High-order quadratures for singular functions and their applications to the evaluation of Fourier and Hankel transforms

Sharad Kapur

YALEU/DCS/TR-1084

April 1995

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Abstract

High-order quadratures for singular functions and their applications to the evaluation of Fourier and Hankel transforms.

Sharad Kapur
Yale University
1995

Evaluation of integrals plays an important role in the numerical solution of many problems in physics, engineering, and other areas; hence the interest in the study and design of rapidly convergent quadrature rules. For example, the discrete Fourier transform (DFT) is a popular numerical tool since it provides an excellent approximation to the continuous Fourier transform of a periodic function. The reason for the accuracy of the approximation is that the trapezoidal rule, which the DFT implements, is superalgebraically convergent for periodic functions.

Numerical quadrature schemes for smooth functions are a well understood subject (see, [11], [3], [28]). When a function is singular, the need for a high order quadrature scheme is met by Gaussian quadrature. Unfortunately, the nodes at which a function is tabulated are often non-Gaussian (such as equispaced nodes, Chebyshev nodes, etc.), for which existing quadrature schemes are inadequate.

In this thesis we introduce a class of quadrature formulae applicable to both non-singular and singular functions, generalizing the classical end-point corrected trapezoidal quadrature rules. While the standard end-point corrected trapezoidal rules are usually derived by means of the Euler Maclaurin formula, their generalizations are obtained as solutions of certain systems of linear algebraic equations. A procedure is developed for the construction of very high-order quadrature rules, applicable to functions with apriori specified singularities, and relaxing the requirements on the distribution of nodes.

We also present two applications based on these high-order quadratures. An algorithm is developed for the rapid evaluation of the Fourier transform of functions with singularities. The algorithm is based on a combination of the fast Fourier transform (FFT) with a quadrature scheme tailored to the singularity. A related algorithm for the fast Hankel transform is also presented. The algorithm decomposes the Hankel transform into a product of two integral operators, the first of which is evaluated rapidly by a combination of the fast cosine transform with a quadrature formula of the type developed in this thesis. The second operator is evaluated rapidly by a combination of a version of the fast multipole method with yet another quadrature formula derived in this thesis. All calculations are performed to full double precision accuracy.

Numerical experiments are presented demonstrating the practical usefulness and efficiency of all the algorithms developed. Tables of quadrature weights are included for singularities of the form $s(x) = |x|^\lambda$ for a variety of values of λ, and $s(x) = log(|x|)$.

High-order quadratures for singular functions and their applications to the evaluation of Fourier and Hankel transforms.

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Sharad Kapur
December, 1995

To my family, my friends and my teachers.

Contents

List of Tables iii

List of Figures iv

1 Introduction 1

1.1 High-order quadrature rules for singular functions 2

1.2 Applications of quadratures to Fourier analysis 2

1.3 Applications of quadratures to the fast Hankel transform 3

2 Corrected trapezoidal rules for singular functions 5

2.1 Introduction 5

2.2 Mathematical Preliminaries 7

2.3 End-point Corrections for Non-singular Functions 14

2.3.1 End-point corrected trapezoidal rules 14

2.4 End-point Corrections for Singular Functions 16

2.4.1 Convergence Rates for Singularites of the forms $|x|^\lambda$ and $log(|x|)$ . . 20

2.4.2 Asymptotic behaviour of correction coefficients as $n \to \infty$ 21

2.4.3 Central Corrections for Singular Functions 24

2.4.4 Central Corrections for Singular Functions $f(x) = \phi(x)s(x)$ 26

2.4.5 Corrected trapezoidal rules for other singularities 27

2.5 Numerical Results 29

2.6 Generalizations and Conclusions 32

3 Fourier transforms of singular functions **34**

3.1 Introduction 34

3.2 Mathematical and numerical preliminaries 36

3.2.1 Corrected trapezoidal quadrature rules for singular functions 36

3.2.2 The Fast Fourier transform 38

3.3 Application of local corrections to the evaluation of Fourier transforms of functions with singularities 38

3.3.1 The corrected discrete Fourier transform 39

3.3.2 Rapid evaluation of the corrected discrete Fourier transform 40

3.4 Application of global corrections to the evaluation of Fourier transforms of functions with singularities 41

3.4.1 Integration of periodic functions 41

3.4.2 The Fast fourier transform of singular functions 42

3.5 The FFT for functions with singularities in two-dimensions 43

3.6 Applications of quadratures to the design of VLSI masks 44

3.6.1 An algorithm applicable to VLSI 47

3.7 Numerical Examples 49

3.8 Generalizations and conclusions 53

4 An algorithm for the fast Hankel transform **57**

4.1 Introduction 57

4.2 Mathematical and Numerical Preliminaries 59

4.2.1 An expression for the Bessel function J_0 59

4.2.2 Chebyshev Polynomials 59

4.2.3 The fast cosine transform 61

4.3 Numerical apparatus 62

4.3.1 The corrected fast cosine transform 62

4.3.2 End-point corrected trapezoidal quadrature rules for singular functions of the form $\frac{F(u)}{(a^2-u^2)^{1/2}}$ 64

4.3.3 A fast multipole method in one-dimension for sums of the form $\sum_{k=1}^{j-1} \frac{\alpha_k}{(y_j^2-x_k^2)^{1/2}}$ 69

4.4 The fast Hankel transform 74

4.4.1 Informal description of the algorithm 74
4.4.2 Detailed description of the algorithm, and complexity analyis 75
4.5 Numerical Results 77
4.6 Generalizations and Conclusions 83

5 Appendix **85**
5.1 Correction weights for Non-singular and Singular functions 85
5.1.1 Quadrature Weights β_k^m for Non-singular Functions 85
5.1.2 Quadrature Weights γ_j^k for Singular Functions 88
5.1.3 Quadrature Weights μ_j^k for Singular Functions 93
5.1.4 Quadrature Weights ρ_j^k for Singular Functions 95

Bibliography **98**

List of Tables

2.1 Convergence of quadrature rules $T^n_{\beta m}$ for non-singular functions 31

2.2 Convergence of quadrature rules $T^n_{\gamma^k \beta m}$ for singular functions (10th order) . 31

2.3 Convergence of quadrature rules $T^n_{\mu^k \beta m}$ for singular functions (10th order) . 31

2.4 Convergence of the quadrature rule $T^n_{\rho^k \beta m}$ for functions $f(x) = \phi(x) log(|x|)$ 31

3.1 Accuracy of the locally corrected FFT for evaluating singular integrals . . . 55

3.2 Applications of quadratures to VLSI 56

4.1 Accuracy of the uncorrected trapezoidal rule for evaluating integrals $\int_0^{2\pi} sin(bx) \cdot cos(bx) \cdot e^{-x^2}$ 81

4.2 Accuracy of the corrected trapezoidal rule for evaluating the integrals $\int_0^{2\pi} sin(bx) \cdot cos(bx) \cdot e^{-x^2}$ 81

4.3 Accuracy of the quadrature rule $T^j_{\nu_j}$ for evaluating the integrals $\int_{-\pi}^{\pi} \cos(b \cdot u)/(\pi^2 - u^2)^{\frac{1}{2}} du$ 81

4.4 Numerical results for Example 1 (single precision) 82

4.5 Numerical results for Example 1 (double precision) 82

4.6 Numerical results for Example 2 (single precision) 82

4.7 Numerical results for Example 2 (double precision) 82

List of Figures

4.1 Well-separated intervals on the line. 70
4.2 Hierarchy of subintervals. 74
4.3 Correction nodes 75

Chapter 1

Introduction

The evaluation of integrals of singular functions plays an important role in the study of many physical systems. For example, the need for accurate numerical evaluation of integrals of singular functions is found in areas such as computational fluid dynamics [8], potential theory for the Laplace equation [16], crystal growth [15], integral equations [1], and many others; hence the interest in the design of high-order quadrature rules. When a function is singular, the need for a rapidly convergent scheme is met by Gaussian quadrature. Unfortunately, the nodes at which the function is tabulated are often non-Gaussian (such as equispaced nodes, Chebyshev nodes, etc.), and alternate quadrature schemes have to be devised. In recent years, various quadrature schemes have been developed (see for example, [1], [14], [26]) for the integration of singular functions. Many of these procedures provide satisfactory low order quadratures; for higher orders, the quadrature weights grow rapidly, rendering the schemes useless.

In this thesis we introduce a class of quadrature formulae applicable to both non-singular and singular functions, generalizing the classical end-point corrected trapezoidal quadrature rules. While the standard end-point corrected trapezoidal rules are usually derived by means of the Euler Maclaurin formula, their generalizations are obtained as solutions of certain systems of linear algebraic equations. A procedure is developed for the construction of very high-order quadrature rules, applicable to functions with apriori specified singularities, and relaxing the requirements on the distribution of nodes.

1.1 High-order quadrature rules for singular functions

In Chapter 2, a group of quadrature formulae is presented applicable to functions with singularities, generalizing the classical end-point corrected trapezoidal quadrature rules. More specifically, high-order corrected trapezoidal quadrature rules are developed to approximate definite integrals of singular functions $f : [a, b] \to \mathbf{R}$ of the form

$$f(x) = \phi(x)s(x) + \psi(x), \tag{1.1}$$

and

$$f(x) = \phi(x)s(x), \tag{1.2}$$

where $\phi(x), \psi(x) \in c^k[a, b]$, and s is an integrable function with a singularity on the interval $[a, b]$.

Remark 1.1 The one drawback of the approach is the fact that in order to obtain a set of weights for a given singularity, a system of linear algebraic equations has to be solved, whose matrix can be extremely ill-conditioned. For example, in order to obtain quadrature rules of order 12, we had to solve $12X12$ systems of equations; the condition number was of the order 10^{20}. Fortunately, for each singularity, such a system has to be solved only once, after which the weights are tabulated; we used MAPLE to perform such calculations using 60 significant digits.

In this thesis we also present two applications based on these high-order quadratures. The first application is an algorithm for the rapid evaluation of the Fourier transform of functions with singularities.

1.2 Applications of quadratures to Fourier analysis

Fourier techniques have been an extremely important and popular analytical tool in mathematics and physics for more than two centuries. The introduction of the fast Fourier transform (FFT) algorithm in the 1960s has greatly broadened the scope of application of the Fourier transform to data handling, and has also brought prominence to the discrete

Fourier transform (DFT). The DFT is often used as a quadrature rule for the evaluation of integrals of the form

$$\int_0^{2\pi} f(x)e^{i\omega x}dx, \tag{1.3}$$

for various values of ω. It is well known that the DFT is an extremely effective numerical tool of applied analysis, for the approximation of integrals of the form (1.3), when the function f is periodic. However, when the function has a jump discontinuity, or is singular, the accuracy of the DFT is significantly reduced. This loss of accuracy is due to the fact that the DFT, which implements the trapezoidal rule, is slowly convergent for non-smooth functions; the numerical error that results from the discontinuity is on the order of $\frac{1}{n}$ for a problem of size n. Such errors make even single precision calculations, especially in higher dimensions, prohibitively expensive. Richardson extrapolation alleviates the situation, but double precision calculations are still virtually impossible.

In Chapter 3, we present an algorithm for the rapid evaluation of the Fourier transform of functions with singularities. The algorithm is based on a combination of the fast Fourier transform (FFT) with a quadrature scheme tailored to the singularity. More specifically, we observe that the quadrature scheme derived for singular functions can be applied, in a simple and straightforward manner, to numerically integrate singular functions of the form $f(x) \cdot e^{iwx}$. In this case resulting systems of linear algebraic equations are well-conditioned, and in fact can be solved rapidly by means of the FFT; thus, the correction weights do not need to be precomputed and tabulated.

The second application of our quadrature rules is a fast Hankel transform.

1.3 Applications of quadratures to the fast Hankel transform

Hankel transforms are frequently encountered in applied mathematics and computational physics. Their applications include vibrations of a circular membrane, flow of heat in a circular cylinder, wave propagation in a three-dimensional medium and many others. However, attempts to use Hankel transforms as a numerical tool (as opposed to analytical apparatus) tend to meet with a serious difficulty: given a function $f : [0, A] \to \mathbf{R}$, tabulated at N nodes, it takes $O(N^2)$ operations to obtain the numerical Hankel transform

$$\int_0^A f(x)J_0(a \cdot x)dx, \tag{1.4}$$

for N values of a. In other words, unlike the Fourier series, the Chebyshev expansion or the Legendre series, the Bessel functions do not have a fast transform associated with them. Therefore, whenever possible, the Hankel transform is avoided in favor of an expansion for which a fast transform exists.

In chapter 4, we develop a procedure for the rapid evaluation of integrals of the form (1.4), to any degree of precision, requiring CPU time proportional to $N \log N$. More specifically, suppose that $h = \frac{A}{N-1}$, $x_i = ih$, $a_j = \frac{\pi j}{(N-1)h}$, and $f : [0, A] \to \mathbf{R}$ is a function tabulated at N equispaced nodes $x_0, x_1, ..., x_{N-1}$. Then the integrals

$$g(a_j) = \int_0^A f(x)J_0(a_j x)dx, \tag{1.5}$$

are computed for all $j = 0, 1, 2..., N-1$, in $O(N \log N)$ operations.

Our algorithm for the Hankel transform is based on several well known facts from classical analysis. The algorithm decomposes the Hankel transform into a product two integral operators, the first of which is evaluated rapidly by a combination of the fast cosine transform with quadrature formula of the type developed in this thesis. The second operator is evaluated rapidly by a combination of a version of the fast multipole method with yet another quadrature formula derived in this thesis. All calculations are performed to full double precision accuracy.

Chapter 2

Corrected trapezoidal rules for singular functions

2.1 Introduction

The trapezoidal rule is known to be an easy and numerically stable means for numerical integration. If a function is periodic and analytic on the interval of integration, the trapezoidal rule converges exponentially fast (see, for example, [11]). However, for non-periodic functions the trapezoidal rule is second order convergent, and end-point corrections are often used to improve the convergence rate. A standard end-point corrected trapezoidal rule is fourth order convergent, and is given by the formula

$$\int_a^b f(x)dx = \sum_{i=1}^{n-2} f(x_i) + \frac{f(x_0)+f(x_{n-1})}{2} + \frac{h}{24}(-f(x_{-1})+f(x_1)+f(x_{n-2})-f(x_n)), \tag{2.1}$$

where, $h=(b-a)/(n-1)$ and $x_i = a+ih$ for $i=0,1,2,...,n-1$ (see, for example [2]).

More recently, the Euler-Maclaurin formula is used in [1] to obtain a high-order end-point corrected trapezoidal rule of the form

$$T_\alpha^n(f) = \sum_{i=1}^{n-2} f(x_i) + \frac{f(x_0)+f(x_{n-1})}{2} + h\sum_{j=1}^{m} \alpha_j(f(x_{n-j})-f(x_j)), \tag{2.2}$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$ are coefficients such that

$$| T^n_\alpha(f) - \int_a^b f(x)dx |< \frac{c}{n^m}. \tag{2.3}$$

for some $c > 0$.

The scheme of [1] provides satisfactory quadratures upto order 12; for higher orders, the coefficients α grow rapidly, rendering the scheme useless. In this chapter we develop a different class of end-point corrected trapezoidal rules, whereby the growth of correction weights is suppressed, enabling the construction of end-point corrected trapezoidal rules of arbitrarily high order for non-singular functions.

In [26], end-point corrected quadrature formulae are developed to approximate definite integrals of singular functions $f : [a, b] \to R^1$ of the form

$$f(x) = \phi(x)s(x) + \psi(x), \tag{2.4}$$

and

$$f(x) = \phi(x)s(x), \tag{2.5}$$

where $a \leq 0 \leq b$, $\phi(x), \psi(x) \in c^k[a, b]$, and $s(x) \in c[a, b]$ is an integrable function with a singularity at 0. The procedure developed in [26] provides satisfactory quadratures only upto order 4; for higher orders, the quadrature weights grow rapidly, also rendering the scheme useless. In this chapter we construct a different class of end-point corrected trapezoidal rules, whereby the growth of quadrature weights is partially suppressed for functions of the form (2.4), obtaining useful quadratures of order upto 12; and completely suppressed for functions of the form (2.5), providing quadratures of arbitrarily high order. Moreover, we obviate the programming inconvenience associated with the procedure developed in [26], which requires that functional information be tabulated on a grid finer than that required for the uncorrected trapezoidal rule.

