||_{L^2}$, for the reason that it is then possible to make direct comparisons with results from [3] and [1]. These results being on various shapes, such as a circles and convex objects with a smooth boundary (both of which are looked at here) in [3] and polyhedra in [1]. In looking at the lower bound for $||S||_{L^2}$, we numerically

compute $\frac{||Su||_{L^2}}{||u||_{L^2}}$, where we seek to find a u such as our numerical approximation gives a value close to the real value of $||S||_{L^2}$. This lower bound can then be combined with the upper bound to provide us with the figures in Section 8 and Section 9.

In Section 6 we then talk about the numerical methods used to compute these upper and lower bounds. Gaussian quadrature is introduced here, and an example is used to give an idea into how it works. Also, we include some equations which test the method (and the code) to indicate how Gaussian quadrature converges with n (the number of Gaussian quadrature points) going to infinity. With Section 7 continuing this numerics by discussing how it is possible to speed up the code in MATLAB with the use of vectors and matrices. Then we go on to discuss one way to use the quadrature points more effectively by use of a graded mesh, with at the same time taking consideration of the singularity.

Next we have Section 8 on the results that have been obtained. With a table showing how our numerically approximated $||S||_{L^2}$ changes as $k \to \infty$. Also here we include figures of the lower and upper bounds as k increases. Then we go on to our conclusions and relate then to literature, and finally future work which would be the next stages if more time was available.

2 Background

There is much theory on wave scattering, and hence here is a condensed and brief outline of the general problem for the *direct scattering problem*. That is, the problem of finding the *scattered field* after a wave has hit an object, when the *incident wave* is known.

with $J_R(x)$ being a Bessel function of the first kind and $Y_R(x)$ being a Bessel function of the second kind (also known as the Neumann function). These Bessel functions originate from being solutions of a special case of the

becomes

$$\int_{\Gamma} \frac{\partial \varphi(x,y)}{\partial n(x)} \frac{\partial u(y)}{\partial n} ds(y) - \frac{\partial u(x)}{\partial n}, \quad x \in \Gamma$$

(see [8] and [7]). By multiplying (2.4) by $i\eta$ and adding (2.5), then multiplying all through by 2 we get,

$$\frac{\partial u(x)}{\partial n} + 2 \int_{\Gamma} \{ \frac{\partial \varphi(x,y)}{\partial n} + i\eta \varphi(x,y) \} \frac{\partial u(y)}{\partial n} ds(y) = f(x), \quad x \in \Gamma, \qquad (2.6)$$

where $f(x) = 2 \left[\frac{\partial u^i}{\partial n} + i\eta u^i(x) \right]$. Thus we have our unknown

$$\frac{\partial u}{\partial n}$$

and the following integral equation,

$$(I - B) \frac{\partial u}{\partial n} = f. \qquad (2.7)$$

Then denoting

$$Bv = \int_{\Gamma} R_k \cdot v$$

where

$$R_k = 2 \left[\frac{\partial \varphi(x,y)}{\partial n(x)} + \right.$$

to solve, where $A = I + 2(D + i\eta S) = I - B$, and we seek the solution φ. To solve (2.1) numerically one can use a projection method, such as the Galerkin and collocation methods. Writing the problem as $A\varphi = (I - B)\varphi = f$ then we can look at how projection methods can be used to look at the error analysis of this problem. From Section 3 in [9] there is much description on precisely how the projection methods can be used to look at the error analysis. We will take the key outline of the argument relevant for the problem that has been looked at in this project.

We have a projection operator P_n which when applied to (2.10) we have,

$$P_n A\varphi_n = P_n f$$

and hence

$$P_n(I - B)\varphi_n = P_n f \text{ (equation (3.1.24) in [9]).}$$

Noting that $P_n\varphi_n = \varphi_n$, we can then write,

$$(I - P_n B)\varphi_n = P_n f \text{ (equation (3.1.25) in [9]),}$$

where the $A_n = (I - P_n B)$. A_n is determined by the method. Therefore we now have,

$$A_n\varphi_n = P_n f, \text{ i.e. } \varphi_n = A_n^{-1}P_n f.$$

We are interested in φ_n since this is our approximation to the solution φ, and thus knowledge of A_n^{-1} is also of interest. Using equation (3.1.27) from [9] we can write that,

$$A_n = A[I + A^{-1}(B - P_n B)].$$

Then using Theorem 3.1.1 in [9] and assuming that $||B - P_n B|| \to 0$ as $n \to \infty$ we can say that A_n^{-1} exists and is also bounded (i.e. $||A_n^{-1}|| < \infty$ for sufficiently large n). Information about the operators A and A^{-1} can tell

us about A_n and A_n^{-1}, and hence tell us something about how φ_n depends on k. For our particular problem we can use the Theorem 3.1.1, and the equations (3.1.30) and (3.1.31) to say that,

$$\frac{1}{||A_n||}||\varphi - P_n\varphi|| \leq ||\varphi - \varphi_n|| \leq ||A_n^{-1}||||\varphi - P_n\varphi||.$$

This gives us upper and lower bounds for the error term norm, $||\varphi - \varphi_n||$. In this project we are concentrating on a part of the operator A. We are looking at the operator S, our single layer potential. Using this is the first step in investigating the norm of A, $||A||$. Much work has been done on the inverse in such papers as [3] and [1]. Information on $||A||$ is not so vast, with this project giving a first stage on how to tackle $||A||$. Also, of great interest is to know the condition number of A,

$$\text{cond } A := ||A|| \cdot ||A^{-1}||.$$

The condition number indicates how much a change in f effects the approximate solution φ_n. A large condition number represents a large change, and hence a small amount of error in f creates a big error in the solution. Clearly, knowledge of $||A||$ helps to understand the condition number.

much of the results gathered have been on $||A^{-1}||$. These results include $||A^{-1}|| \leq 2$ (for su□ciently large k) for a circle shown in [3] and an upper bound for $||A^{-1}||$ for polyhedra in [1].

3 A Brief Outline of Norms

A set X with an operation + is a *linear space* if $x + y \quad X$, $\alpha x \quad X \quad \alpha$ C, $x, y \quad X$ and the following axioms hold:

- $x + y = y + x$
- $(x + y) + z = x + (y + z)$
- $0 \quad X$ such that $0 + x = x \quad x \quad X$
- $x \quad X$ " $-x$" $\quad X$ such that $x + (-x) = 0$
- $(\alpha + \beta)x = \alpha x + \beta x$
- $\alpha(x + y) = \alpha x + \alpha y$
- $1 \cdot x = x, \quad x \quad X$

If X, Y are linear spaces and $A : X \rightarrow Y$ is a map with,

- $A(x + y) = Ax + Ay, \quad x, y \quad X$
- $A(\alpha x) = \alpha \cdot Ax, : \quad \alpha \quad \mathrm{C}, x \quad X$

then we call A a *linear operator* from X to Y.