Remark 2.1 The approach of this chapter is somewhat related to that of [14]. However, [14] constructs quadratures in higher dimensions, and these quadratures are of relatively low order. In this thesis, we construct one-dimesional rules of very high order. Furthermore, most rules of this thesis are "standard" in the sense that the correction coefficients do not depend on the number of nodes in the trapezoidal rule being corrected, or on the sampling interval.

2.2 Mathematical Preliminaries

In this section we summarize some well-known results to be used in the reminder of the thesis. Lemmas 2.1, 2.2 and 2.3 can be found, for example, in [2].

Definition 2.1 *Suppose that a, b are a pair of real numbers such that $a < b$, and that $n \geq 2$ is an integer. For a function $f : [a, b] \to R^1$, we define the n-point trapezoidal rule $T_n(f)$ by the formula*

$$T_n(f) = h(\sum_{i=0}^{n-1} f(a + ih) - (\frac{f(a) + f(b)}{2})), \tag{2.6}$$

with

$$h = (b - a)/(n - 1). \tag{2.7}$$

The following lemma provides an error estimate for the approximation to the integral given by the trapezoidal rule. Along with Lemma 2.2, it can be found, for example in [2].

Lemma 2.1 *(Euler-Maclaurin formula) Suppose that a, b are a pair of real numbers such that $a < b$, and that $m \geq 1$ is an integer. Further, let B_k denote the Bernoulli numbers*

$$B_2 = \frac{1}{6}, B_4 = \frac{-1}{30}, B_6 = \frac{1}{42}, \ldots, . \tag{2.8}$$

If $f \in c^{2m+2}[a, b]$ (i.e., f has $2m + 2$ continuous derivatives on $[a, b]$), then there exists a real number ξ, with $a < \xi < b$, such that

$$\int_a^b f(x)dx = T_n(f) + \sum_{l=1}^{m} \frac{h^{2l} B_{2l}}{(2l)!}(f^{(2l-1)}(b) - f^{(2l-1)}(a)) - \frac{h^{2m+2} B_{2m+2}}{(2m+2)!} f^{2m+2}(\xi). \tag{2.9}$$

The following well-known lemma provides an error estimate for Lagrange interpolation.

Lemma 2.2 *(Lagrange interpolation formula) Suppose that a, b are a pair of real numbers such that $a < b$, $m \geq 3$ be an odd integer, and $f \in c^m[a,b]$. Furthermore, let h be defined in (2.7), and f be tabulated at equispaced points, $x_k = \frac{b-a}{2} + kh$. Then for any real number p there exists a real number ξ, $-mh < \xi < mh$, such that*

$$f(x_0 + ph) = \sum_{k=\frac{-(m-1)}{2}}^{\frac{(m-1)}{2}} A_k^m(p) f(x_k) + R_{m-1}, \tag{2.10}$$

with

$$A_k^m(p) = \frac{(-1)^{\frac{m-1}{2}+k}}{(\frac{m-1}{2}+k)!(\frac{m-1}{2}-k)!(p-k)} \prod_{t=0}^{m-1} (p + \frac{m-1}{2} - t), \tag{2.11}$$

and

$$R_{m-1} = \frac{1}{m!} \prod_{k=-\frac{m-1}{2}}^{\frac{m-1}{2}} (p-k) h^m f^{(m)}(\xi). \tag{2.12}$$

Lemma 2.3 *If $f : [a,b] \to R^1$ is a function satisfying the conditions of Lemma 2.2, and the coefficients $D_{i,k}^m$ are given by the formula*

$$D_{i,k}^m = \frac{\partial^{(2i-1)}}{\partial p^{(2i-1)}} (A_k^m(p)) \,|^{p=0}, \tag{2.13}$$

then

$$f^{(2i-1)}(x_0) = \sum_{k=-\frac{m-1}{2}}^{\frac{m-1}{2}} \frac{D_{i,k}^m}{h^{2i-1}} f(x_k) + O(h^m), \tag{2.14}$$

for any m, i such that $-\frac{m-1}{2} \leq k \leq \frac{m-1}{2}$, and $1 \leq k \leq \frac{m-1}{2}$.

Proof. The proof is as an immediate consequence of (2.10) and (2.13). □

Lemma 2.4 *Suppose that m, l, k are integers, and the coefficients $a^m_{k,l}$ are defined by the recurrence relation*

$$a^3_{1,1} = 1, \tag{15-a}$$
$$a^3_{1,2} = 1, \tag{15-b}$$
$$a^{2k+1}_{k,l} = (k-k^2)a^{2k-1}_{k-1,l} + a^{2k-1}_{k-1,l-1} + a^{2k-1}_{k-1,l-2}, \tag{15-c}$$
$$a^{m+2}_{k,l} = a^m_{k,l-2} - (\frac{m+1}{2})^2 a^m_{k,l}, \tag{15-d}$$

with $a^m_{k,l} = 0$, for all $k \le 0$, or $l \le 0$, or $m \le 1$. Then

$$A^m_k(p) = \frac{(-1)^{\frac{m-1}{2}+k}}{(\frac{m-1}{2}+k)!(\frac{m-1}{2}-k)!}\sum_{l=1}^{\frac{m-1}{2}} a^m_{k,l}p^l, \tag{16}$$

for any odd $m \ge 3$, $1 \le k \le \frac{m-1}{2}$ and $A^m_k(p)$ is defined by (2.11).

Proof. Due to (2.11),

$$A^m_k(p) = \frac{(-1)^{\frac{m-1}{2}+k}}{(\frac{m-1}{2}+k)!(\frac{m-1}{2}-k)!}C^m_k(p), \tag{2.17}$$

where

$$C^m_k(p) = \frac{1}{(p-k)}\prod_{t=0}^{m-1}(p+\frac{m-1}{2}-t). \tag{2.18}$$

Thus it is sufficient to show that

$$C^m_k(p) = \sum_{l=1}^{\frac{m-1}{2}} a^m_{k,l}p^l. \tag{2.19}$$

This will be shown by induction. Indeed, if $m = 3$ then, due to (2.18),

$$C^3_1(p) = p^2 + p, \tag{2.20}$$

which is equivalent to (15-a),(15-b).

Assume now that for some m, k such that $-\frac{m-1}{2} \le k \le \frac{m-1}{2}$,

$$C^m_k(p) = \sum_{l=1}^{\frac{m-1}{2}} a^m_{k,l}p^l. \tag{2.21}$$

Combining (2.18) and (2.21), we have

$$\begin{aligned} C_k^{m+2}(p) &= (p+\frac{m+1}{2})(p-\frac{m+1}{2})\sum_{l=1}^{\frac{m-1}{2}} a_{k,l}^m p^l \\ &= (p^2-(\frac{m+1}{2})^2)\sum_{l=1}^{\frac{m-1}{2}} a_{k,l}^m p^l \\ &= \sum_{l=1}^{\frac{m-1}{2}} a_{k,l}^m p^{l+2} - (\frac{m+1}{2})^2 \sum_{l=1}^{\frac{m-1}{2}} a_{k,l}^m p^l, \end{aligned} \tag{2.22}$$

which is equivalent to (15-d).

Now, assume that for some k

$$C_k^{2k+1}(p) = \sum_{l=1}^{k} a_{k,l}^{2k+1} p^l. \tag{2.23}$$

Combining (2.23) and (2.18), we have

$$\begin{aligned} C_k^{2k+3}(p) &= (p-k)(p-(k+1))\sum_{l=1}^{k} a_{k,l}^{2k+1} p^l \\ &= (p^2+p-(k^2+k))\sum_{l=1}^{k} a_{k,l}^{2k+1} p^l \\ &= \sum_{l=1}^{k} a_{k,l}^{2k+1} p^{l+2} + \sum_{l=1}^{k} a_{k,l}^{2k+1} p^{l+1} - (k^2+k)\sum_{l=1}^{k} a_{k,l}^{2k+1} p^l, \end{aligned} \tag{2.24}$$

which is equivalent to (15-c). □

Lemma 2.5 *Suppose that $m \geq 3$ is odd. Then,*

$$D_{i,k}^m = \frac{(-1)^{\frac{m-1}{2}+k}}{(\frac{m-1}{2}+k)!(\frac{m-1}{2}-k)!} a_{k,2i-1}^m (2i-1)!, \tag{2.25}$$

for any k, i such that $-\frac{m-1}{2} \leq k \leq \frac{m-1}{2}$, and $1 \leq i \leq \frac{m-1}{2}$,

with the coefficients $a_{k,l}^m$ defined by the recurrence relation in Lemma 2.4.

Proof. Substituting (16) into (2.13), we immediately obtain

$$D_{i,k}^m = \frac{(-1)^{\frac{m-1}{2}+k}}{(\frac{m-1}{2}+k)!(\frac{m-1}{2}-k)!} \frac{\partial^{(2i-1)}}{\partial p^{(2i-1)}} \sum_{l=1}^{m-1} a_{k,l}^m p^l \,|^{p=0}$$

$$= \frac{(-1)^{\frac{m-1}{2}+k}}{(\frac{m-1}{2}+k)!(\frac{m-1}{2}-k)!} a^m_{k,2i-1}(2i-1)!. \quad (2.26)$$

□

The following six lemmas provide identities which are used in the proof of Theorem 2.13.

Lemma 2.6 *If $k \geq 2$ is an integer and $a^m_{k,l}$ is defined in Lemma 2.4.*

$$\mid (l)! \cdot a^{2k+1}_{k,l} \mid \ < \ \mid (l+2)! \cdot a^{2k+1}_{k,l+2} \mid, \quad (2.27)$$

for all $l = 1, 2, ..., 2k-3$.

Proof.

If $k = 2$, and $l = 1$ then $\mid (1)! \cdot a^5_{2,1} \mid = 2$, $\mid (3)! \cdot a^5_{2,3} \mid = 12$, and therefore (2.27) is obviously true.

Now, assume that

$$\mid (l)! \cdot a^{2k+1}_{k,l} \mid \ < \ \mid (l+2)! \cdot a^{2k+1}_{k,l+2} \mid, \quad (2.28)$$

for some $k \geq 2$ and all $l = 1, 2, ..., 2k-3$.

Now, due to (15-a), (15-b), (15-c), and (15-d),

$$(l)! \cdot a^{2k+3}_{k+1,l} = (((k+1)-(k+1)^2)a^{2k+1}_{k,l} + a^{2k+1}_{k,l-1} + a^{2k+1}_{k,l-2}) \cdot (l!), \quad (2.29)$$

and

$$(l+2)! \cdot a^{2k+3}_{k+1,l+2} = (((k+1)-(k+1)^2)a^{2k+1}_{k,l+2} + a^{2k+1}_{k,l+1} + a^{2k+1}_{k,l}) \cdot (l+2)!. \quad (2.30)$$

Finally, combining (2.28), (2.29), and (2.30) we easily obtain

$$\mid (l)! \cdot a^{2k+3}_{k+1,l} \mid \ < \ \mid (l+2)! \cdot a^{2k+3}_{k+1,l+2} \mid, \quad (2.31)$$

for all $l = 1, 2, ..., 2k-1$. □

Lemma 2.7 *If $k \geq 2$ is an integer, and $a^m_{k,l}$ is defined in Lemma 2.4 then*

$$\mid (l)! \cdot a^m_{k,l} \mid \ < \ \mid (l+2)! \cdot a^m_{k,l+2} \mid, \quad (2.32)$$

for all $m \geq 2k+1$ and $l = 1, 2, ..., 2k-3$.

Proof. Lemma (2.6) establishes the base case, i.e., that (2.32) is true when $m = 2k+1$. Now, assume that

$$| (l)! \cdot a^m_{k,l} | \quad < \quad | (l+2)! \cdot a^m_{k,l+2} |, \tag{2.33}$$

for some odd $m \geq 2k+1$, and all $l = 1, 2, ..., 2k-3$.

Now, due to (15-a), (15-b), (15-c), and (15-d),

$$(l)! \cdot a^{m+2}_{k,l} = (a^m_{k,l-2} - (\frac{m+1}{2})^2 a^m_{k,l}) \cdot (l!), \tag{2.34}$$

and

$$(l+2)! \cdot a^{m+2}_{k,l+2} = (a^m_{k,l} - (\frac{m+1}{2})^2 a^m_{k,l+2}) \cdot (l+2)!. \tag{2.35}$$

Finally, combining (2.33), (2.34), and (2.35) we easily obtain

$$(l)! \cdot | a^{m+2}_{k,l} | \quad < \quad | (l+2)! \cdot a^{m+2}_{k+1,l+2} |, \tag{2.36}$$

for all $l = 1, 2, ..., 2k-3$. □

Lemma 2.8 *If m, k are integers such that $m \geq 3$ is odd, and $-\frac{m-1}{2} \leq k \leq \frac{m-1}{2}$, then*

$$| (1)! \cdot a^m_{k,1} |<| (3)! \cdot a^m_{k,3} |<| (5)! \cdot a^m_{k,5} |< ... <| (m-2)! \cdot a^m_{k,m-2} | . \tag{2.37}$$

Proof. This Lemma follows directly from Lemma 2.6 and Lemma 2.7 □

Lemma 2.9 *If $m \geq 3$ is odd, then*

$$\frac{(m-1)(m-2)!}{2((\frac{m-1}{2})!)^2} < \frac{(2\pi)^{m-1}}{4}. \tag{2.38}$$

Proof. If $m = 3$ then obviously $1 < \frac{(2\pi)^2}{4}$.

Now, assume that for some odd $m \geq 3$,

$$\frac{(m-1)(m-2)!}{2((\frac{m-1}{2})!)^2} < \frac{(2\pi)^{m-1}}{4}. \tag{2.39}$$

Obviously,

$$\frac{(m+1)(m)}{(\frac{(m+1)}{2})^2} = \frac{4m}{(m+1)} < (2\pi)^2, \tag{2.40}$$

and combining (2.39) and (2.40) we obtain

$$\frac{(m+1)(m)}{(\frac{(m+1)}{2})^2}\frac{(m-1)(m-2)!}{2((\frac{m-1}{2})!)^2} < \frac{(2\pi)^{m-1}}{4}(2\pi)^2, \tag{2.41}$$

which is equivalent to

$$\frac{(m+1)(m)!}{2((\frac{m+1}{2})!)^2} < \frac{(2\pi)^{m+1}}{4}. \tag{2.42}$$

Now, the conclusion of the lemma is an immediate consequence of (2.39) and (2.42). □

Lemma 2.10 *If $m \geq 3$ is odd then*

$$\mid D^m_{i,k} \mid < \frac{(2\pi)^{m-1}}{4}, \tag{2.43}$$

for any k, i such that $-\frac{m-1}{2} \leq k \leq \frac{m-1}{2}$, and $1 \leq i \leq \frac{m-1}{2}$.