A linear space X is called a *normed linear space* if there is a map $||.|| : X \longrightarrow \mathrm{R}_+$ such that

- $||x|| \geq 0, \quad x \quad X$ with $||x|| = 0 \qquad x = 0$
- $||\alpha x|| = |\alpha| \cdot ||x||, \quad \alpha \quad \mathrm{C}, x \quad X$

- $||x + y|| \leq ||x|| + ||y||, \quad x, y \quad X$.

Examples of some linear spaces with norms are

1. $X = \mathrm{R}^n$ with $||(x_1, ..., x_n)|| = \sqrt[p]{|x_1|^p + ... + |x_n|^p}$ where $p \quad [1, \infty)$.
   - For $p = 2$, we have the Euclid Norm $||(x_1, x_2)|| = \sqrt{|x_1|^2 + |x_2|^2}$
2. $X = \mathrm{R}^n$ with

$$\begin{aligned} ||(x_1, ..., x_n)|| &= \max\{|x_1|, ..., |x_n|\} \\ &= \lim_{p \to \infty} \sqrt[p]{|x_1|^p + ... + |x_n|^p} \end{aligned}$$

3. $X = L^p[0, 1]$ with $||f|| = \sqrt[p]{\int_0^1 |f(t)|^p dt}$
   - Note: $L^p[0, 1]$ is the space such that $\int_0^1 |f(x)|^p dt$ exists and is finite.

4.1 The Single Layer Potential

This project will be concentrating on the single layer potential operator,

$$(Su)(x) = \int_\Gamma H_0^1(k|x$$

If $||u|| = 1$ we have

$$||S|| = \sup_{||u||=1} ||Su||$$

hence

$$||S||_{L^p}$$

Then parameterising this gives,

$$\begin{aligned} ||Su||_{L^1(\Gamma)} &= \int_0^{2\pi} \left(\int_0^{2\pi} |\kappa(t,\tau)||y'(t)|dt \right) |u(\tau)||y'(\tau)|d\tau \\ &\leq \sup_{\tau \in [0,2\pi]} \int_0^{2\pi} |\kappa(t,\tau)||y'(t)|dt \int_0^{2\pi} |u(\tau)||y'(\tau)|d\tau \\ &= M_1 ||u||_{L^1[0,2\pi]}, \end{aligned}$$

where

$$M_1 := \sup_{\tau \in [0,2\pi]} \int_0^{2\pi} |\kappa(t,\tau)||y'(t)|dt.$$

We can choose appropriate functions (as shown in section(4.4)) so that this M_1 is the smallest $M_1 \geq 0$ and hence we can write,

$$||S||_{L^1(\Gamma)} = \sup_{\tau \in [0,2\pi]} \int_0^{2\pi} |\kappa(t,\tau)||y'(t)|dt. \tag{4.6}$$

The advantage of (4.6) compared to (4.4) is not only that it is a single integral and thus less costly to numerically compute, but more importantly there is no u term. This means it is not needed to look for a u, as is required when computing (4.4).

(Results computed using (4.6) can be seen in Table 8.1 and Table 8.2).

4.3 The $p = \infty$-norm Single Integral

As with the one norm we can perform some manipulations on (4.2) to obtain

$$||Su||_{L^\infty[0,2\pi]} \leq M_\infty ||u||_{L^\infty[0,2\pi]}.$$

We have

$$\begin{aligned} ||Su||_{L^\infty(\Gamma)} &= \sup_{x \in \Gamma} |(Su)(x)| \\ &= \sup_{x \in \Gamma} \left| \int_\Gamma \kappa(x,y)u(y)ds(y) \right| \\ &\leq \sup_{x \in \Gamma} \int_\Gamma |\kappa(x,y)||u(y)|ds(y) \end{aligned}$$

Taking the supremum of $|u(y)|$, it can then be taken out of the integral such that,

$$||Su||_{L^\infty(\Gamma)} \leq \sup_{x \in \Gamma} \int_\Gamma |\kappa(x,y)| ds(y) \sup_{y \in \Gamma} |u(y)|$$

Then parameterising we have,

$$||Su||_{L^\infty[0,2\pi]} = \sup_{t \in [0,2\pi]} \int_0^{2\pi} |\kappa(t,\tau)||\gamma'(\tau)| d\tau \sup_{\tau \in [0,2\pi]} |u(\tau)|$$

and hence as required,

$$||Su||_{L^\infty(\Gamma)} = M_\infty ||u||_{L^\infty(\Gamma)}$$

where

$$M_\infty := \sup_{t \in [0,2\pi]} \int_0^{2\pi} |\kappa(t,\tau)||\gamma'(\tau)| d\tau.$$

This can be shown to be the smallest $M_\infty \geq 0$ (as shown in Section(4.4)) and therefore we have,

$$||S||_{L^\infty(\Gamma)} = \sup_{t \in [0,2\pi]} \int_0^{2\pi} |\kappa(t,\tau)||\gamma'(\tau)| d\tau. \tag{4.7}$$

As with the single integral for the one norm, this gives us the same advantages. In fact, because of the symmetry of the Hankel function we have that,

$$\begin{aligned} ||S||_{L^1(\Gamma)} &= \sup_{y \in \Gamma} \int_\Gamma \left| \frac{i}{4} H_0^1(k|x-y|) \right| ds(x) \\ &= \sup_{y \in \Gamma} \int_\Gamma \left| \frac{i}{4} H_0^1(k|y-x|) \right| ds(x) \\ &= \sup_{x \in \Gamma} \int_\Gamma \left| \frac{i}{4} H_0^1(k|x-y|) \right| ds(y) \\ &= ||S||_{L^\infty(\Gamma)}, \end{aligned}$$

so $M_1 = M_\infty$. More generally we have the result

$$||S||_{L^p} = ||S||_{L^q}, \text{ if } \frac{1}{p} + \frac{1}{q} = 1 \tag{4.8}$$

for all $p \in [1, \infty]$. This means that $||S||_{L^p}$ is symmetric about $p = 2$ (in the sense that $p = 2$ is the mid-point of the interval $[1, \infty]$), which will be used later on.