Proof. Combining Lemmas 2.5, 2.6, 2.7, and 2.8, it is easy to see that

$$\mid D^m_{1,k} \mid < \mid D^m_{2,k} \mid < \dots < \mid D^m_{\frac{m-1}{2},k} \mid . \tag{2.44}$$

Consequently, it is sufficient to show that

$$\mid D^m_{\frac{m-1}{2},k} \mid < \frac{(2\pi)^{m-1}}{4}. \tag{2.45}$$

First we observe that (obviously) for any k such that $-\frac{m-1}{2} \leq k \leq \frac{m-1}{2}$,

$$\frac{k(m-2)!}{(\frac{m-1}{2}+k)!(\frac{m-1}{2}-k)!} < \frac{(m-1)(m-2)!}{2((\frac{m-1}{2})!)^2}. \tag{2.46}$$

Then, we combine (15-a), (15-b), (15-c), (15-d), and (2.25) to obtain

$$\mid D^m_{\frac{m-1}{2},k} \mid = \frac{k(m-2)!}{(\frac{m-1}{2}+k)!(\frac{m-1}{2}-k)!}. \tag{2.47}$$

Now, (2.45) follows immediately from the combination of (2.46), (2.47), and Lemma 2.9. □

Lemma 2.11 *For any $l \geq 1$ the Bernoulli number B_{2l} satisfies the inequality*

$$\mid \frac{B_{2l}}{(2l)!} \mid < \frac{4}{(2\pi)^{2l}}. \tag{2.48}$$

Proof. As is well known (see for example, [2]), for any $l \geq 1$

$$B_{2l} = \frac{(-1)^{l-1}2(2l)!}{(2\pi)^{2l}} \sum_{k=1}^{\infty} \frac{1}{k^{2l}}, \tag{2.49}$$

and

$$\sum_{k=1}^{\infty} \frac{1}{k^{2l}} < 2. \tag{2.50}$$

Now, the conclusion of the lemma is an immediate consequence of (2.49) and (2.50). □

The proof of the following lemma can be found in [26].

Lemma 2.12 *Suppose that* $m \geq 1$, $s \in c^m(0,1]$ *possesses a finite integral on the interval* $[0,1]$, *and that* $s^{(m)}(x)$ *is monotonic in some neighborhood of* 0. *Then the product* $x \cdot s(x)$ *is bounded on* $[0,1]$. *Suppose further that* $w \in c^m[0,1]$ *is such that* $w(0) = w'(0) = w''(0) = \ldots = w^{(m)}(0) = 0$. *Then the function* $\psi(x) = s(x) \cdot w(x)$ *is defined on the closed interval* $[0,1]$, *and* $\psi(0) = \psi'(0) = \psi''(0) = \ldots = \psi^{(m)}(0) = 0$.

2.3 End-point Corrections for Non-singular Functions

2.3.1 End-point corrected trapezoidal rules

While the authors have failed to find the contents of this section in the literature, it is an immediate consequence of well-known facts from classical analysis. We present it here for completeness, and because we found the resulting high-order quadrature rules quite useful (see Section 5.1.1).

Suppose that n, m, are a pair of integers with $m \geq 3$ and odd, and $n \geq 2$. Further, suppose that a, b are a pair of real numbers such that $a < b$, $h = (b-a)/(n-1)$, and $f : [a - mh, b + mh] \to R^1$ is an integrable function. We define the corrected trapezoidal rule $T^n_{\beta m}$ for non-singular functions by the formula

$$T^n_{\beta^m}(f) = T_n(f) + h \sum_{k=-\frac{m-1}{2}}^{\frac{m-1}{2}} (f(b+kh) - f(a+kh))\beta^m_k. \tag{2.51}$$

The real coefficients β_k^m are given by the formula

$$\beta_k^m = \sum_{l=1}^{\frac{m-1}{2}} \frac{D_{l,k}^m B_{2l}}{(2l)!}, \tag{2.52}$$

where $D_{l,k}^m$ are defined in (2.13) (also, see (2.25)) and B_{2l} are the Bernoulli numbers.

We will say that the rule $T_{\beta^m}^n$ is of order m if for any $f \in c^m[a-mh, b+mh]$, there exists a real number $c > 0$ such that

$$\mid T_{\beta^m}^n(f) - \int_a^b f(x)dx \mid < \frac{c}{n^m}. \tag{2.53}$$

Theorem 2.13 *If $m \geq 3$ is an odd integer then for any k such that $-\frac{m-1}{2} \leq k \leq \frac{m-1}{2}$,*

$$\mid \beta_k^m \mid < \frac{m-1}{2}, \tag{2.54}$$

where the coefficients β_k^m are defined in (2.52).

Proof. Combining Lemma 2.10 and Lemma 2.11 we immediately observe that

$$\mid \frac{D_{l,k}^m B_{2l}}{(2l)!} \mid < 1, \tag{2.55}$$

and hence

$$\mid \beta_k^m \mid = \sum_{l=1}^{\frac{m-1}{2}} \frac{D_{l,k}^m B_{2l}}{(2l)!} < \frac{m-1}{2}. \tag{2.56}$$

□

Remark 2.2 A somewhat more involved argument shows that in fact $|\beta_k^m| < 1$ for all k, m; empirically this can also be seen from the tables in Section 5.1.1 below. However, for the purposes of this thesis (2.56) is sufficient.

Theorem 2.14 *Suppose that m, n are a pair of integers with $m \geq 3$ and odd, and $n \geq 2$. Further, suppose that a, b are a pair of real numbers such that $a < b$. Then, the end-point corrected trapezoidal rule $T_{\beta^m}^n$ is of order m, i.e., for any $f : [a-mh, b+mh] \rightarrow R^1$ such that $f[a-mh, b+mh] \in c^m[a-mh, b+mh]$, there exists a real number $c > 0$ such that*

$$\mid T_{\beta^m}^n(f) - \int_a^b f(x)dx \mid < \frac{c}{n^m}. \tag{2.57}$$

Proof. Combining (2.52) and (2.51), we obtain

$$\begin{aligned} T^n_{\beta^m}(f) &= T_n(f) + h \sum_{k=-\frac{m-1}{2}}^{\frac{m-1}{2}} (f(b+kh) - f(a+kh)) \sum_{l=1}^{\frac{m-1}{2}} \frac{D^m_{l,k} B_{2l}}{(2l)!} \\ &= T_n(f) + \sum_{l=1}^{\frac{m-1}{2}} \frac{h^{2l} B_{2l}}{(2l)!} \Big(\sum_{k=-\frac{m-1}{2}}^{\frac{m-1}{2}} \frac{D^m_{l,k}(f(b+kh) - f(a+kh))}{h^{2l-1}} \Big). \end{aligned} \quad (2.58)$$

Combining (2.14) and (2.58), we have

$$T^n_{\beta^m}(f) = T_n(f) + \sum_{l=1}^{\frac{m-1}{2}} \frac{h^{2l} B_{2l}}{(2l)!} (f^{(2l-1)}(b) - f^{(2l-1)}(a) - 2R^{(2l-1)}_{m-1}). \quad (2.59)$$

Finally, combining (2.59) with Lemma 2.1, we observe that for some $a < \xi < b$,

$$T^n_{\beta^m}(f) = \int_a^b f(x)dx + 2R^{(2l-1)}_{m-1} + \frac{h^m B_m}{m!} f^m(\xi), \quad (2.60)$$

and the theorem immediately follows from (2.60). □

Remark 2.3 It is easy to see that for $m \geq 3$ and odd, and any k such that $-\frac{m-1}{2} \leq k \leq \frac{m-1}{2}$, $D^m_{i,-k} = -D^m_{i,k}$, and $D^m_{i,0} = 0$ (due to (2.13)), and hence $\beta^m_{-k} = -\beta^m_k$ and $\beta^m_0 = 0$ (due to 2.52). Now, instead of (2.51) one could define the end-point corrected trapezoidal rule by the formula

$$T^n_{\beta^m}(f) = T_n(f) + h \sum_{k=1}^{\frac{m-1}{2}} (f(b+kh) - f(b-kh) - f(a+kh) + f(a-kh))\beta^m_k. \quad (2.61)$$

2.4 End-point Corrections for Singular Functions

In this section we construct a group of quadrature formulae for end-point singular functions, generalizing the classical end-point corrected trapezoidal rules. The actual values of end-point corrections are obtained for each singularity as a solution of a system of linear algebraic equations. All the rules developed in this section are simple extensions of the corrected trapezoidal rule $T^n_{\beta^m}$ developed in the preceding section.

A right-end corrected trapezoidal rule $T^n_{R\beta^m}$ is defined by the formula

$$T^n_{R\beta^m}(f) = h(\frac{f(x_{n-1})}{2} + \sum_{i=1}^{n-2} f(x_i)) \quad + \quad h \sum_{k=1}^{\frac{m-1}{2}} (f(b+kh) - f(b-kh))\beta^m_k, \qquad (2.62)$$

where $f(0, b+mh] \to R^1$ is an integrable function, n, m are a pair of natural numbers with $m \geq 3$ and odd, the coefficients β^m_k are given by (2.52), and

$$\begin{aligned} h &= \frac{b}{n-1}, \\ x_i &= ih. \end{aligned} \qquad (2.63)$$

We will say that the rule $T^n_{R\beta^m}$ is of right-end order $m \geq 3$ if for any $f \in c^{m+1}[0, b+mh]$ such that $f(0) = f'(0) = ... = f^{(m)}(0) = 0$, there exists $c > 0$ such that

$$| T^n_{R\beta^m}(f) - \int_0^b f(x)dx | < \frac{c}{n^m}. \qquad (2.64)$$

It easily follows from Theorem 2.14 that $T^n_{R\beta^m}$ is of right-end order m.
Similarly, a left-end corrected trapezoidal rule $T^n_{L\beta^m}$ is defined by the formula

$$T^n_{L\beta^m}(f) = h(\frac{f(x_{-(n-1)})}{2} + \sum_{i=1}^{n-2} f(x_{-i})) \quad + \quad h \sum_{k=1}^{\frac{m-1}{2}} (-f(-b+kh) + f(-b-kh))\beta^m_k, \ (2.65)$$

where $f[-b-mh, 0) \to R^1$ is an integrable function, n, m are a pair of natural numbers with $m \geq 3$ and odd, the coefficients β^m_k are given by (2.52), and h, x_i are defined by (2.63). We will say that the rule $T^n_{L\beta^m}$ is of left-end order $m \geq 3$ if for any $f \in c^{m+1}[-b-mh, 0)$ such that $f(0) = f'(0) = ... = f^{(m)}(0) = 0$, there exists $c > 0$ such that

$$| T^n_{L\beta^m}(f) - \int_{-b}^0 f(x)dx | < \frac{c}{n^m}. \qquad (2.66)$$

It also easily follows from Theorem 2.14 that $T^n_{L\beta^m}$ is of left-end order m.

Suppose now that the function $f(-kh, b+mh] \to R^1$ is of the form

$$f(x) = \phi(x)s(x) + \psi(x), \qquad (2.67)$$

with $\phi, \psi \in c^k(-kh, b+mh]$, and $s \in c(-kh, b+mh]$ an integrable function with a singularity at 0. For a finite sequence $\alpha = (\alpha_{-k}, \alpha_{-(k-1)}, \alpha_{-1}, \alpha_1, ..., \alpha_k)$ and $T^n_{R\beta^m}$ defined in (2.62), we define the end-point corrected rule $T^n_{\alpha\beta^m}$ by the formula

$$T^n_{\alpha\beta^m}(f) = T^n_{R\beta^m}(f) + h \sum_{j=-k, j\neq 0}^{k} \alpha_j f(x_j), \tag{2.68}$$

with $h = b/(n-1)$, $x_j = jh$.

We will use the expression $T^n_{\alpha\beta^m}$ with appropriately chosen α as quadrature formulae for functions of the form (2.67), and the following construction provides a tool for finding α once $\beta^m = (\beta^m_1, \beta^m_2, ..., \beta^m_{\frac{m-1}{2}})$ is given, so that the rule is of order k, i.e., there exists a $c > 0$ such that

$$| T^n_{\alpha\beta^m}(f) - \int_0^b f(x)dx | < \frac{c}{n^k}. \tag{2.69}$$

For a pair of natural numbers k, m, with $k \geq 1$ and $m \geq 3$ and odd, we will consider the following system of linear algebraic equations with respect to the unknowns α^n_j, with $j = 0, \pm 1, \pm 2, ..., \pm k$:

$$\sum_{j=-k, j\neq 0}^{k} x_j^{i-1} \alpha^n_j = \frac{1}{h} \int_0^b x_j^{i-1} dx - T^n_{R\beta^m}(x^{i-1}), \tag{2.70}$$

for $i = 1, 2, ..., k$, and

$$\sum_{j=-k, j\neq 0}^{k} x_j^{i-k-1} s(x_j) \alpha^n_j = \frac{1}{h} \int_0^b x_j^{i-k-1} s(x) dx - T^n_{R\beta^m}(x^{i-k-1} s(x)), \tag{2.71}$$

for $i = k+1, k+2, ..., 2k$, with $h = b/(n-1)$, $x_j = jh$ and $T^n_{R\beta^m}$ defined by (2.62). We denote the matrix of the system (2.70), (2.71) by A^{nk}_s, its right-hand side by Y^{nk}_s and its solution by $\alpha_n = (\alpha^n_{-k}, \alpha^n_{-(k-1)}, ..., \alpha^n_{-1}, \alpha^n_1, ..., \alpha^n_k)$. The use of expressions $T^n_{\alpha^n\beta^m}$ as quadrature formulae for functions of the form (2.67) is based on the following theorem.

Theorem 2.15 *Suppose that a function $s : (-kh, b+mh] \to R^1$ is such that $s \in c^k(-kh, b+mh]$ and s^k is monotonic on either side of 0. Suppose further that the systems (2.70), (2.71) have solutions $(\alpha^n_{-k}, \alpha^n_{-(k-1)}, \alpha^n_{-1}, \alpha^n_1, ..., \alpha^n_k)$ for all sufficiently large n, and that the sums*

$$\sum_{j=-k, j\neq 0}^{k} (\alpha^n_j)^2 \tag{2.72}$$

are bounded uniformly with respect to n. Finally, suppose that the function $f : (-kh, b + mh] \to R^1$ is defined by (2.67). Then, there exists a real $c > 0$ such that

$$\mid T^n_{\alpha^n\beta^m}(f) - \int_a^b f(x)dx \mid < \frac{c}{n^k} \tag{2.73}$$

for all sufficiently large n.

Proof. Applying the Taylor expansion to the function f at $x = 0$ we obtain

$$f(x) = P(f)(x) + R_k(\phi)(x)s(x) + R_k(\psi), \tag{2.74}$$

where

$$P(f)(x) = s(x)\sum_{i=0}^{k} \frac{\phi^{(i)}(0)}{(i!)}x^i + \sum_{i=0}^{k} \frac{\psi^{(i)}(0)}{i!}x^i, \tag{2.75}$$

and $R_k(\phi), R_k(\psi)$ are such functions $[-kh, b + mh] \to R^1$ that

$$R_k'(\phi)(0) = R_k''(\phi)(0) = \dots = R_k^{(k)}(\phi)(0) = 0, \tag{2.76}$$

$$R_k'(\psi)(0) = R_k''(\psi)(0) = \dots = R_k^{(k)}(\psi)(0) = 0. \tag{2.77}$$

Substituting (2.74) into (2.73), we obtain

$$\begin{aligned} \mid T^n_{\alpha^n\beta_m}(f) - \int_0^b f(x)dx \mid \quad &\leq \quad \mid T^n_{\alpha^n\beta_m}(P(f)) - \int_0^1 P(f)(x)dx \mid + \\ \mid T^n_{\alpha^n\beta_m}((R_k(\phi)\cdot s) + R_k(\psi)) - &\int_0^b ((R_k(\phi(x))s(x))(x) + R_k(\psi(x)))dx \mid . \end{aligned} \tag{2.78}$$

Due to (2.70), (2.71)

$$T^n_{\alpha^n\beta_m}(P(f)) - \int_0^1 P(f)(x)dx = 0, \tag{2.79}$$

and we have

$$\begin{aligned} \mid T^n_{\alpha^n\beta_m}(f) - \int_0^b f(x)dx \mid \leq & \\ & \mid RT^n_{\beta_m k}(s\cdot R_k(\phi)) - \textstyle\int_0^b (s\cdot R_k(\phi))(x)dx \mid \\ + & \quad \mid RT^n_{\beta_m k}(R_k(\psi)) - \textstyle\int_0^b (R_k(\psi))(x)dx \mid \\ + & \quad \mid \textstyle\sum_{j=1}^{2k}(R_k(\phi)(jh)s(jh)\alpha_j^n) + (R_k(\psi)(jh)\alpha_j^n) \mid . \end{aligned} \tag{2.80}$$

Due to (2.77) and (2.64), there exists $c_1 > 0$ such that

$$\mid RT^n_{\beta_m k}(R_k(\psi)) - \int_0^b (R_k(\psi))(x)dx \mid < \frac{c_1}{n^k}. \tag{2.81}$$

Combining (2.76), (2.64), and Lemma 2.12 we conclude that for some $c_2 > 0$

$$| RT^n_{\beta_m k}(s \cdot R_k(\phi)) - \int_0^b (s \cdot R_k(\phi))(x)dx |< \frac{c_2}{n^k}. \tag{2.82}$$

Finally, combining (2.76), (2.77) and Lemma 2.12 we conclude that for some $c_3 > 0$,

$$| \sum_{j=-k, j\neq 0}^{k} (R_k(\phi)(jh)s(jh)\alpha_j^n) + (R_k(\psi)(jh)\alpha_j^n) |< \frac{c_3}{n^k}. \tag{2.83}$$

Now, the conclusion of the theorem follows from the combination of (2.81), (2.82), and (2.83). □

2.4.1 Convergence Rates for Singularites of the forms $|x|^\lambda$ and $log(|x|)$

For the remainder of the chapter, $\phi_1, \phi_2, ..., \phi_{2k}$ will denote functions $(-kh, b + mh] \to R^1$ defined by the formulae

$$\phi_i(x) = x^{i-1}, \tag{2.84}$$

for $i = 1, 2, ..., k$, and

$$\phi_i(x) = x^{i-k-1} s(x), \tag{2.85}$$

for $i = k+1, k+2, ..., 2k$. The following lemma is a particular case of a well-known general fact proven, for example, in [20].

Lemma 2.16 *If $s(x) = x^\lambda$ with λ a real number such that $0 < |\lambda| < 1$, then the functions $\phi_1, \phi_2, ..., \phi_{2k}$ constitute a Chebyshev system on the interval $(-kh, b + mh]$ (i.e., the determinant of the $2k \times 2k$ matrix B_{ij} defined by the formula $B_{ij} = \phi_i(t_j)$ is non-zero for any $2k$ distinct points on the interval $(-kh, b + mh]$).*

Theorem 2.17 *If $s(x) = |x|^\lambda$ with $0 < |\lambda| < 1$, then the convergence rate of the quadrature rule $T^n_{\alpha^n \beta_m}$ is at least k.*

Proof. It immediately follows from Lemma 2.16 that the matrix of the system (2.70), (2.71) is non-singular. We rescale the system (2.70), (2.71) by multiplying its ith equation by $\frac{1}{h^{i-1}}$, for $i = 1, 2, ..., k$, and by $\frac{1}{h^{i-1-k+\lambda}}$, for $i = k+1, k+2, ..., 2k$, obtaining the system of equations

$$\sum_{j=-k, j\neq 0}^{k} j^{i-1}\alpha_k^n = \frac{1}{h^i}\left(\int_0^b x^{i-1}dx - T_{R\beta^m}^n(x^{i-1})\right), \tag{2.86}$$

$i = 1, 2, ..., k$, and

$$\sum_{j=-k, j\neq 0}^{k} j^{i-k-1+\lambda}\alpha_k^n = \frac{1}{h^{i-k+\lambda}}\left(\int_0^b x^{i-k-1+\lambda}dx - T_{R\beta^m}^n(x^{i-k-1+\lambda})\right), \tag{2.87}$$

for $i = k+1, k+2, ..., 2k$.

We will denote the matrix of the system (2.86), (2.87) by B_k, and its right hand side by Z_k^n. Obviously, B_k is independent of n, and using Theorem 2.14 we observe that if $m > k$ then $|Z_k^n|$ is bounded uniformly with respect to n. Now, due to Theorem 2.15, the convergence rate of $T_{\alpha^n\beta^m}^n$ is at least k. □

The proof of Theorem 2.17 can be repeated almost verbatim with $s(x) = log(|x|)$, instead of $s(x) = |x|^\lambda$, resulting in the following theorem.

Theorem 2.18 *If $s(x) = log(|x|)$ then the convergence rate of the quadrature rule $T_{\alpha^n\beta_m}^n$ is at least k.*

2.4.2 Asymptotic behaviour of correction coefficients as $n \to \infty$

An obvious drawback of the expressions $T_{\alpha\beta^m}^n$ as practical quadrature rules is the fact that the weights $\alpha^n = (\alpha_{-k}^n, ..., \alpha_{-1}^n, \alpha_1^n, ..., \alpha_k^n)$ have to be determined for each value of n by solving a system of linear algebraic equations. For singularities of the form $s(x) = log(|x|)$, $s(x) = |x|^\lambda$ we eliminate this problem by constructing a new set of quadrature weights $\gamma = (\gamma_{-k}, \gamma_{-(k-1)}, ..., \gamma_{-1}, \gamma_1, ..., \gamma_k)$, independent of n, and such that the quadrature rules $T_{\gamma^k\beta^m}^n$ are still of order not less than k.

Lemma 2.19 *Suppose that $\beta = (\beta_1^m, \beta_2^m, ..., \beta_{\frac{m-1}{2}}^m)$ is such that the right-hand order of the quadrature formula $T_{R\beta^m}^n$ is m. Further, let $z > 0$ be some real number. Then for any integers p, q such that $p < q$,*

$$\left| \frac{1}{h_p^{z+1}}\left(T_{R\beta^m}^p(x^z) - \int_0^b x^z dx\right) - \frac{1}{h_q^{z+1}}\left(T_{R\beta^m}^q(x^z) - \int_0^b x^z dx\right) \right| = O(h_p^{m-z-1}), \tag{2.88}$$

where $h_p = b/(p-1)$, *and* $h_q = b/(q-1)$.

Proof. Due to Theorem 2.14, there exist real $c_1, c_2 > 0$ such that

$$(T^p_{R\beta^m}(x^z) - \int_0^b x^z dx) = c_1 h_p^m - h_p \sum_{j=-k}^{k} (jh_p)^z, \tag{2.89}$$

and

$$(T^q_{R\beta^m}(x^z) - \int_0^b x^z dx) = c_2 h_q^m - h_q \sum_{j=-k}^{k} (jh_q)^z. \tag{2.90}$$

Now, combining (2.89), (2.90) we obtain

$$\begin{aligned}
&\mid \tfrac{1}{h_p^{z+1}}(T^p_{R\beta^m}(x^z) - \textstyle\int_0^b x^z dx) - \tfrac{1}{h_q^{z+1}}(T^q_{R\beta^m}(x^z) - \textstyle\int_0^b x^z dx) \mid \\
= \;& \frac{1}{h_p^{z+1}}(c_1 h_p^m - h_p \sum_{j=-k}^{k} (jh_p)^z) - \frac{1}{h_q^{z+1}}(c_2 h_q^m - h_q \sum_{j=-k}^{k} (jh_q)^z) \\
= \;& (c_1 h_p^{m-z-1} - \sum_{j=-k}^{k} (j)^z) - (c_2 h_q^{m-z-1} - \sum_{j=-k}^{k} (j)^z) \\
= \;& c_1 h_p^{m-z-1} - c_2 h_q^{m-z-1} \\
= \;& O(h_p^{m-z-1}).
\end{aligned} \tag{2.91}$$

□

Theorem 2.20 *Suppose that* k, m *are two natural numbers such that* $k \leq m-1$ *and that* $\beta = (\beta_1^m, \beta_2^m, ..., \beta^m_{\frac{m-1}{2}})$ *is such that the right-hand order of the quadrature* $T_{R\beta^m}$ *is* m. *Suppose further that* $s(x) = |x|^\lambda$ *with* $0 \leq\mid \lambda \mid\leq 1$, *and that the coefficients* $(\alpha^n_{-k}, \alpha^n_{-(k-1)}, \alpha^n_{-1}, \alpha^n_1, ..., \alpha^n_k)$ *are the solutions of the system (2.70),(2.71).Then*

1) There exists a limit

$$lim_{n\to\infty} \alpha_i^n = \gamma_i, \tag{2.92}$$

for each $i = 1, 2, ..., 2k$.
2) For all $i = 1, 2, ..., 2k$,

$$\mid \alpha_i^n - \gamma_i \mid = O(\frac{1}{n^{m-k}}). \tag{2.93}$$

3) γ_i *do not depend on* m*, as long as* $m \geq k+1$*.*

4) The quadrature formulae $T^n_{\gamma\beta^m}$ *are of order at least* k*.*

Proof. Suppose that p, q are two natural numbers, and $p < q$. Obviously,

$$\begin{aligned} \alpha^p &= (B_k)^{-1} Z^p_k, \\ \alpha^q &= (B_k)^{-1} Z^q_k, \\ \alpha^p - \alpha^q &= (B_k)^{-1}(Z^p_k - Z^q_k). \end{aligned} \tag{2.94}$$

Due to Lemma 2.19, there exists $c > 0$ such that

$$||Z^p_k - Z^q_k|| < \frac{c}{p^{m-k}}. \tag{2.95}$$

and by combining (2.94), (2.95), we see that for some $d > 0$

$$||\alpha^p - \alpha^q|| < \frac{d}{p^{m-k}}. \tag{2.96}$$

Since the weights α^n constitute a Cauchy sequence, they converge to some limit $\gamma = (\gamma_{-k}, \gamma_{-(k-1)}, ..., \gamma_{-1}, \gamma_1, ..., \gamma_k)$, which proves 1, and 2, 3, 4 follow easily. □

The proof of the following theorem is a repetition, almost verbatim, of the proofs of the Lemma 2.19 and Theorem 2.20

Theorem 2.21 *If under the conditions of Theorem 2.20 we replace* $s(x) = |x|^\lambda$ *with* $s(x) = log(|x|)$*, conclusions 1-4 remain correct.*

For singularities of the form $|x|^\lambda$ and $log(|x|)$, Theorem 2.20 and 2.21 reduce the quadratures $T^n_{\alpha\beta^m}$ to the more "conventional" form

$$\int_0^b f(x)dx \approx T^n_{\gamma^k\beta^m}(f) = T^n_{R\beta^m}(f) + h \sum_{j=-k, j\neq 0}^{k} \gamma_j f(x_j). \tag{2.97}$$

Remark 2.4 The whole theory in sections 4.1-4.2 has been constructed for functions with a singularity at the left end of the interval. Obviously, an identical theory holds for functions with a singularity at the right end of the interval. However, in all formulae the expression $T^n_{R\beta^m}$ has to be replaced with $T^n_{L\beta^m}$ (see (2.62), (2.65)).

2.4.3 Central Corrections for Singular Functions

In this section, we will be considering functions $f[-b-mh,0)\cup(0,b+mh]\rightarrow R^1$ of the form

$$f(x)=\phi(x)s(x)+\psi(x), \tag{2.98}$$

with $\phi,\psi\in c^l[-b-mh,b+mh]$, and $s\in c[-b-mh,0)\cup(0,b+mh]$ an integrable function with a singularity at 0. We will define the central-point corrected trapezoidal rule

$$T^n_{\mu^n\beta^m}(f)=T^n_{R\beta^m}(f)+T^n_{L\beta^m}(f)+h\sum_{j=1}^{l}\mu^n_j(f(x_j)+f(x_{-j})), \tag{2.99}$$

with h, x_j defined by(2.63), β^m_i defined by (2.52), $T^n_{R\beta^m}$, $T^n_{L\beta^m}$ defined by (2.62) and (2.65) respectively, and $\mu^n=(\mu^n_1,\mu^n_2,...,\mu^n_l)$ an arbitrary sequence of length l.

We will use the expression $T^n_{\mu^n\beta^m}$ with appropriately chosen μ^n as quadrature formulae for functions of the form (2.98), and the following construction provides a tool for finding μ^n once β^m is given, so that the rule is of order $2l$, i.e., there exists some $c>0$ such that

$$\mid T^n_{\mu^n\beta^m}(f)-\int_{-b}^{b}f(x)dx\mid<\frac{c}{n^{2l}}. \tag{2.100}$$

For a pair of natural numbers l, m, we will consider the following system of linear algebraic equations with respect to the unknowns μ^n_j:

$$\sum_{j=1}^{l}x_j^{2i-2}\mu^n_j=\int_{-b}^{b}x^{2i-2}dx-T^n_{R\beta^m}(x^{2i-2})-T^n_{L\beta^m}(x^{2i-2}), \tag{2.101}$$

for $i=1,2,...,l$, and

$$\sum_{j=1}^{l}x_j^{2i-2-2l}s(x_j)\mu^n_j=\int_{-b}^{b}x^{2i-2-2l}s(x)dx-T^n_{R\beta^m}(x^{2i-2-2l}s(x))-T^n_{L\beta^m}(x^{2i-2-2l}s(x)), \tag{2.102}$$

for $i=l+1,l+2,...,2l$, with $h=b/(n-1)$, $x_j=jh$.

The proofs of Theorem 2.22, 2.23, and 2.24 are almost identical to those of Theorems 2.15, 2.17, and 2.20 respectively, and are thus stated below without proof.

Theorem 2.22 *Suppose that a function $s:[-b-mh,0)\cup(0,b+mh]\rightarrow R^1$ is such that $s\in c^l[-b-mh,0)\cup(0,b+mh]$ and s^l is monotonic on either side of* 0. *Suppose further that* *the systems (2.101), (2.102) have solutions* $(\mu^n_{-l}, \mu^n_{-(l-1)}, \mu^n_{-1}, \mu^n_1, ..., \mu^n_l)$ *for all sufficiently large n, and that the sums*

$$\sum_{j=-l, j\neq 0}^{l} (\mu^n_l)^2 \tag{2.103}$$

are bounded uniformly with respect to n. Finally, suppose that the function $f : [-b-mh, 0) \cup (0, b+mh] \to R^1$ *is defined by (2.98). Then, there exists such* $c > 0$ *that*

$$| T^n_{\mu^n \beta^m}(f) - \int_a^b f(x)dx | < \frac{c}{n^{2l}} \tag{2.104}$$

for all sufficiently large n.

Theorem 2.23 *If* $s(x) = |x|^\lambda$ *with* $0 < |\lambda| < 1$*, or* $s(x) = log(|x|)$*, then the convergence rate of the quadrature rule* $T^n_{\mu^n \beta_m}$ *is at least 2l.*

Theorem 2.24 *Suppose that k, m are two natural numbers such that* $k \leq m-1$ *and that* $\beta = (\beta^m_1, \beta^m_2, ..., \beta^m_{\frac{m-1}{2}})$ *is such that the right-end order of the quadrature* $T_{R\beta^m}$ *is m, and the left-end order of the quadrature* $T_{L\beta^m}$ *is m. Suppose further that* $s(x) = |x|^\lambda$, $0 \leq |\lambda| \leq 1$*, or* $s(x) = log(|x|)$*, and that the coefficients* $(\mu^n_{-k}, \mu^n_{-(k-1)}, \mu^n_{-1}, \mu^n_1, ..., \mu^n_k)$ *are the solutions of the system (2.70),(2.71).Then*

1) There exists a limit

$$lim_{n\to\infty}\mu^n_i = \mu_i, \tag{2.105}$$

for each $i = 1, 2, ..., 2k$.

2) For all $i = 1, 2, ..., 2k$,

$$| \mu^n_i - \mu_i | = O(\frac{1}{n^{m-l}}). \tag{2.106}$$

3) μ_i *do not depend on m, as long as* $m \geq l+1$.