4.4 Double Integral for the One and Infinity Norms

Although we have the single integrals for the 1-norm and the ∞-norm, (4.6) and (4.7) respectively, it is also possible to compute them accurately using the double integral (4.4). Simplifying this as a matrix problem we have,

$$Su = \begin{bmatrix} & \leftarrow \ \tau \ \rightarrow & \\ \uparrow & & \\ t & K & \\ \downarrow & & \end{bmatrix} \begin{bmatrix} & & \\ & & \\ & u & \\ & & \\ & & \end{bmatrix},$$

where $K = \kappa(t,\tau)|y(\tau)|$. The 1-norm corresponds to the maximum column sum and the ∞-norm corresponds to the maximum row sum of K.

Equation (4.4) for $p = 1$ is

$$||S||_{L^1(\Gamma)} = \sup_{||u||_{L^1}=1} \int_0^{2\pi} \left[\int_0^{2\pi} \kappa(t,\tau)u(\tau)|y(\tau)|d\tau \right] |y(t)|dt \qquad (4.9)$$

With the 1-norm being equivalent to a column sum, we need to find the value of τ say τ where this is greatest. Thus, by choosing u such that it 'picks' out this column it would give an approximation for the supremum of the 1-norm. Looking at Figure 4.1 we have a piecewise constant over the interval $[a_j, b$

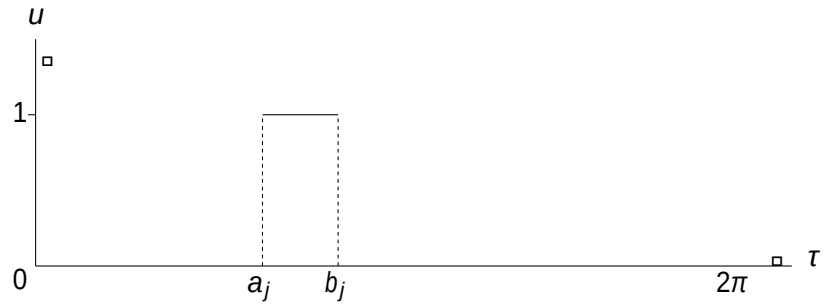

Figure 4.1: u Chosen for the One Norm

Thus, to keep $||u||_{L^1} = 1$ we are required to divide by $\int_{a_j}^{b_j} |y(\tau)|d\tau$. Trying N different u with just one piecewise constant where $u = 1$ over the interval $[a_j, b_j]$ for $j = 1, ..., N$, and $u = 0$ over the rest of the interval (i.e. $[0, a_j]$ and $[b_j, 0]$). For each u,

$$\int_0^{2\pi} \left| \int_0^{2\pi} \kappa(t,\tau)u(\tau)|y(\tau)|d\tau \right| |y(t)|dt.$$

needs to be divided by

$$\int_{a_j}^{b_j} |y(\tau)|d\tau.$$

We then take the largest value and this is the approximation for $||S||_{L^1}$. Then as N is increased the approximation improves.

Table 4.1 shows that as N (which concentrates u on a smaller and smaller part of $K(t,\tau)$) increases the values given by the double integral become closer to the single integral value (in Table 8.2).

With the ∞-norm being the maximum row sum, it is required to find a u such that

$$||u||_{L^\infty(\Gamma)} = \sup_{\tau \ [0,2\pi]} |u(\tau)| = 1 \tag{4.11}$$

and which 'picks' out the maximum row sum. The maximum row sum can be found in this case by using the single integral. This is used by choosing

No. of Piecewise Constants	$k = 1$	$k = 2$
1	0.8324275994	0.23012335522458
2	1.0414252323	0.49073840309276
4	1.4298599218	0.71693171534810
8	1.5807037799	1.09042637401177
16	1.6554936274	1.15937045669100
32	1.6839774230	1.21630454860917
64	1.6925392013	1.23577686095915
128	1.6950365160	1.24129630635884
256	1.6957527179	1.24283441218079

Table 4.1: Values of (4.4) for $y(\tau) = (2\cos\tau + \sin\tau)$

many t, and remembering the value of t say t where the single integral is largest. This value of t indicates the row which has the maximum sum. Choosing a u which then multiplies each entry in such a way that this is maximised would give the required u to find $||u||_{L^\infty(\Gamma)}$. Hence, since it is only necessary for the supremum of the $|u|$ to be one it is possible to let all $|u| = 1$. Splitting u into piecewise constants, and choosing the piecewise constants such that they return the maximum real positive value. This means each $K(t\ ,\tau)$ needs to be multiplied by

$$U_j = \frac{|K(t\ ,\tau)|}{K(t\ ,\tau)},$$

where the U_j is the j*th* piecewise constant, where $j = 1, \ldots, N$ as with the 1-norm. Therefore, in this case the real and imaginary part of u would look

Number of Piecewise Constants	$k = 1$	$k = 2$
1	1.02268644630414	0.36106346459050
2	1.31546183059806	0.58603913271996
4	1.49238353781992	0.81049822013627
8	1.62799322382511	1.09309495375556
16	1.66182296004815	1.17544163253251
32	1.68166215056378	1.22204280535008
64	1.68608506923276	1.23270470271323
128	1.68792226061701	1.23673616359326
256	1.68834810758787	1.23767714177830
512	1.68852172630963	1.23802042310760

Table 4.2: Values of (4.12) for $y(\tau) = (2\cos\tau, \sin\tau)$

4.5 Other Upper Bounds On $||S||_{L^p}$ for $p \in (1, \infty)$

The $p = 2$ norm has the most interest and it would be good to see how the $||S||_{L^2}$ changes as *gF,fdpJFfdJFcfhanges as*

and the *Hölder* Inequality for Integrals (taken from [4])

$$\left| \int_a^b f(x)g(x)dx \right| \leq \left(\int_0^b |f(x)|^c dx \right)^{\frac{1}{c}} \left(\int_a^b |g(x)|^d dx \right)^{\frac{1}{d}}$$

if

$$\frac{1}{c} + \frac{1}{d} = 1, c > 1, d > 1.$$

we get

$$||Su||_{L^p(\Gamma)} \leq \left(\int_\Gamma \left(\int_\Gamma |\kappa(x,y)|^q ds(y) \right)^{\frac{1}{q}} \left(\int_\Gamma |u(y)|^p ds(y) \right)^{\frac{1}{p}p} ds(x) \right)^{\frac{1}{p}}$$

with

$$||u||_{L^p} = \left(\int_\Gamma |u(y)|^p \right)^{\frac{1}{p}}.$$

Then

$$||Su||_{L^p(\Gamma)} = \left(\int_\Gamma \left(\int_\Gamma |\kappa(x,y)|^q ds(y) \right)^{\frac{1}{q}p} ||u||_{L^p} \right) dfdjjJFfdLJFfdpJFfadJFJFfdLJFfF$$