4) The quadrature formulae $T^n_{\mu\beta^m}$ *are of order at least 2l.*

For singularities of the form $|x|^\lambda$ and $log(|x|)$, the Theorem 2.24 reduces the quadrature to the more "conventional" form

$$\int_{-b}^{b} f(x)dx \approx T^n_{\mu\beta^m}(f) = T^n_{R\beta^m}(f) + T^n_{L\beta^m}(f) + \sum_{j=1}^{l} \mu_j(f(x_j) + f(x_{-j})). \tag{2.107}$$

2.4.4 Central Corrections for Singular Functions $f(x) = \phi(x)s(x)$

In this section we construct a quadrature formula specifically for the purpose of approximating definite integrals of functions of the form

$$f(x) = \phi(x)s(x), \tag{2.108}$$

where $\phi(x) : c^p[-b-mh, b+mh] \rightarrow R^1$, and $s \in c[-b-mh, 0) \cup (0, b+mh]$ an integrable function with a singularity at 0. For a finite sequence $\rho = (\rho_0, \rho_1, \rho_2, ..., \rho_p)$, and $T_{R\beta^m}$, $T_{L\beta^m}$ defined in (2.62) and (2.65) respectively, we define the corrected trapezoidal rule $T^n_{\rho^n\beta^m}$ by the formula

$$T^n_{\rho^n\beta^m}(f) = T^n_{R\beta^m}(f) + T^n_{L\beta^m}(f) + h\sum_{j=0}^{p} \rho^n_j(\phi(jh) + \phi(-jh))). \tag{2.109}$$

For integers n, m, p where $n \geq 2$, $p \geq 1$, $m \geq 3$ and odd, we will consider the following system of equations with respect to the unknowns $\rho^n = (\rho^n_0, \rho^n_1, \rho^n_2, ..., \rho^n_p)$:

$$\sum_{j=0}^{p} x_j^{2i-2}\rho^n_j = \frac{1}{h}\int_{-b}^{b}(x^{2i-2}s(x))dx - T^n_{R\beta^m}(x^{2i-2}s(x)) - T^n_{L\beta^m}(x^{2i-2}s(x)), \tag{2.110}$$

where, $h = b/(n-1)$, $x_j = jh$, and $i = 1, 2, ..., p+1$.

The proof of the following theorem is almost identical to the proof of Theorem 2.15.

Theorem 2.25 *Suppose that $n > 2$ is an integer, and h, x_i are defined by (2.63). Further, suppose that $f(x) = \phi(x)s(x)$ where $\phi : [-b-mh, b+mh] \rightarrow R^1$, and $s \in c[-b-mh, 0) \cup (0, b+mh]$ is an integrable function with a singularity at 0. Finally, suppose that the system of equations (2.110) has a solution $(\rho^n_0, \rho^n_1, ..., \rho^n_p)$ for any sufficiently large n and that the sums $\sum_{j=0}^{p}(\rho^n_j)^2)$ are bounded uniformly with respect to n. Then there exists a real $c > 0$ such that*

$$| T^n_{\rho^k h}(f) - \int_{-b}^{b} f(x)dx |< \frac{c}{n^{2p}}. \tag{2.111}$$

The proof of the following theorem is almost identical to that of Theorem 2.20, and is omitted.

Theorem 2.26 *Suppose that $s(x) = log(|x|)$. Then for all $n \geq 2p$, the system (2.110) has a solution $\rho^n = (\rho^n_0, \rho^n_1, \rho^n_2, ..., \rho^n_p)$, and*

$$\rho_0 = \frac{1}{h}\int_{-b}^{b} log(|x|)dx - T^n_{R\beta^m}log(|x|) - T^n_{L\beta^m}log(|x|) - \sum_{j=1}^{p}\rho_j. \tag{2.112}$$

Furthermore, there exist such real numbers $\rho_1, \rho_2, ..., \rho_p$ *and a real* $d > 0$ *such that*

$$lim_{n\to\infty}\rho_j^n = \rho_j, \tag{2.113}$$

and

$$\mid \rho_j^n - \rho_j \mid < d \cdot h^{m-p} \tag{2.114}$$

for all $j = 1, 2, ..., p$. *Finally, there exists a real* c_0 *such that*

$$\mid \rho_0^n - (c_0 + 0.5log(h) - \sum_{j=1}^{p} \rho_j \mid < d \cdot h^{m-p} \tag{2.115}$$

for all $n \geq 2p$.

Remark 2.5 Formulae (2.114), (2.115) indicate that for sufficiently large m, the convergence of $\rho_1^n, \rho_2^n, ..., \rho_p^n$ to $\rho_1, \rho_2, ..., \rho_p$ is virtually instantaneous, and that (2.115) is a nearly perfect approximation to ρ_0^n. The numerical values of $\rho_1, \rho_2, ..., \rho_p$ can be found for various values of p in Section 5.1.4. Also, note that c_0 does not depend on p, and its numerical value (to 16 digits) is $-.9189385332046727$.

The proof of the following theorem is similar to the proof of Theorem 2.20.

Theorem 2.27 *Suppose that* $s(x) = |x|^\lambda$, *with* λ *a real number such that* $0 < |\lambda| < 1$, *and (2.110) has a solution* $\rho^n = (\rho_0^n, \rho_1^n, \rho_2^n, ..., \rho_p^n)$. *Then for all* $n > 2p$, *the quadrature weights* $\rho_0^n, \rho_1^n, \rho_2^n, ..., \rho_p^n$ *are independent of* n.

2.4.5 Corrected trapezoidal rules for other singularities

In the preceding sections quadrature formulae are provided for singular functions of the form

$$f(x) = \phi(x)s(x) + \psi(x), \tag{2.116}$$

and

$$f(x) = \phi(x)s(x), \tag{2.117}$$

where the singularity $s(x)$ is of the form $log(|x|)$, or x^λ $(0 < |\lambda| < 1)$. Obviously the procedure developed in the preceding sections can be applied to other singularities. As an example, we construct a quadrature formula to approximate the definite integral,

$$\int_{-a}^{a} f(x)dx, \tag{2.118}$$

where f is of the form (2.117),

$$s(x) = \frac{1}{\sqrt{a^2 - x^2}}, \tag{2.119}$$

with $a > 0$, and $\phi(x) \in c^k[-a - kh, a + kh]$ and even (i.e., $\phi(-x) = \phi(x)$).

Remark 2.6 The choice of the singularity (2.119) is dictated by the frequency with which it is encountered in the numerical solution of partial differential equations, in signal processing, and other areas. Otherwise, almost any integrable, monotone singularity could have been chosen.

We define the corrected trapezoidal rule $T^n_{\nu^n}$ by the formula

$$T^n_{\nu^n}(f) = \sum_{j=-(n-2)}^{n-2} f(x_j) + h \sum_{i=1}^{k} \nu_i^n f(y_i), \tag{2.120}$$

where $h = a/(n-1)$, $x_j = jh$, $y_i = a - hi$ for $1 \leq i \leq k/2$, and $y_i = a + h(i - k/2)$ for $k/2+1 \leq i \leq k$. We will use the expression $T^n_{\nu^n}$ with appropriately chosen ν^n as quadrature formulae for functions of the form (2.117), and the following construction provides a tool for finding ν^n, so that the rule is of order $2k-2$, i.e., there exists a real $c > 0$ such that

$$\mid T^n_{\nu^n}(f) - \int_{-a}^{a} f(x)dx \mid < \frac{c}{n^{2k-2}}. \tag{2.121}$$

For an even integer $k \geq 2$, we will consider the following system of linear algebraic equations with respect to the unknowns ν_j^n, with $j = 1, 2, ..., k$:

$$\sum_{j=1}^{k} \frac{y_j^{2(i-1)}}{\sqrt{a^2 - x^2}} \nu_j^n = \int_{-a}^{a} \frac{x^{2(i-1)}}{\sqrt{a^2 - x^2}} dx - \sum_{l=-(n-2)}^{n-2} \left(\frac{x_l^{2(i-1)}}{\sqrt{a^2 - x_l^2}}\right), \tag{2.122}$$

with $h = \frac{a}{n-1}$, $x_j = jh$, $y_j = a - hj$ for all $1 \leq j \leq k/2$, and $y_j = a + h(j - k/2)$ for all $k/2+1 \leq j \leq k$. It is easy to see that the linear system (2.122) is independent of the length of the interval a, and the unknowns $\nu_1^n, \nu_2^n, ..., \nu_k^n$ can be determined by solving the system of equations

$$\sum_{j=1}^{k} \frac{y_j^{2(i-1)}}{\sqrt{1 - x^2}} \nu_j^n = \int_{-1}^{1} \frac{x^{2(i-1)}}{\sqrt{1 - x^2}} dx - \sum_{l=-(n-2)}^{n-2} \left(\frac{x_l^{2(i-1)}}{1 - x_l^2}\right), \tag{2.123}$$

with $h = \frac{1}{n-1}$, $x_j = jh$, $y_j = 1 - hj$ for all $1 \le j \le k/2$, and $y_j = 1 + h(j - k/2)$ for all $k/2 + 1 \le j \le k$.

The proof of the following theorem is quite similar to the proof of Theorem 2.15, and is omitted.

Theorem 2.28 *Suppose that for some $a > 0$, $f(x) = \frac{\phi(x)}{\sqrt{(a^2-x^2)}}$ with $\phi \in c^k[-a-kh, a+kh]$. Then there exists such $c > 0$ that*

$$| T^n_{\nu^n}(f) - \int_{-a}^{a} f(x)dx |< \frac{c}{n^{2k-2}} \tag{2.124}$$

for all sufficiently large n.

The authors have been unable to construct a quadrature rule for singularities of the form (2.119), which is independent of the number n of points used in the uncorrected trapezoidal rule. However, this is a relatively minor deficiency since the weights in such cases can be precomputed and stored.

2.5 Numerical Results

Algorithms have been implemented for the construction of the quadratures $T^n_{\beta^m}$, $T^n_{\gamma^k\beta^m}$, $T^n_{\mu^k\beta^m}$, $T^n_{\rho^k\beta^m}$, and $T^n_{\nu^n}$.

The correction coefficients β^m are calculated using (2.25), and (2.52). In the tables in Section 5.1.1 the correction coefficients for orders of convergence upto 43 are tabulated. In Table 1, convergence results are presented for some of the rules $T^n_{\beta^m}$. Column 1 of this table contains the number of nodes discretizing the interval $[0, 1]$ was discretized. In column 2 are the relative errors of the standard 1-sided 4th order corrected trapezoidal rule, given here for comparison. Columns 3-9 contain the relative errors for the rule $T^n_{\beta^m}$ for various orders of convergence m. In all cases the integrand was of the form

$$f(x) = \sin(200x) + \cos(201x). \tag{2.125}$$

The quadrature weights for the rules $T^n_{\gamma^k\beta^m}$, $T^n_{\mu^k\beta^m}$, $T^n_{\rho^k\beta^m}$, and $T^n_{\nu^n}$ are all obtained as solutions of linear systems, and it is easy to see that the linear systems used for determining these weights (see, for example (2.70), (2.71)) are very ill-conditioned. In order to combat the high condition number, all systems were solved using the mathematical package MAPLE using 200 significant digits.

In order to evaluate the coefficients γ for singularities of the form $s(x) = |x|^\lambda$ or $s(x) = log(|x|)$, we start with the right-end corrected trapezoidal rule $T^n_{\beta m}$ of order 40. Under these conditions,

$$| \alpha_i^n - \gamma_i |< O(\frac{1}{n^{40-k}}) \tag{2.126}$$

for all $-k \leq i \leq k, k \neq 0$ (see Theorem 2.20) and for reasonable k, the convergence of α_i^n to γ_i is almost instantaneous. The construction of the quadrature weights μ_i is performed in a similar manner. In Section 5.1.2 the coefficients γ_i are listed for the singularities $log(|x|)$, $|x|^{\frac{1}{2}}$, $|x|^{-\frac{1}{2}}$, $|x|^{\frac{1}{3}}$, $|x|^{-\frac{1}{3}}$, $|x|^{-\frac{1}{9}}$ and for the same singularities, the quadrature weights μ_i are listed in Section 5.1.3. In Table 2, convergence results are presented for some of the rules $T^n_{\gamma^k\beta m}$ for various singularities. Column 1 of this table contains the number of nodes in the discretization of the interval $[0,1]$. In Table 3, convergence results are presented for some of quadrature rules $T^n_{\mu^k\beta m}$ for various singularities. Column 1 of this table contains the number of nodes in the discretization of the interval $[-1,1]$. In all cases the integrand was of the form

$$f(x) = (\sin(20x) + \cos(21x)) + (\sin(23x) + \cos(22x))s(x), \tag{2.127}$$

and the order of convergence used was 10.

Finally, algorithms have been implemented for evaluating quadratures $T^n_{\rho^p h}$, to integrate functions of the form

$$f(x) = \phi(x)log(|x|). \tag{2.128}$$

The quadrature weights are obtained by solving the linear system (2.110). Note that the quadrature weights are independent of the discretization h, except for the first weight ρ_0 which is calculated using the formula (2.115). Presented in Table 4 are convergence results for integrating functions of the form (2.128) where,

$$\phi(x) = \sin(200x) + \cos(201x). \tag{2.129}$$

Column 1 shows the number of nodes in the discretization of the interval $[-1,1]$. Columns 3-6 show the relative errors for the various orders of convergence m as shown.

Table 2.1: Convergence of quadrature rules $T^n_{\beta m}$ for non-singular functions

N	k=4	m=3	m=9	m=15	m=21	m=27	m=33	m=39
20	.230E-01	.112E-01	.131E-01	.136E-01	.138E-01	.138E-01	.138E-01	.138E-01
40	.132E-01	.120E-01	.122E-01	.122E-01	.122E-01	.122E-01	.122E-01	.122E-01
80	.457E-02	.108E-02	.654E-03	.430E-03	.292E-03	.202E-03	.142E-03	.100E-03
160	.216E-03	.804E-04	.223E-05	.743E-07	.264E-08	.972E-10	.365E-11	.139E-12
320	.310E-05	.522E-05	.292E-08	.199E-11	.116E-14	.304E-15	.306E-15	.306E-15
640	.191E-06	.328E-06	.304E-11	.105E-15	.703E-16	.703E-16	.703E-16	.703E-16
1280	.239E-07	.205E-07	.266E-14	.345E-15	.346E-15	.346E-15	.346E-15	.346E-15

Table 2.2: Convergence of quadrature rules $T^n_{\gamma^k \beta m}$ for singular functions (10th order)

N	$log(\|x\|)$	$\|x\|^{\frac{1}{2}}$	$\|x\|^{\frac{-1}{2}}$	$\|x\|^{\frac{1}{3}}$	$\|x\|^{\frac{-1}{3}}$
40	0.29128E-03	0.25056E-04	0.11650E-02	0.42510E-04	0.53715E-03
80	0.72599E-07	0.30493E-07	0.98819E-06	0.53217E-07	0.52449E-06
160	0.56928E-10	0.17499E-10	0.10903E-08	0.32715E-10	0.49582E-09
320	0.65586E-13	0.59119E-14	0.76827E-12	0.12962E-13	0.31491E-12
640	0.18596E-14	0.16376E-14	0.66613E-15	0.17208E-14	0.13878E-14

Table 2.3: Convergence of quadrature rules $T^n_{\mu^k \beta m}$ for singular functions (10th order)

N	$log(\|x\|)$	$\|x\|^{\frac{1}{2}}$	$\|x\|^{\frac{-1}{2}}$	$\|x\|^{\frac{1}{3}}$	$\|x\|^{\frac{-1}{3}}$
40	0.57489E-03	0.49592E-04	0.23137E-02	0.84150E-04	0.10655E-02
80	0.14438E-06	0.60500E-07	0.19680E-05	0.10563E-06	0.10436E-05
160	0.11348E-09	0.34867E-10	0.21762E-08	0.65197E-10	0.98921E-09
320	0.13357E-12	0.13614E-13	0.15360E-11	0.28103E-13	0.62927E-12
640	0.61062E-15	0.16237E-14	0.42188E-14	0.16237E-14	0.50515E-14

Table 2.4: Convergence of the quadrature rule $T^n_{\rho^k \beta m}$ for functions $f(x) = \phi(x)log(|x|)$

N	m=3	m=9	m=15	m=21	m=27	m=33	m=39
40	.546E-01	.536E-01	.536E-01	.536E-01	.536E-01	.536E-01	.536E-01
80	.291E-03	.764E-03	.265E-03	.129E-03	.640E-04	.300E-04	.107E-04
160	.282E-03	.241E-04	.209E-05	.255E-08	.482E-09	.125E-11	.143E-13
320	.437E-04	.190E-04	.912E-06	.392E-09	.162E-09	.294E-12	.147E-14
640	.573E-05	.315E-05	.468E-06	.166E-07	.108E-09	.583E-13	.119E-14

2.6 Generalizations and Conclusions

A group of algorithms has been presented for the construction of high-order corrected trapezoidal rules for functions with various types of singularities, both end-point and in the middle of the interval of integration. In many cases, the corrected rule can have effectively an arbitrarily high order, without the attendent growths of correction weights. The drawback of the approach is the need for the integrand to be available in a small area outside the interval of integration, whenever the singularity being corrected is on one of the ends of that interval.