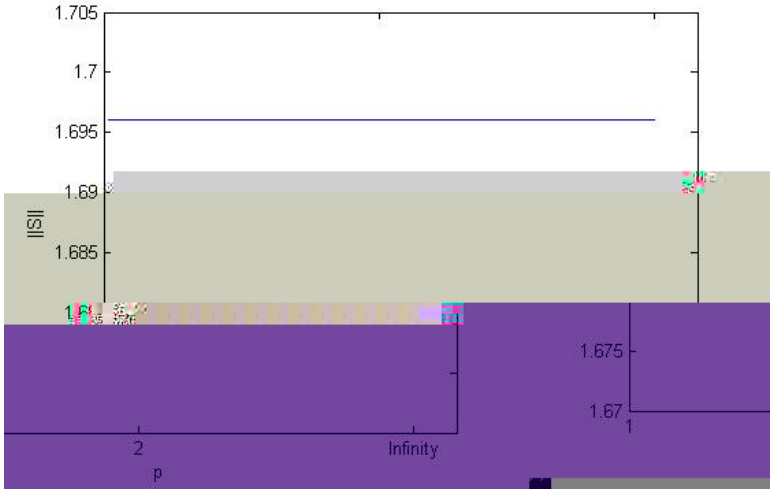

Figure 4.4: Upper bound Using the Riesz-Thorin Interpolation 4.14 for $k = 1$

5 Lower Bounds on $||S||_{L^p}$

The lower bound for the 2-norm can be found by looking at equation (4.4). Letting $p = 2$ we have the following,

$$||S||_{L^2[0,2\pi]} = \sup_{||u||_{L^2}=1}$$

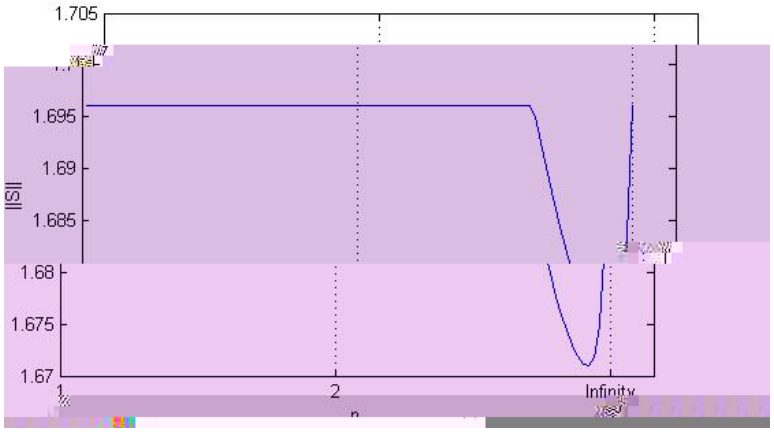

Figure 4.5: Combined Upper Bounds from Figures 4.4 and 4.3 for $k = 1$

where $a_j = \frac{(j-1)2\pi}{N}$, $b_j = \frac{j2\pi}{N}$ and U_j is a constant. The constants are chosen randomly over a uniform distribution, and are of the form,

$$U_j = A_j + B_j i, \text{ with } A, B \quad [-1, 1], \ i = \sqrt{-1}.$$

Then we also have,

$$||u||_{L^2} = \left[\sum_{j=1}^{N} \int_{a_j}^{b_j} |U_j|^2 |y(\tau)| \right]^{\frac{1}{2}}$$

which gives us an lower bound for $||S||_{L^2[0,2\pi]}$. Using this technique Table 5.1 shows the lower bound values that were found.

6 Numerical Methods for the Lower and Upper Bounds

All integrals we approximated using *Legendre Gaussian Quadrature*. Although, for highly oscillatory functions such as the Hankel function this does not give exactness, all results shown have converged as in 6.3. We begin this section by giving an overview of Gaussian Quadrature.

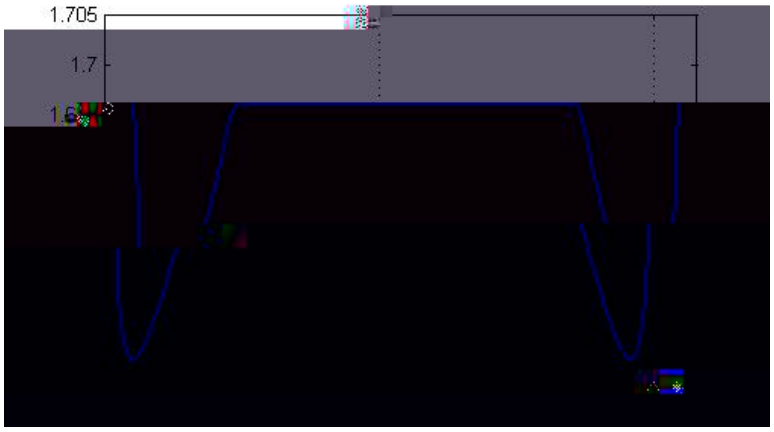

Figure 4.6: Upper Bound using the Symmetric Property for Figure 4.5

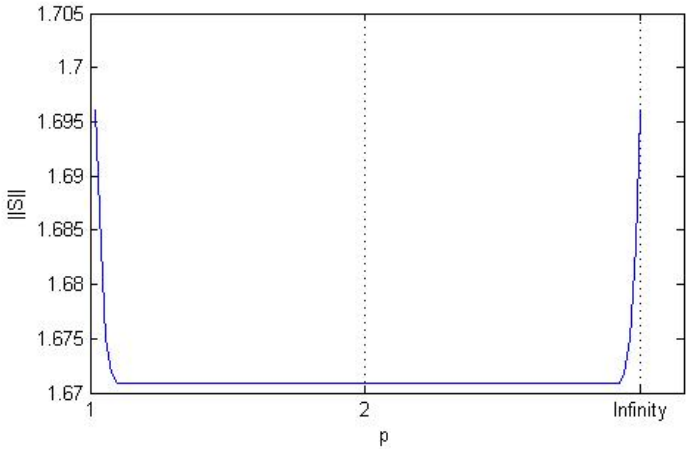

Figure 4.7: Symmetric Upper Bound for k = 1 Using 4.14 Again

k	$\|\|S\|\|_{L^2}$ Lower Bound for $y(\tau) = (\cos\tau, \sin\tau)$	$\|\|S\|\|_{L^2}$ Lower Bound for $y(\tau) = (2\cos\tau, \sin\tau)$
1	0.92591123989195	0.91814408816802
2	0.48317060607222	0.54140030886589
4	0.24311615739346	0.26110271592417
8	0.11254776960321	0.13052717540095
16	0.05744056422210	0.07149107040318
32	0.02268706077302	0.03127729495608

Table 5.1: Lower Bound Numeric Approximations for the 2-norm

6.1 Gaussian Quadrature

Gaussian Quadrature is a type of numerical integration. The likes of the *Simpson Rule* and the *Trapezoidal Rule* converge much slower for almost all functions. With the Simpson Rule being accurate for quadratic curves between nodes, and the Trapezoidal Rule being accurate for linear curves between nodes. Gaussian Quadrature is based around taking the roots of a specific polynomial and using this to decide which points to use. This is the di□erence 297(and)-297.375(297(and)-Aiet91(auss)3ono)-27(ds)3o1(d)-404(th)1(e)-405(T)83(raps)3o

polynomials of degree $2n - 1$ with n being the number of points used.