The algorithm of the chapter admits several straightforward generalizations.

1. There are classes of singularities not covered by this chapter for which some versions of Theorem 2.15 can be fairly easily proven.

2. The quadratures can be easily modified to handle functions of the form

$$f(x) = \psi(x) + \sum_{i=1}^{m} \phi_i(x) \cdot s_i(x), \tag{2.130}$$

where ψ, ϕ_1,ϕ_2,...,ϕ_m are smooth functions, and s_1, s_2, ..., s_m are several different singularities.

3. Quadrature rules developed of this chapter have fairly obvious analogues in two and three dimensions. However, the proofs of the multidimensional versions of the theorems in this chapter are somewhat more involved than those of their one dimesional counterparts. These results will be reported at a later date.

4. High-order corrected trapezoidal rules can be used to approximate integrals

$$\int_0^{\pi} cos(a \cos\theta) d\theta \tag{2.131}$$

by rewriting the integral as

$$\int_{-a}^{a} \frac{cos(x)}{\sqrt{a^2 - x^2}} dx \tag{2.132}$$

and using the quadrature rule T_{ν^n} defined in (2.120). This rule proves to be of fundamental importance in the development of the fast Hankel Transform (see, for example Chapter 4).

5. Integral equations of the form

$$\int_0^{L} \sigma(w) log(|z - w|) ds_w = C \tag{2.133}$$

are encountered in the study of partial differential equations (see, for example [22]). In order to apply the Nystrom algorithm to the integral equation (2.133), the left-hand side is decomposed into a sum

$$\int_0^L \sigma(w)log(|z-w|)ds_w = I(z) + J(z), \tag{2.134}$$

where the integral operators I and J are defined by the formulae

$$I(z) = \int_0^L \sigma(w)log(|\frac{z-w}{\gamma^{-1}(z)-\gamma^{-1}(w)}|)ds_w, \tag{2.135}$$

$$J(z) = \int_0^L \sigma(w)log(|\gamma^{-1}(z)-\gamma^{-1}(w)|)ds_w. \tag{2.136}$$

Now, the integral operator I can be discretized by the uncorrected trapezoidal rule and the operator J can be discretized by the corrected trapezoidal rule $T^n_{\rho h}$ defined in (2.109) to a rapidly convergent finite-dimesional approximation to (2.133).

Chapter 3

Fourier transforms of singular functions

3.1 Introduction

Fourier techniques have been an extremely important and popular analytical tool in mathematics and physics for more than two centuries. The introduction of the fast Fourier transform (FFT) algorithm in the 1960s has greatly broadened the scope of application of the Fourier transform to data handling, and has also brought prominence to the discrete Fourier transform (DFT). The impact of the FFT can be felt in such diverse areas as signal processing, electrical engineering, VLSI circuit modeling, medical imaging.

The DFT is often used for the evaluation of integrals of the form

$$\int_0^{2\pi} f(x)e^{i\omega x}dx, \tag{3.1}$$

for various values of ω. It is well known that the DFT is an extremely effective numerical tool, when the function f is smooth and periodic. However, when the function has a jump discontinuity, or is singular, the accuracy of the DFT is significantly reduced. The reason for the loss of accuracy is that the DFT, which implements the trapezoidal rule, is slowly convergent. In fact, the numerical error that results from the discontinuity is on the order of $\frac{1}{n}$ for a problem of size n. Such errors make even single precision calculations, especially in higher dimensions, prohibitively expensive. Richardson extrapolation alleviates the situation, but double precision calculations are still virtually impossible.

In recent papers (see, for example, [29], [5]) algorithms are described for the rapid, and accurate, computation of Fourier transforms of functions with jump discontinuities. More specifically, these papers present algorithms for the evaluation of the integrals

$$\hat{f}(m,n) = \int_0^1 \int_0^1 f(x,y) e^{-2\pi i m x} e^{-2\pi i m y} dx dy, \tag{3.2}$$

with a given accuracy ϵ for $-M \leq m \leq M$, and $-N \leq n \leq N$, where f is a piecewise constant function, or, more generally, a piecewise smooth function. The approach of [29] is to use Green's theorem to replace area integrals, with line integrals, thereby reducing the number of nodes required for integration. Thereafter, the weights are redistributed on a uniform grid with the help of Lagrange interpolation, bringing the data into a form suitable for the FFT. The approach of [5] is similar; a wavelet based scheme is devised to redistribute integration weights onto a uniform grid. However, both algorithms are very involved, and not as general, as the algorithms developed in this chapter.

In Chapter 2, a group of quadrature formulae is presented applicable to functions with singularities, generalizing the classical end-point corrected trapezoidal quadrature rules. More specifically, corrected trapezoidal quadrature rules are developed to approximate definite integrals of singular functions $f : [a,b] \to \mathbf{R}$ of the form

$$f(x) = \phi(x)s(x) + \psi(x), \tag{3.3}$$

and

$$f(x) = \phi(x)s(x), \tag{3.4}$$

where $\phi(x), \psi(x) \in c^k[a,b]$, and s is an integrable function with a singularity on the interval $[a,b]$. In this chapter, we observe that the quadrature scheme developed in Chapter 2 can be applied, in a straightforward manner, to numerically integrate singular functions of the form $f(x) \cdot e^{iwx}$.

We also develop an alternate scheme to evaluate the Fourier transform of functions of the form (3.3), (3.4) where ϕ, ψ are periodic. This procedure is based on using trigonometric basis functions to approximate the integrand $f(x)$. In this case resulting systems of linear algebraic equations are well-conditioned, and in fact can be solved rapidly by means of the FFT; thus, the correction weights do not need to be precomputed and tabulated.

In the following section, we summarize several facts from approximation theory, classical numerical analysis, and Chapter 2 to be used in subsequent sections. In §3.3 we describe a procedure to evaluate the Fourier transform of singular functions based on the results of Chapter 2. In Section 3.4 we describe an alternate procedure to evaluate the Fourier transform of periodic singular functions, using global corrections. Finally in §3.7 we provide results of several numerical experiments, and in §3.8 we discuss several straightforward generalizations and applications of the algorithms of this chapter.

3.2 Mathematical and numerical preliminaries

In this section, we summarize some mathematical, and numerical results, to be used in the rest of the chapter.

3.2.1 Corrected trapezoidal quadrature rules for singular functions

In Chapter 2, we construct a class of quadrature formulae for end-point singular functions, generalizing the classical end-point corrected trapezoidal rules. In this subsection we summarize some of the results obtained in Chapter 2.

Suppose that $n > 2$ is an integer. Suppose further that $b > 0$ is a real numbers with $h = b/(n-1)$. Finally, suppose that $\{w_{-m}, \ldots, w_m\}$ are appropriately chosen correction coefficients. We define a corrected trapezoidal rule T_n^w by the formula

$$T_n^w(f) = T_n(f) + h \sum_{i=-m}^{m} w_i \cdot f(x_i), \tag{3.5}$$

where $x_i = i \cdot h$, for all $i = -m, \ldots, m$. For a function $f : [-mh, b] \to \mathbf{R}$, we will look upon the sum $T_n^w(f)$ as an approximation to the integral

$$\int_0^b f(x)dx. \tag{3.6}$$

Remark 3.1 Consider a function $f : [-mh, b] \to \mathbf{R}$ which has a singularity at $x_0 = 0$. Suppose that there are $2m + 1$ correction coefficients $\{w_{-m}, \ldots, w_m\}$ surrounding the singularity. Then we say that T_n^m is a locally corrected rule if $m < n$. On the other hand, if there are as many corrections as there are nodes on the interval of integration (i.e., $m = n$), then we say that T_n^m is a globally corrected rule.

The proof of the following theorem is almost identical to the proof of Theorem 2.17 in Chapter 4.

Theorem 3.1 *Suppose that $n > 2$ is an integer. Suppose further that $h, b > 0$ are real numbers with $h = b/(n-1)$, and $f \in c^{2m}[-mh, b]$ is an integrable function. Then there exist real coefficients $\{w_{-m}^n, \ldots, w_m^n\}$ ($|w_i^n| < 1$) such that the rule T_n^w is of order $2m$, i.e., there exists a real number $c > 0$ such that*

$$| T_n^w(f) - \int_0^b f(x)dx |< \frac{c}{n^{2m}}, \tag{3.7}$$

for all $m > 2$.

In Chapter 2, end-point corrected quadrature formulae T_n^w are developed to approximate definite integrals of singular functions $f : [a, b] \to \mathbf{R}$ of the form

$$f(x) = \phi(x)s(x) + \psi(x), \tag{3.8}$$

and

$$f(x) = \phi(x)s(x), \tag{3.9}$$

where $a \leq 0 \leq b$, $\phi(x), \psi(x) \in c^k[a, b]$, and s is an integrable function with a singularity on the interval $[a, b]$.

The actual quadrature weights $\{w_{-m}, \ldots, w_{N+m}\}$ are obtained for each singularity as a solution of a linear algebraic system. For a pair of natural numbers k, m, with $k \geq 1$ and $m \geq 3$ and odd, we will consider the following system of linear algebraic equations with respect to the unknowns w_j, with $j = 0, \pm 1, \pm 2, ..., \pm m$:

$$\sum_{j=-m, j\neq 0}^{m} x_j^{i-1} w_j = \frac{1}{h} \int_0^b x_j^{i-1} dx - T_n(x^{i-1}), \tag{3.10}$$

for $i = 1, 2, ..., m$, and

$$\sum_{j=-m, j\neq 0}^{m} x_j^{i-m-1} s(x_j) w_j = \frac{1}{h} \int_0^b x_j^{i-m-1} s(x) dx - T_n(x^{i-m-1} s(x)), \tag{3.11}$$

for $i = m+1, m+2, ..., 2m$, with $h = b/(n-1)$, $x_j = jh$.

3.2.2 The Fast Fourier transform

The following definition of the discrete fourier transform (DFT) can be found, for example, in [25].

Definition 3.1 *For a complex sequence* $\{f_0, \ldots, f_{N-1}\}$ *the discrete Fourier transform* $\{F_j\}$ *is defined by the formula*

$$F_j = \frac{2\pi}{N} \sum_{k=0}^{N-1} f_k \cdot e^{\frac{i2\pi jk}{N}} \tag{3.12}$$

for all $j = 0, 1, ..., N-1$*, where* $i = \sqrt{-1}$.

Remark 3.2 The FFT algorithm reduces the number of operations for the DFT from $O(N^2)$ to $O(N \log N)$ by a sequence of algebraic manipulations (see, for example, [25]).

Remark 3.3 The DFT is a trapezoidal approximation (see Definition 2.1) to the continuos Fourier transform. More specifically, suppose that $f \in c^2[0, 2\pi]$ is a periodic function with period 2π. Suppose further that $h = \frac{2\pi}{N}$, $f_i = f(ih)$, and F_j is defined in (3.12). Then

$$F_j \approx \int_0^{2\pi} f(x) \cdot e^{ijx} dx, \tag{3.13}$$

for all $j = 0, 1, \ldots, N-1$.

The following corollary is an immediate consequence of Lemma 2.1.

Corollary 3.2 *Suppose that* $f \in c^m[0, 2\pi]$ *is a periodic function with period* 2π*. Suppose further that* $h = \frac{2\pi}{N}$*, and* $f_i = f(ih)$*. Then the DFT is a rule of order* m*, i.e., there exists some real* $c > 0$ *such that*

$$\mid F_j - \int_0^{2\pi} f(x) \cdot e^{ijx} dx \mid < \frac{c}{N^m} \tag{3.14}$$

for all $j = 0, 1, ..., N-1$.

3.3 Application of local corrections to the evaluation of Fourier transforms of functions with singularities

Remark 3.4 It is clear from Corollary 3.2 that, when a function is smooth, and periodic on the interval $[0, 2\pi]$, the DFT provides an an excellent approximation to the continuous

Fourier transform. However, when a function has a jump discontinuity, or is singular the DFT provides a poor approximation to the exact Fourier transform.

In this section we develop a scheme for the accurate, and rapid evaluation of the Fourier transform of functions with singularities. The algorithm is based on a combination of the fast Fourier transform (FFT) with a quadrature scheme tailored to the singularity. More specifically, we observe that the quadrature scheme derived for singular functions (see, Section 3.2.1) can be applied, in a straightforward manner, to the numerical integration of singular functions of the form $f(x) \cdot e^{iwx}$.

The following corollary is an immediate consequence of Theorem 3.1.

Corollary 3.3 *Suppose that $f : [0, 2\pi] \to \mathbf{R}$ is a function of the form*

$$f(x) = \phi(x)s(x) + \psi(x), \tag{3.15}$$

and,

$$f(x) = \phi(x)s(x), \tag{3.16}$$

where $\phi(x), \psi(x) \in c^m[0, 2\pi]$, and $s(x) = x^\alpha$ ($|\alpha| < 1$) or $s(x) = \log(x)$. Consider the rule T_n^w (introduced in (3.5)) approximating integrals of functions of the form (3.16), i.e.,

$$\mid T_n^w(f) - \int_0^{2\pi} f(x)dx \mid < \frac{c}{n^{2m}}. \tag{3.17}$$

Then the same rule T_n^m, also integrates the exact Fourier transform with the order of convergence $2m$, i.e., there exists a real number $d > 0$ such that

$$\mid T_n^w(f \cdot e^{ijx}) - \int_0^{2\pi} f(x) \cdot e^{ijx} dx \mid < \frac{d}{n^{2m}}, \tag{3.18}$$

for all $j = 0, 1, \ldots, N-1$.

3.3.1 The corrected discrete Fourier transform

Definition 3.2 *For a finite real sequence $\{f_{-m}, \ldots f_{N-1}\}$ we define the corrected discrete Fourier transform $\{F_j^C\}$ by the formula*

$$F_j^C = F_j + \frac{2\pi}{N} \sum_{i=-m}^{m} f_i \cdot w_i \cdot e^{\frac{i2\pi jk}{N}}, \tag{3.19}$$

for all $j = 0, 1, ..., N-1$, where $\{F_j\}$ is defined in (3.12), and $\{w_{-m}, \ldots, w_m\}$ are appropriate correction coefficients.

Remark 3.5 It follows from Definition 3.2, and (3.5) that

$$F_j^C = T_n^w(f \cdot e^{ijx}), \tag{3.20}$$

for all $j = 0, \ldots, N-1$. Now, due to Corollary 3.3 the corrected discrete Fourier transform is a rule of order $2m$. That is, there exists a real number $c > 0$ such that

$$\mid F_j^C - \int_0^{2\pi} f(x) \cdot e^{ijx} dx \mid < \frac{c}{n^{2m}}. \tag{3.21}$$

3.3.2 Rapid evaluation of the corrected discrete Fourier transform

It is possible to rapidly evaluate the sums (3.19) by means of the FFT.