6.1.1 Legendre Polynomial

The Legendre polynomials provide one option for the weights and points to be used for Gaussian Quadrature. The Legendre polynomials are an infinite set of orthogonal polynomials on the interval $(-1, 1)$ with respect to the weight function $w(x) \equiv 1$. These polynomials are the solution of Legendre's differential equation,

$$\frac{d}{dx}\left[(1 - x^2)\frac{d}{dx}P(x)\right] + n(n + 1)P(x) = 0 \tag{6.1}$$

They can be expressed using *Rodrigues' formula* [4]

$$P_n(x) = (2^n n!)^{-1}\frac{d^n}{dx^n}\left[(x^2 - 1)^n\right]. \tag{6.2}$$

Hence the first four are:

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= \frac{3}{2}x^2 - \frac{1}{2} \\ P_3(x) &= \frac{5}{2}x^3 - \frac{3}{2}x \end{aligned}$$

6.2 How Gaussian Quadrature Works in 1-D

Gaussian Quadrature approximates a function $f(x)$ on a defined interval $[a, b]$ say. This function is ideally continuous. However, if it is not but it is known where the function is not continuous the interval can be split up to avoid the discontinuity [1]. We require

[1] Of course notice should be made that the discontinuous area is an important part of the integration, and would be foolish to ignore off hand.

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \tag{6.3}$$

The weights w_i and points x_i are determined by the polynomial used. The domain usually taken for the rule is $[-1, 1]$, and therefore requires a transformation for a general $[a, b]$ domain where $a, b \in R$. This is done by

$$\int_a^b f(x)dx = \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right) dx$$

which after applying the Gaussian quadrature rule is

$$\frac{b-a}{2} \sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}x_i + \frac{a+b}{2}\right). \tag{6.4}$$

The accuracy of Gaussian Quadrature (provides the exact value of a polynomial of degree $(2n-1)$ with n

where $a_0, a_1, \ldots, a_5$ are constants (noticing there are the same number of constants as there are weights and points). Therefore it is needed that,

$$\int_{-1}^{1} (a_0 + a_1 x + \ldots + a_5 x^5)dx = a_0 \int_{-1}^{1} 1dx + a_1 \int_{-1}^{1} xdx + \ldots + a_5 \int_{-1}^{1} x^5 dx$$

which is equivalent to showing that the Gaussian quadrature method gives exact results when $f(x) = 1, x, \ldots, x^5$. Thus we need,

$$\begin{aligned}
w_1 + w_2 + w_3 &= \int_{-1}^{1} 1dx = 2 \\
w_1 x + w_2 x + w_3 x &= \int_{-1}^{1} xdx = 0 \\
w_1 x^2 + w_2 x^2 + w_3 x^2 &= \int_{-1}^{1} x^2 dx = \frac{2}{3} \\
w_1 x^3 + w_2 x^3 + w_3 x^3 &= \int_{-1}^{1} x^3 dx = 0 \\
w_1 x^4 + w_2 x^4 + w_3 x^2 &= \int_{-1}^{1} x^4 dx = \frac{2}{5} \\
w_1 x^5 + w_2 x^5 + w_3 x^5 &= \int_{-1}^{1} x^5 dx = 0.
\end{aligned}$$

This system of equations has the unique solution,

$$x_1 = -0.77459666924148, x_2 = 0, x_3 = 0.77459666924148$$

$$w_1 = 0.55555555555556, w_2 = 0.88888888888889, w_3 = 0.55555555555556,$$

where the x_i are the roots of $P_3(x) = 0$. This works for all

$$\int_{-1}^{1} f(x,y)dxdy = \sum_{j=1}^{m}\sum_{i=1}^{n} w_i v_j f(x_i, y_j)dxdy$$

where the w_i and the x_i are the weights and the points in the x direction and the v_j and the y_j are the weights and the points in the y direction . This double integral again needs to be transformed into the interval $[a, b] \times [c, d]$. This is done in a similar way to before.

$$\int_{a}^{b} f(x,y)dxdy = \sum_{j=1}^{m}\sum_{i=1}^{n} w_i v_j f\left(\frac{b-a}{2}x_i + \frac{b+a}{2}, \frac{d-c}{2}y_j + \frac{d+c}{2}\right) dxdy \tag{6.5}$$

6.3 Testing the Method

According to the theory if I take n Gaussian quadrature points over the interval $[-1, 1]$ then this should give the exact value for a polynomial of order $2n - 1$. Thus if I take $n = 2$ then this should give the exact value for any cubic. By taking many different n and testing the MATLAB code we have written on the relevant polynomial we can test if the code is correct and that the Gaussian Quadrature method is working. Looking at the values in Table 6.1 where n Gaussian Quadrature points and $(n - 1)$ Gaussian Quadrature points have been used, and the interval $[-2, 3]$ you can see that a polynomial of order $2n - 1$ is integrated correctly for n points but not for $(n - 1)$ points. Also notice how the accuracy improves for a function such as sin and cos as n increases. Testing the double integral code is also necessary. Choosing $a = 1, b = 2, c = 3$ and $d = 4$ and increasing n and m as we increase the order of the polynomials lets us see whether the code is as accurate as it should be. (See Table 6.2)

Polynomial	Exact Value	n	n Points	$(n-1)$ Points
$2x^3 + x^2 - 2$ (cubic)	34.1667	2	34.1667	−7.5000
$-x^5 + 3x^4$ (quintic)	54.1667	3	54.1667	45.4861
$7x^7 + 7x^6$ (7^{th} order)	7831.87	4	7831.87	6952.96
$\sin(x) + \cos(x)$	1.62426	5	1.62428	1.62298

Table 6.1: Single Integral Gaussian Quadrature Code

Polynomial	Exact Value	n, m	n, m Points	$(n-1), (m-1)$ Points
$x^3 + y^3$	47.5	2, 2	47.500	46.250
$x^5 + y^5$	571.667	3, 3	571.667	571.569
$x^7 - y^7$	−7.34000	4, 4	−7.34000	−7.33999
$sin(x) + cos(y)$	−0.268522	3, 3	−0.268522	−0.268442

Table 6.2: Double Integral Gaussian Quadrature Code

6.4 Evaluating the Single Integral

We have the two single integrals

$$||S||_{L^1(\Gamma)} = \sup_{\tau \ [0,2\pi]} \int_0^{2\pi} |\frac{i}{4} H_0^1(k|y(t) - y(\tau)|)|y\,(t)|dt \tag{6.6}$$

$$||S||_{L^\infty(\Gamma)} = \sup_{t \ [0,2\pi]} \int_0^{2\pi} |\frac{i}{4} H_0^1(k|y(t) - y(\tau)|)|y\,(\tau)|d\tau. \tag{6.7}$$

We have H_0^1 as the Hankel function, this is made up of the first kind and second kind Bessel functions. $y(\tau)$ represents the shape which we are looking at, for instance in the circle case we have $y(\tau) = (\cos(\tau), \sin(\tau))$. In this case the $|y\,(\tau)|$ term would be $\sqrt{\cos^2(\tau) + \sin^2(\tau)}$. Also, there is the $|y(t) - y(\tau)|$

term. This term is the vector modulus of $y(t) - y(\tau)$. Thus for the circle

$$|y(t)-y(\tau)| = \left\| \begin{matrix} \cos(t) - \cos(\tau) \\ \sin(t) - \sin(\tau) \end{matrix} \right\| = \sqrt{(cos(t) - cos(\tau))^2 + (sin(t) - sin(\tau))^2}.$$

Performing some manipulation,

$$\sqrt{(cos(t) - cos(\tau))^2 + (sin(t) - sin(\tau))^2} = \sqrt{2 - 2\cos(t - \tau)}$$

n	$k = 1$	$k = 2$	$k = 64$
8	1.25599845796364	0.92636623468931	0.17206436985466
16	1.26820470397750	0.93836905487429	0.17805048230319
32	1.27153224321317	0.94166907961211	0.18061491732263
64	1.27240328043588	0.94253590847969	0.18141859166216
128	1.27262646245432	0.94275842180936	0.18163342024274
256	1.27268300857123	0.94281485758333	0.18168871333813
512	1.27269725094977	0.94282908104077	0.18170275604897
1024	1.27270082697329	0.94283265370303	0.18170629876787
2048	1.27270172332930	0.94283354944311	0.18170718933635
4096	1.27270194779501	0.94283377379287	0.18170741275990
8192	1.27270200397511	0.94283382995065	0.18170746874657
16384	1.27270201803147	0.94283384400261	0.18170748276610
32768	1.27270202154766	0.94283384751793	0.18170748627512

Table 6.3: Approximations of $||S||_{L^1} = ||S||_{L^\infty}$ for a Circle with $t = 0$

7 Speeding Up the Code

There are two key ways that the code has been improved. This is by using vectors and matrices instead of *for* loops, and also by using a graded mesh.

7.1 *For* Loops vs Vectors and Matrices

In MATLAB *for* loops are computed significantly slower than multiplying vectors and matrices. This is because MATLAB was originally designed to

n	$t = \frac{pi}{4}$	$t = \frac{\pi}{2}$	$t = \frac{3\pi}{2}$	$t = 2\pi = 0$
1024	1.2722513420	1.2716660046	1.2722513420	1.2727008269
2048	1.2731297511	1.2721816171	1.2731297511	1.2727017233
4096	1.2725886266	1.2724409165	1.2725886266	1.2727019477
8192	1.2728096451	1.2725711201	1.2728096451	1.2727020039
16384	1.2726735661	1.2726364333	1.2726735661	1.2727020180
32768	1.2727290360	1.2726691723	1.2727290360	1.2727020215

Table 6.4: Approximations of $||S||_{L^1} = ||S||_{L^\infty}$ for Different t for a Circle

manipulate vectors and matrices, e.g. finding inverses, and therefore needed to do this efficiently. A *for* loop can be written equivalently by using a vector multiplied by another vector. This can be seen clearly by taking an example of the code that was used in computing Gaussian Quadrature integration for the single integral, i.e. to evaluate $\int_a^b f(t, \tau, k)d\tau \approx \sum_{j=1}^{n} w_i f(t_j, \tau_j, k)$.

```
for 1:n;
    h = h + w(i)*f(t(i),tau,k)
end
```

This can be re-written using two vectors,

```
w*f(t,tau,k)
```

What is going on above can be expressed as,

$$\begin{bmatrix} w(1) & w(2) & \dots & w(n) \end{bmatrix} \begin{bmatrix} f(t(1), tau, k) \\ f(t(2), tau, k) \\ \dots \\ f(t(n), tau, k) \end{bmatrix}.$$

Similarly, two *for* loops can be written by *vector* × *matrix* × *vector*. Taking another example of code that was used,

```
for i = 1:m;
        for j = 1:n;
            h = h + w(i)*f(t(i),tau(j),k);
        end
        g = g + v(j)*h;
end
```

can be re-written,

```
w*f(t,tau,k)*v;
```

where `f(t,tau,k)` is the matrix

$$f(t,tau,k) = \begin{bmatrix} f(t(1), tau(1), k) & \ldots & f(t(1), tau(m), k) \\ f(t(2), tau(1), k) & \ldots & f(t \end{bmatrix}$$

on the edges a graded mesh can be used. The original double integral (4.4)

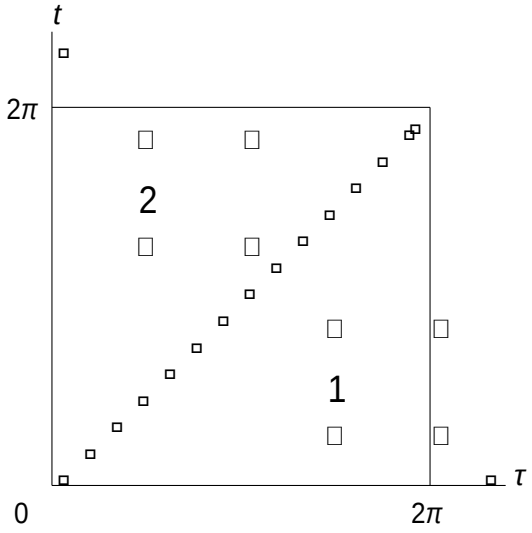

Figure 7.1: Region Before Transformation

can be written,

$$\int_0^{2\pi}\left[\int_0^{2\pi} F(t,\tau)|y'(\tau)|d\tau\right]|y'(t)|dt$$

$$= \int_0^{2\pi}\left[\int_0^{\tau} F(t,\tau)|y'(t)|dt\right]|y'(\tau)|d\tau + \int_0^{2\pi}\left[\int_t^{2\pi} F(t,\tau)|y'(\tau)|d\tau\right]|y'(t)|dt$$

where $F(t,\tau) = \frac{i}{4}H_0^1(k|y(t) - y(\tau)|)$. For the Gaussian Quadrature method, we need to make sure that the correct points are taken from the intervals. Triangle 1 takes points using the integral $\int_0^{2\pi}\left[\int_0^{\tau} F(t,\tau)|y'(t)|dt\right]^p |y'(\tau)|d\tau$ and triangle 2 takes points using $\int_0^{2\pi}\left[\int_t^{2\pi} F(t,\tau)|y'(\tau)|d\tau\right]^p |y'(t)|dt$.