More specifically, given the real sequence $\{f_{-m}, \ldots f_{N-1}\}$, we define the sequence $\{\hat{f}_0, \ldots, \hat{f}_{N-1}\}$ by the formulae

$$\hat{f}_i = f_i w_i + f_{-i} w_{-i}, \tag{3.22}$$

for all $i = 0, 1, \ldots, m$, and

$$\hat{f}_i = f_i, \tag{3.23}$$

for all $i = m+1, \ldots, N-1$. Then it immediately follows from the combination of (3.19), (3.22), and (3.23) that

$$F_j^C = \frac{2\pi}{N} \sum_{k=0}^{N-1} \hat{f}_k \cdot e^{\frac{i2\pi jk}{N}} \tag{3.24}$$

Observation 3.6 *The sequence $\{\hat{f}_0, \ldots, \hat{f}_{N-1}\}$ is computed in $O(m)$ operations using (3.22). Subsequently, the sums (3.23) are computed, by means of the FFT, in $O(N \log N)$ operations. Hence, the corrected Fourier transform is computed in $O(N \log N) + O(m)$ operations.*

Remark 3.7 We have observed that the system of linear algebraic equations used to obtain the quadrature weights can be extremely ill-conditioned (see, Remark 1.1). It turns out, that for periodic functions, an alternate procedure may be devised; in this case, the resulting system of linear algebraic equations is well-conditioned. We develop this approach in the following section.

3.4 Application of global corrections to the evaluation of Fourier transforms of functions with singularities

In this section we develop a procedure for the evaluation of the Fourier transform of functions of the form $f(x) = \phi(x) \cdot s(x)$, where ϕ is periodic, and s is an integrable singularity. We devise a quadrature scheme which uses trigonometric basis functions to obtain correction weights. Although, the scheme for deriving quadrature weights is similar to the approach of Chapter 2, the resulting systems of linear algebraic equations are well-conditioned, and in fact can be solved rapidly by means of the FFT. Thus, the correction weights do not need to be precomputed and tabulated. More importantly, since the growth of quadrature weights is suppressed, we can construct schemes of arbitrarily high order.

3.4.1 Integration of periodic functions

Remark 3.8 For the rest of this section we shall assume, without loss of generality that $N > 1$ is an odd integer, $h = \frac{2\pi}{N}$. Further, we shall assume that $x_j = h \cdot j$, for all $j = 0, 1, \ldots, N-1$, are equispaced nodes at which functions are tabulated.

Theorem 3.4 *Suppose s is singular function integrable on $[0, 2\pi]$. Suppose further that $\phi : \mathbf{R}^1 \to \mathbf{C}^1$ is any periodic function of the form*

$$\phi(x) = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_k \cdot e^{i \cdot k \cdot x}, \tag{3.25}$$

where $\{\alpha_k\}$ is a set of real numbers. Then there exists a sequence of real numbers (weights) $\{w_0, w_1, \ldots, w_{N-1}\}$ such that any function $f : [0, 2\pi] \to \mathbf{R}$ of the form

$$f(x) = \phi(x) \cdot s(x) \tag{3.26}$$

is integrated exactly, i.e.,

$$\int_0^{2\pi} f(x)dx = \sum_{j=0}^{N-1} w_j \phi(x_j). \tag{3.27}$$

Proof. Since f, defined in (3.26), is a linear combination of $\{s(x) \cdot e^{-i \cdot \frac{N-1}{2} \cdot x}, \ldots, s(x) \cdot e^{i \cdot \frac{N-1}{2} \cdot x}\}$, it is sufficient to show the existence of weights $\{w_0, w_1, \ldots, w_{N-1}\}$ which integrate each of the above basis functions exactly. That is,

$$\sum_{j=0}^{N-1} e^{i\cdot k\cdot x_j} \cdot w_j = \int_0^{2\pi} e^{i\cdot k\cdot x} \cdot s(x)dx, \tag{3.28}$$

for all $k = -(N-1)/2, \ldots, (N-1)/2$. We observe that the left hand side of (3.28) is simply the DFT of the sequence $\{w_0, w_1, \ldots, w_{N-1}\}$. Thus, the correction weights $\{w_0, w_1, \ldots, w_{N-1}\}$ can be obtained as the inverse DFT of the sequence of the Fourier coefficients of s. More specifically, suppose that

$$\hat{s}_k = \int_0^{2\pi} e^{i\cdot k\cdot x} \cdot s(x)dx, \tag{3.29}$$

for all $k = -(N-1)/2, \ldots, (N-1)/2$. Then

$$w_j = \frac{1}{N}\sum_{k=0}^{N-1} \hat{s}_k \cdot e^{-i\cdot j\cdot x_k}, \tag{3.30}$$

for all $j = 0, 1, \ldots, N-1$. □

Remark 3.9 Suppose that the sequence of Fourier coefficients $\{\hat{s}_k\}$ (see, (3.29)) are computed and stored. Then the weights $\{w_0, w_1, \ldots, w_{N-1}\}$ can be computed in $O(N \log N)$ operations (using an FFT to evaluate the sums in (3.30)).

3.4.2 The Fast fourier transform of singular functions

In this subsection we describe a procedure, based on Theorem 3.4, for the accurate evaluation of Fourier transforms of singular functions. It follows from Theorem 3.4 that the the quadrature weights $\{w_0, w_1, \ldots, w_{N-1}\}$ will exactly integrate functions $f(x) \cdot e^{ikx}$ whose maximum (combined) frequency is $k = \frac{N-1}{2}$. Hence, in order to exactly evaluate the exact Fourier transform of the a function $f(x)$ (defined in (3.26)), twice as many nodes and weights are required (i.e., the function needs to be oversampled by a factor of 2).

More specifically, suppose that functions s, f, and ϕ are as specified in Theorem 3.4. Suppose further that the $2N-1$ quadrature weights $\{w_0, w_1, \ldots, w_{2N-1}\}$ are computed by means of the formula (3.30). Then

$$\int_0^{2\pi} f(x)e^{i\cdot k\cdot x}dx = \sum_{j=0}^{2N-1} w_j \phi_j e^{i\cdot k\cdot x_j} \tag{3.31}$$

for all $k = -(N-1)/2, \ldots, (N-1)/2$.

Remark 3.10 It was observed in Remark 3.9 that the quadrature weights can be computed in $O(N \log N)$ operations. Subsequently, sums of the form (3.31) can be computed in $O(N \log N)$ operations using the FFT (see, Remark 3.2). Thus the corrected Fourier transform, including computation of correction weights, can be computed in $O(N \log N)$ operations

The following corollary is an immediate consequence of Theorem 3.4.

Corollary 3.5 *Suppose that $\{s_1, s_2, \ldots, s_p\}$ are singular functions integrable on $[0, 2\pi]$. Due to Theorem 3.4, we know that there exists a set of quadrature weights $\{w_0^l, \ldots, w_{2N-1}^l\}$ for each singularity $\{s_l\}$, such that the corrected Fourier transform is exact, i.e., for each $l = 1, 2, \ldots, p$*

$$\int_0^{2\pi} f(x) s_l(x) e^{i \cdot k \cdot x} dx = \sum_{j=0}^{2N-1} w_j^l \phi_j e^{i \cdot k \cdot x_j} \tag{3.32}$$

for all $k = -(N-1)/2, \ldots, (N-1)/2$. Suppose further that S is a linear combination of the singularities, i.e.,

$$S(x) = \sum_{l=1}^{p} \beta_l \cdot s_l(x), \tag{3.33}$$

where $\{\beta_1, \ldots, \beta_p\}$ are a sequence of real numbers, and define the real coefficients

$$W_j = \sum_{i=1}^{p} w_j^i \cdot \beta_i \tag{3.34}$$

for all $j = 0, 1, \ldots, 2N-1$. Then the quadrature weights $\{W_0, \ldots, W_{2N-1}\}$ integrate a linear combination of the singularities, i.e.,

$$\int_0^{2\pi} f(x) S(x) e^{i \cdot k \cdot x} dx = \sum_{j=0}^{2N-1} W_j \phi_j e^{i \cdot k \cdot x_j}, \tag{3.35}$$

for all $k = -(N-1)/2, \ldots, (N-1)/2$.

3.5 The FFT for functions with singularities in two-dimensions

All the results of the previous section generalize to the two-dimensional case. The following theorem is the two dimensional analog of Theorem 3.4. The proof is almost identical to the proof of Theorem 3.4.

Theorem 3.6 *Suppose that $s : \mathbf{R}^2 \to \mathbb{C}^1$ is singular function integrable on $[0, 2\pi]X[0, 2\pi]$. Suppose further that $\phi : \mathbf{R}^2 \to \mathbb{C}^1$ is any periodic function of the form*

$$\phi(x, y) = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_{k,m} \cdot e^{i\cdot(k\cdot x + m\cdot y)}, \tag{3.36}$$

where $\{\alpha_{k,m}\}$ is a set of real numbers.

Then there exists a sequence of weights $\{w_{0,0}, w_{0,1}, \ldots, w_{2N-1,2N-1}\}$ such that the Fourier transform of any function $f[0, 2\pi]^2 \to \mathbb{C}^1$ of the form

$$f(x, y) = \phi(x, y) \cdot s(x, y) \tag{3.37}$$

is evaluated exactly, i.e.,

$$\int_0^{2\pi} \int_0^{2\pi} f(x, y) \cdot e^{i(kx+my)} dx dy = \sum_{l=0}^{2N-1} \sum_{j=0}^{2N-1} w_{j,l} \cdot \phi(x_j, y_l) \cdot e^{i\cdot(k\cdot x_j + m\cdot y_l)}, \tag{3.38}$$

for all $k = -(N-1)/2, \ldots, (N-1)/2$, and for all $m = -(N-1)/2, \ldots, (N-1)/2$.

Remark 3.11 When $s(x, y) = s_1(x) \cdot s_2(y)$, the problem can be reduced from a two dimesional problem to a one dimensional one. Clearly,

$$\int_0^{2\pi} \int_0^{2\pi} f_1(x) \cdot f_2(y) \cdot e^{i(kx+my)} dx = (\int_0^{2\pi} f_1(x) \cdot e^{ikx} dx) \cdot (\int_0^{2\pi} f_2(y) \cdot e^{imy}) dy, \tag{3.39}$$

for all $k = -(N-1)/2, \ldots, (N-1)/2$, and for all $m = -(N-1)/2, \ldots, (N-1)/2$. In other words, suppose that the weights $\{w_0^1, w_1^1, \ldots, w_{N-1}^1\}$ exactly integrate $f_1(x)$, and $\{w_0^2, w_1^2, \ldots, w_{N-1}^2\}$ exactly integrate $f_2(y)$. Then the tensor product of these weights $\{w_0^1 w_0^2, w_0^1 w_1^2, \ldots, w_{N-1}^1 w_{N-1}^2\}$ exactly integrate $f(x, y) = f_1(x) \cdot f_2(y)$. Hence 2-D Fourier transforms of singularites which are naturally separable (such as the characteristic function of a rectangle) are reduced to 1-D Fourier transforms.

3.6 Applications of quadratures to the design of VLSI masks

In recent papers (see, for example, [29], [5]) algorithms are described for the rapid, and accurate, computation of Fourier transforms of functions with jump discontinuities. More specifically, these papers present algorithms for the rapid evaluation of the integrals

$$\hat{f}(m, n) = \int_0^{2\pi} \int_0^{2\pi} f(x, y) e^{imx} e^{imy} dx dy, \tag{3.40}$$

where f is a piecewise constant function. In this section we outline a scheme to solve a more general problem. We present an algorithm for the rapid evaluation of the integrals

$$\hat{f}(m,n) = \int_0^{2\pi}\int_0^{2\pi} \phi(x,y)\cdot s(x,y)\cdot e^{imx}e^{imy}dxdy, \tag{3.41}$$

where s is piecewise constant, and ϕ is a periodic function.

Lemma 3.7 *Suppose that ϕ is as defined in (3.25). Suppose further that s_l is a step function*

$$\begin{aligned} s_l(x) &= 1 \qquad \textit{if} \qquad l\cdot h \le x \le (l+1)\cdot h,\\ s_l(x) &= 0 \quad \textit{otherwise}, \end{aligned} \tag{3.42}$$

for all $l = 0,1,\ldots,N-1$. Finally, suppose that the weights $\{w_0^l, w_1^l,\ldots,w_{N-1}^l\}$ are chosen (see, Theorem 3.4) so that each function $f_l(x) = \phi(x)\cdot s_l(x)$ is integrated exactly, i.e.,

$$\int_0^{2\pi} f_l(x)dx = \sum_{j=0}^{N-1} w_j^l\phi(x_j). \tag{3.43}$$

for all $l = 1,2,\ldots,N-1$. Then the weights $\{w_j^l\}$ are expressed by $\{w_j^0\}$ by the following formulae

$$w_j^l = w_{j-l}^0, \tag{3.44}$$

for all $l \le j$, and

$$w_j^l = w_{j-l+N}^0, \tag{3.45}$$

for all $l > j$.

Proof. Due to Theorem 3.4 the weights $\{w_j^l\}$ are given by the formula

$$w_j^l = \frac{1}{N}\sum_{k=0}^{N-1} e^{-i\cdot k\cdot j\cdot h}\int_0^{2\pi} e^{i\cdot k\cdot x}\cdot s_l(x)dx, \tag{3.46}$$

for all $j = 0,1,\ldots,N-1$, and all $l = 0,1,\ldots,N$. Now, for a fixed l, we express the Fourier coefficients of $s_l(x)$ by those of $s_0(x)$, i.e.,

$$\int_0^{2\pi} e^{i\cdot k\cdot x}\cdot s_l(x)dx = \int_0^{2\pi} e^{i\cdot k\cdot x}\cdot s_0(x-l\cdot h)dx = e^{i\cdot k\cdot l\cdot h}\int_0^{2\pi} e^{i\cdot k\cdot x}\cdot s_0(x)dx \tag{3.47}$$

Hence,

$$w_j^l = \frac{1}{N}\sum_{k=0}^{N-1} e^{-i\cdot k\cdot j\cdot h} e^{i\cdot k\cdot l\cdot h} \int_0^{2\pi} e^{i\cdot k\cdot x}\cdot s_0(x)dx = \tag{3.48}$$

$$\frac{1}{N}\sum_{k=0}^{N-1} e^{-i\cdot k\cdot (j-l)\cdot h} \int_0^{2\pi} e^{i\cdot k\cdot x}\cdot s_0(x)dx. \tag{3.49}$$

Or,

$$w_j^l = w_{j-l}^0, \tag{3.50}$$

for all $l \leq j$, and

$$w_j^l = w_{j-l+N}^0, \tag{3.51}$$

for all $l > j$. □

In other words, the set of weights $\{w_0^i, w_1^i, \ldots, w_{N-1}^i\}$, is obtained from the set $\{w_0^0, w_1^0, \ldots, w_{N-1}^0\}$, by a shift in the index.

The proof of the following corollary is an immediate consequence of Lemma 3.7.

Corollary 3.8 *Suppose that $S(x)$ is a linear combination of step functions, i.e.,*

$$S(x) = \sum_{l=0}^{N-1} \beta_l s_l(x). \tag{3.52}$$

We know that (see, Corollary 3.5) the set of weights that integrate the linear combination of singularities, is given by the formula

$$W_j = \sum_{l=0}^{N-1} w_j^l \beta_l \tag{3.53}$$

for all $j = 0, 1, \ldots, N-1$. It follows immediately from the conclusion of Lemma 3.7 that the weights may instead be computed by the following discrete convolution

$$W_j = w_j^i * \beta^i. \tag{3.54}$$

Remark 3.12 It is obvious that both Lemma 3.7, and Corollary 3.8 generalize to the two-dimensional case. In the following subsection, we make use of this fact to construct an algorithm used for the evaluation of the integrals (3.41).

3.6.1 An algorithm applicable to VLSI

The following problem is encountered in the design and study of VLSI circuit design.