Taking the *Du y Transformation* $\tau = st, \frac{d\tau}{ds} = t$ triangle 2 becomes a rectangle as in Figure 7.2. The singularities being where $\tau = t$ are therefore now at $s = 1$ and $t = 0$. After this transformation there is the integral

$$\int_0^{2\pi}\left[\int_0^{1} F(t,st)|y'(st)|tds\right]^p |y'(t)|dt \qquad (7.1)$$

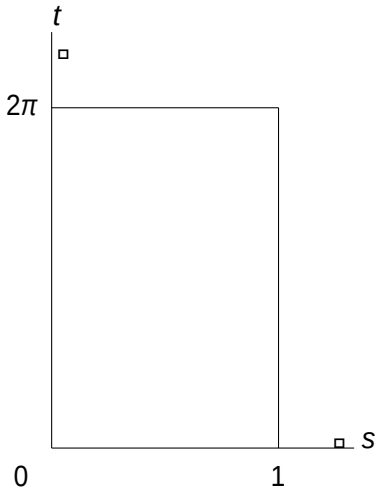

Figure 7.2: Transformed Triangle 2

for triangle 2. The use of a graded mesh is now a lot simpler. Using four different meshes on the rectangle it is possible to have many points near the singularity points, and save on expense by using relatively few points far away from the singularity. Dividing the rectangle as follows (shown in Figure 7.3):

- In the region, $[0, \frac{k-1}{k}] \times [\frac{2\pi}{k}, 2\pi]$ using n by n Gaussian Quadrature points in each dotted area shown in Figure 7.3..
- In the region, $[0, \frac{k-1}{k}] \times [0, \frac{2\pi}{k}]$ using n Gaussian Quadrature points in the s direction. With n Gaussian points every $[\frac{2\pi}{k}(0.15^i), \frac{2\pi}{k}(0.15^{i+1})]$ with $i = \{0, 1, \ldots,$ [illegible] in the t direction for each dotted area.
- In the region, $[\frac{k-}{k}$ [illegible]$] \times [\frac{2\pi}{k}, 2\pi]$ using n Gaussian Quadrature points in the t direction. With n Gaussian Quadrature points every $[1 - \frac{0.15^i}{k}, 1 - {}^{0.15^{i+1}}$

graded mesh. This means that where a lot is going on, we are concentrating a lot of our resources. Therefore the larger oscillations near the singularity can be approximated with greater accuracy.

7.3 The Duffy Transformation for Piecewise Constants

If the τ axis is split up into evenly spaced piecewise constants (see Figure 4.2) then the Duffy transformation can still be used. The transformation works when both integrals start at zero. Taking the j*th* piecewise constant from a_j to b_j, then we have the rectangular area as shown in Figure 7.3 from a_j to b_j on the τ axis and 0 to 2π on the t axis. This can be split into three separate double integrals,

$$\int_0^{2\pi} \int_{b_j}$$

Now, performing the Duffy transformation as before by letting $\tau = st$, $\frac{d\tau}{ds} = t$ this becomes,

$$\int_0^{b_j - a_j} \left(\int_0^1 \kappa(t + a_j, st + a_j)|y'(st + a_j)|tds \right) |y'(t + a_j)|ds.$$

This is a rectangle similar to before, where the singularities are at $t = 0$ and $s = 1$, and the grading can be done as before.

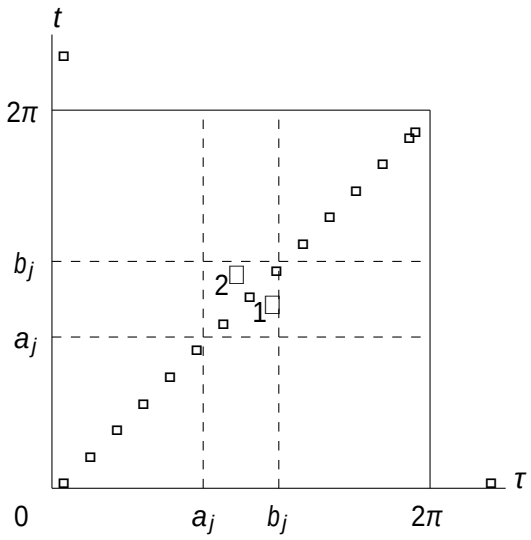

Figure 7.4: Region Before Transformation Using Piecewise Constants

8 Results

We have 1-norm and ∞-norm results for the operator S (i.e. $||S||_{L^1}$ and $||S||_{L^\infty}$). Using these results (shown in Table 8.1) we are able to say how the $||S||$ changes as we increase k for the circle. Table 8.2 shows this for an ellipse of the form $y(\tau) = (2\cos\tau, \sin\tau)$. Upper bounds (Figure 8.1) and lower bounds (Table 5.1) found in Sections 4 and 5 can be used together, and by analysing the region in which the 2-norm can lie, results can be gathered (Table 8.3 and Table 8.4) to determine behaviour of $||S||_{L^2}$.

8.1 1-norm and the ∞-norm

Using the single integrals it was possible to look at how the values changed as k was increased. Looking at the circle first we obtain Table 8.1. From this it can be seen that log_k

k	$\|\|S\|\|_{L^1}$ for $y(\tau) = (\cos\tau, \sin\tau)$	$log_k\|\|S\|\|_{L^1}$ for $y(\tau) = (\cos\tau, \sin\tau)$
1	1.27269909813074	-
2	0.94283092667553	-0.08492901233913
4	0.68817972895875	-0.26957134912594
8	0.49731659257114	-0.33592117619175
⋮	⋮	⋮
512	0.06497404812582	-0.43822139961133
1024	0.04602585988168	-0.44414115146863
2048	0.03258593605505	-0.44905515924963
⋮	⋮	⋮
32768	0.00816349758682	-0.46243979257239
65536	0.00577323389157	-0.46477528813075
131072	0.00408254822470	-0.46684202100476

Table 8.1: Approximation to the 1-norm for the Circle as $k \to \infty$ using the Single Integral

k	$\|\|S\|\|_{L^2}$ for $y(\tau) = (2\cos\tau, \sin\tau)$	$log_k \|\|S\|\|_{L^2}$ for $y(\tau) = (2\cos\tau, \sin\tau)$
1	1.294548530868040	-
2	0.883817805885845	-0.178179098161715
4	0.575077907006125	-0.399085340338845
8	0.384944183600455	-0.459092940931912
16	0.264450390167755	-0.479732747741775
32	0.178648232053725	-0.496961291630781

8.2 The 2-norm

The upper bounds and lower bounds found in Sections 4 and 5 allow us to investigate the 2-norm. The upper bounds for the circle as k increase can be seen in Figure 8.2, and for the ellipse $y(\tau) = (2\cos\tau, sin\tau)$ can be seen in Figure 8.1. The lower bounds as k increase can be seen in Table 5.1. Using these bounds we then have a region where the 2-norm must lie. By taking the mid-point of this region, we can use them to calculate a table as we did for the 1-norm and the ∞-norm. This Table 8.3 shows results for the ellipse. Taking the mid-point as we increase k the value of $log_k||S||_{L^2}$ seems to be approaching $-\frac{1}{2}$. Table 8.4 shows results for the circle, where again the

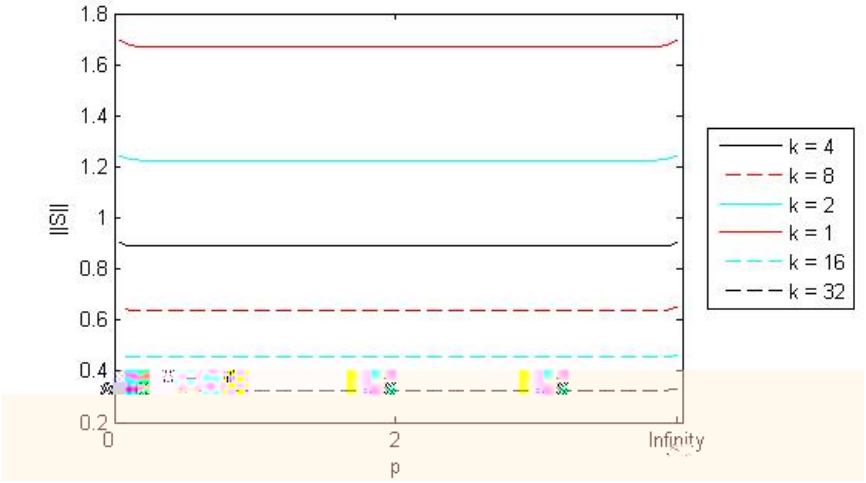

Figure 8.1: Upper Bounds of $||S|_{L^p}$ with k increasing for $y(\tau) = (2\cos\tau, \sin\tau)$

mid-point of the upper and lower bound for $||S||_{L^2}$ has been taking. This time the value seems to be larger than $-\frac{1}{2}$.

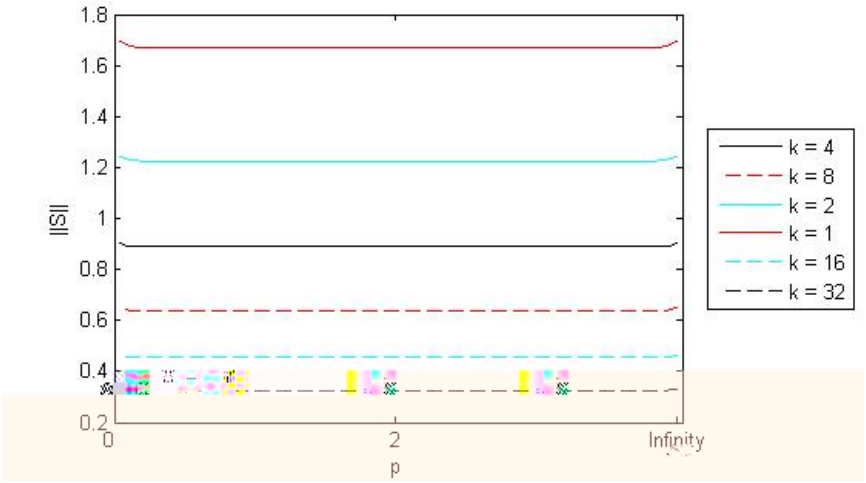

Figure 8.2: Upper Bounds of $||S||_{L^p}$ with k increasing for the Circle

9 Conclusion

The aim of this project was to look at behaviour of the single layer potential, S, which would be a crucial step in analysing the operator A and its

that $||S||_{L^2}$ behaves like $k^{-\frac{1}{2}}$. Then by choosing $\eta = k$ again, this tells us that we have behaviour of $||A||_{L^2}$ like $k^{\frac{1}{2}}$. This is the same result as stated in [3] (see Section 2.2). Similarly, the numerical results for the circle for the 2-norm behave like k^{β} where β seems less than $-\frac{1}{2}$. If we had the result that $||S||_{L^2}$

10.1 Lower Norms

We say that A is bounded below if $M > 0$ such that,

$$||Ax|| \geq M, \quad x \quad X, \text{ with } ||x|| = 1$$

or

$$||Ax|| \geq M||x|| \quad x \quad X.$$

The largest M is called the *lower norm* of A, and is denoted $|A|$. We can also write,

$$|A| = \inf_{||x||=1} |Ax|$$

or

$|A| = \inf_{x=0} \frac{||Ax||}{||x||}$, $M \leq |Ax| \leq C$ if $||x|| = 1$ where M and C are constants.

It is known that

$$|A| = \frac{1}{||A^{-1}||}$$

These lower norms are useful when looking at properties of the sound soft

References

[1] Simon N. Chandler-Wilde and Peter Monk, Wave-Number-Explicit Bounds in Time-Harmonic Scattering (submitted for publication)

[2] Mark Webber, The Point Source Method in Inverse Acoustic Scattering, MSc Dissertation, The University of Reading 2004

[3] V. Dominguez, I.G. Graham and V.P. Smyshylaev, A hybrid numerical-asymptotic boundary integral method for high-frequency acoustic scattering, Bath Institute For Complex Systems PreprΩln(,)-253al-0o93(2604)]TJ-16.969-30.645Td[