Problem: Given a two-dimensional function $f : \mathbf{R}^2 \to \mathbf{R}^1$,

$$f(x,y) = \sum_{j=1}^{N}\sum_{k=1}^{N} \phi(x,y) \cdot \alpha_{j,k} \cdot s_{j,k}(x,y), \tag{3.55}$$

where ϕ is a function periodic $[0, 2\pi]$, $\{\alpha_{i,j}\}$ is any sequence of real numbers, and $s_{j,k}$ represent characteristic functions of squares, i.e.,

$$\begin{aligned} s_{j,k}(x,y) &= 1 \qquad \text{if } x \in [j \cdot h, (j+1) \cdot h], \text{ and, } y \in [k \cdot h, (k+1) \cdot h], \\ s_{j,k}(x,y) &= 0 \qquad \text{otherwise.} \end{aligned} \tag{3.56}$$

Evaluate the following integrals (the Fourier transform),

$$\hat{f}(l,m) = \int_0^{2\pi}\int_0^{2\pi} f(x,y) e^{ilx} e^{imy} dx dy, \tag{3.57}$$

for all $l = -(N-1)/2, \ldots, (N-1)/2$, and $m = -(N-1)/2, \ldots, (N-1)/2$.

An algorithm, based on the results of this section is presented below.

Algorithm 3.1

Step	Complexity	Description
0	$O(N^2)$	**Comment** [Precomputation: Input Problem size N. Compute the Fourier coeficients of the square $s_{0,0}$ analytically.]

$$\hat{s}_{j,k} = \int_0^{2\pi}\int_0^{2\pi} s_{0,0}(x,y) \cdot e^{ijx} \cdot e^{iky} dx dy, \tag{3.58}$$

for $j = -(N-1)/2, \ldots, (N-1)/2$, and $k = -(N-1)/2, \ldots, (N-1)/2$.

Step	Complexity	Description
1	$O(N^2 \log N)$	**Comment** [Initialization: Compute the Fourier transform of the location vector (the coefficients $\alpha_{l,m}$), by means of the FFT.]

$$\hat{\alpha}_{l,m} = \sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_{j,k} \cdot e^{\frac{i2\pi jl}{N}} \cdot e^{\frac{i2\pi km}{N}}, \tag{3.59}$$

for $l = 0, \ldots, N-1$, $m = 0, \ldots, N-1$.

2 $O(N^2)$ **Comment**[Multiply the Fourier coefficients of the location vector, and the Fourier coefficients of the square]

$$\hat{W}_{l,m} = \hat{s}_{l,m} \cdot \hat{\alpha}_{l,m}, \tag{3.60}$$

for $l = 0, \ldots, N-1$, $m = 0, \ldots, N-1$.

3 $O(N^2 \log N)$ **Comment**[Inverse FFT obtaining the sum of the weights associated with all singularities]

$$W_{j,k} = \sum_{l=1}^{N} \sum_{m=1}^{N} \hat{W_{l,m}} \cdot e^{\frac{-i2\pi jl}{N}} \cdot e^{\frac{-i2\pi km}{N}}, \tag{3.61}$$

for $j = 0, \ldots, N-1$, $k = 0, \ldots, N-1$.]

4 $O(N^2)$ **Comment**[Multiply the weights by the function values.]

$$\bar{\phi}_{j,k} = W_{j,k} \cdot \phi(x_j, y_k), \tag{3.62}$$

for $j = 0, \ldots, N-1$, $k = 0, \ldots, N-1$.

5 $O(N^2 \log N)$ **Comment**[Obtain the corrected Fourier transform]

$$\hat{f}_{l,m} = \sum_{j=1}^{N} \sum_{k=1}^{N} \bar{\phi}_{j,k} \cdot e^{\frac{i2\pi jl}{N}} \cdot e^{\frac{i2\pi km}{N}}, \tag{3.63}$$

for $l = 0, \ldots, N-1$, $m = 0, \ldots, N-1$.

Remark 3.13 Suppose that $\phi(x, y)$ is a constant function (this is the problem addressed in [29], [5]). Then it is clear that Algorithm 3.1 reduces to Step 0, 1, 2. In other words, only 1 FFT and 1 multiplication are required to solve the problem.

Remark 3.14 It is also clear that the discontinuity is not restriced to the charateristic function of a square (see, 3.56). In fact, the algorithm would perform identically for any discontinuity or integrable singularity, as long as the Fourier coefficients (see, 3.58) are analytically available.

3.7 Numerical Examples

We have written FORTRAN implementations of the algorithms of this chapter using double precision arithmetic, and have applied these programs to a variety of situations. The implementations of these algorithms have been tested on a SPARCstation 10. Five experiments are described in this section, and their results are summarized in Tables 2.1, 2.2.

Example 1:

We consider the cosine transform of the singular function f,

$$\int_0^{\pi} f(x) \cdot \cos(k \cdot x)dx, \tag{3.64}$$

where $f(x)$ is given by

$$f(x) = \log(x) \cdot e^{-4x^2}. \tag{3.65}$$

Combining formula (3.2) with the appropriate quadrature weights, (see, Chapter 2) we construct a corrected cosine transform. In the example presented we use 20 quadrature weights to obtain a 40th order rule. In Table 1, we present convergence results of the corrected FFT. The first column of Table 1 contains the number N of nodes discretizing the interval $[0, \pi]$. Columns 2-4 contain the relative error of the numerical solution as compared with the analytically obtained one, for various values of k. We empirically observe that that the function tabulated at the Nyquist sampling rate (of two-points per wavelength) is integrated to single precision accuracy, and when tabulated at twice the Nyquist rate is integrated to double precision accuracy. Similar results are obtained for other singularities of the form $s(x) = x^{\alpha}$ where $(|\alpha| < 1)$ using the weights in Chapter 5.

Example 2:

We consider the cosine transform of the singular function,

$$\int_{-\pi}^{\pi} f(x) \cdot \cos(k \cdot x)dx \tag{3.66}$$

where $f(x)$ is given by

$$f(x) = [\frac{1}{(\pi^2 - x^2)^{\frac{1}{2}}}] \tag{3.67}$$

Combining formula (3.2) with the appropriate quadrature weights, (see, Chapter 2) we constructed a corrected cosine transform. In the example presented we used 20 quadrature weights to obtain a 40th order rule. In Table 2, we present convergence results of the corrected FFT. The first column of Table 1 contains the number of nodes discretizing the interval $[-\pi, \pi]$. Columns 2-4 contain the relative error of the numerical solution as compared with the analytically obtained one for various values of k. We empirically observe that the function tabulated at the twice the Nyquist sampling rate (of two-points per wavelength) is integrated to single precision accuracy, and when tabulated at four times the Nyquist rate is integrated to double precision accuracy.

Example 3:

We use the term non-separable for functions of the form

$$f(x) = \phi(x) + s(x) \cdot \psi(x), \tag{3.68}$$

where the singularity cannot be separated (i.e., it is not possible to express the function in the form $g(x) = s(x) \cdot h(x)$). However, it is still possible to numerically compute the FFT accurately (albeit, with a slightly inferior convergence rate compared to the previous two examples). We consider the Fourier transform of the singular function,

$$\int_0^{2\pi} f(x) \cdot e^{ikx}, \tag{3.69}$$

where f is given by

$$f(x) = e^{-x^2} \cos(x) + e^{-x^2} \sin(x) \log(x)). \tag{3.70}$$

In Chapter 2, a quadrature formula is derived to approximate integrals of functions of the form (3.70). Combining formula (3.2) with the appropriate correction weights we constructed a corrected FFT. In the example presented we used a 10th order corrected rule. In Table 3, we present convergence results of the corrected FFT. The first column of Table 1 contains the number of nodes discretizing the interval $[0, 2\pi]$. Columns 2-4 contain the relative error of the numerical solution as compared with the analytically obtained one for various values of k. We observe that the convergence is slower than in the previous two examples. The reasons for being able to use only a 10th order rule, as opposed to a 40th order rule, when the singularity is non-separable, are discussed in detail in Chapter 2.

Example 4:

In this example, we evaluate the Fourier transform of a piecewise *constant* function $s(x,y)$ which is a linear combination of characteristic functions of a pseudo-random combination of rectangles. More specifically, suppose that $h = \frac{2\pi}{N}$, $s(x,y) = s_{k,j}$ for $x \in [k,(k+1)h]$, $y \in [j,(j+1)h]$, and the N^2 coefficients $s_{k,j}$ are randomly distributed numbers on the unit square $[0,1]$. Then we evaluate the integrals

$$\hat{f}(k,j) = \int_0^{2\pi}\int_0^{2\pi} s(x,y)e^{ikx}e^{ijy}dxdy, \tag{3.71}$$

to double precision accuracy, for all $k = -(N-1)/2,\ldots,(N-1)/2$, and $j = -(N-1)/2,\ldots,(N-1)/2$ by means of Algorithm 3.1.

In Table 4, we present the results of the corrected FFT. The first column of Table 3 contains the number of nodes discretizing the interval $[0,2\pi]$. Columns 2-4 contains the evaluation time required for Step 0, Step 1, Step 2, of Algorithm 3.1. Column 5 contains the ratio of the time required by the algorithm compared to the time required for the standard implemenation of the FFT. Finally, Column 5 contains the L_2 error of the numerical solution as compared with the analytically obtained one.

Example 5:

In this example, we evaluate the Fourier transform of a piecewise *periodic* function $s(x,y)$ which is a linear combination of characteristic functions of a pseudo-random combination of rectangles. More specifically, $h = \frac{2\pi}{N}$, $s(x,y) = s_{k,j}$ for $x \in [k,(k+1)h]$, $y \in [j,(j+1)h]$, and the N^2 coefficients $s_{k,j}$ were randomly distributed numbers on the unit square $[0,1]^2$. We evaluated the integrals

$$\hat{f}(k,j) = \int_0^{2\pi}\int_0^{2\pi} \phi(x,y)\cdot s(x,y)e^{ikx}e^{ijy}dxdy, \tag{3.72}$$

to double precision accuracy, for all $k = -(N-1)/2,\ldots,(N-1)/2$, and $j = -(N-1)/2,\ldots,(N-1)/2$, and

$$\phi(x,y) = e^{-5/2((x-\pi)^2+(y-\pi)^2)}. \tag{3.73}$$

by means of Algorithm 3.1.

In Table 5, we present the results of the corrected FFT. The first column of Table 3 contains the number of nodes discretizing the interval $[0, 2\pi]$. Columns 2-7 contain the evaluation time required for Steps 0-5 of Algorithm 3.1. Column 8 contains the ratio of the time required by the algorithm compared to the time required for the standard implemenation of the FFT. Finally, Column 9 contains the L_2 error of the numerical solution as compared with the analytically obtained one.

Remark 3.15 The reason for the low of accuracy for small N ($N = 16, 32$) is that for these values, the function (3.73) is under sampled.

3.8 Generalizations and conclusions

A group of algorithms has been presented for the rapid evaluation of the Fourier transforms of function with singularities. The algorithms are based on a combination of quadrature schemes with the FFT; they overcome the accuracy problems associated with computing the Fourier transform of discontinuous functions. All the algorithms admit several straightforward generalizations.

1. There are classes of singularities not covered in this chapter for which versions of Theorem 3.4, and Theorem 3.6 can be fairly easily proven.

2. In [9], [10] a collection of algorithms are designed for the efficient computation of certain generalizations of the DFT, namely the application and inversion of the transformation $F : \mathbf{C}^N \to \mathbf{C}^N$ defined by the formulae

$$F(\alpha)_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx_j}, \tag{3.74}$$

for $j = 1, \ldots, N$, where $x = \{x_1, \ldots, x_N\}$ is a sequence of real numbers in $[-\pi, \pi]$ and $\alpha = \{\alpha_{-N/2}, \ldots, \alpha_{N/2-1}\}$ is a sequence of complex numbers. The number of arithmetic operations required by each of the algorithms of this paper is proportional to

$$N \cdot \log N + N \cdot \log\left(\frac{1}{\varepsilon}\right) \tag{3.75}$$

where ε is the desired accuracy, compared with $O(N^2)$ operations required for the direct application and $O(N^3)$ for the direct inversion of the transformation described by (3.74).

The algorithms designed for the non-equispaced FFT are slowly convergent schemes for the approximation integrals. More specifically, since the function tabulation is at non-Gaussian nodes the order of convergence is the same as the convergence of the trapezoidal rule which the DFT implements, i.e.,

$$\left| \int_0^{2\pi} \alpha(k) e^{ikx_j} dk - \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx_j} \right| < O(1/N^2). \tag{3.76}$$

The procedures developed in this chapter can be applied to this situation to design rapidly convergent schemes. This work has been completed and the results will be reported at a later date.

In Section 3.6 we assume that the singularities lie on grid points. In this situation the equispaced FFT is used for the derivation of quadrature weights. However, there are situations when the singularities do not lie on grid points. It is clear that in this situation a non-equispaced FFT may be used to obtain the quadrature weights instead.

3. There are a number of applications based on the approach of this chapter to the study of diffraction techinques in optics. For example, the Fraunhofer diffraction formula is of the form

$$I(u_k, u_j, D) \approx |\frac{1}{\lambda D} \int_{\mathbb{R}^2} A(x,y) e^{-i\frac{2\pi}{\lambda D} u_x + u_y} dx dy| \tag{3.77}$$

where $A(x, y)$ is the aperture function. In many situations the aperture function accounts for the finite extent of the aperture; numerically a discontinuity. Currently, standard numerical techniques, approximate integrals of the form (3.77) by means of the DFT. Oversampling and extrapolation are used to combat the error resulting from the discontinuity; the algorithms of this chaper may be implemented in a straightforward manner to the evalutation of these integrals.

Table 3.1: Accuracy of the locally corrected FFT for evaluating singular integrals

Example 1:

$$\int_0^{\pi} [\log(x) \cdot e^{-4x^2}] \cdot \cos(k \cdot x) dx$$

N	k=N	k=N/2	k=N/4
64	0.107E+00	0.578E-07	0.437E-14
128	0.914E-01	0.570E-08	0.109E-13
256	0.855E-01	0.255E-08	0.320E-13
512	0.829E-01	0.199E-08	0.699E-13
1024	0.818E-01	0.184E-08	0.120E-13
2048	0.812E-01	0.179E-08	0.957E-13

Example 2:

$$\int_{-\pi}^{\pi} [\frac{1}{\sqrt{\pi^2 - x^2}}] \cos(k \cdot x) dx$$

N	k=N/2	k=N/4	k=N/8
32	0.165E-01	0.829E-08	0.897E-15
64	0.109E-02	0.334E-07	0.868E-14
128	0.671E-02	0.668E-08	0.298E-13
256	0.784E-02	0.175E-07	0.107E-13
512	0.798E-02	0.242E-07	0.361E-13
1024	0.795E-02	0.257E-07	0.597E-14

Example 3:

$$\int_0^{2\pi} [e^{-x^2} \cos(x) + e^{-x^2} \sin(x) \cdot \log(x)] \cdot e^{ikx},$$

N	k=N/8	k=N/16	k=N/32	k=N/64	k=N/128
256	0.709E-01	0.131E-03	0.119E-06	0.372E-09	0.130E-10
512	0.134E+00	0.750E-04	0.114E-07	0.462E-10	0.265E-12
1024	0.192E+00	0.490E-04	0.486E-07	0.510E-10	0.162E-12
2048	0.250E+00	0.301E-04	0.802E-07	0.607E-10	0.183E-12
4096	0.309E+00	0.130E-04	0.113E-06	0.774E-10	0.295E-12

Table 3.2: Applications of quadratures to VLSI

Example 4: Piecewise constant function.

$$\hat{f}(k,j) = \int_0^{2\pi}\int_0^{2\pi} s(x,y)e^{ikx}e^{ijy}dxdy,$$

N	Step 0	Step 1	Step 2	t_{alg}/t_{FFT}	error
16	0.03	0.16	0.02	1.12	0.132E-14
32	0.11	0.85	0.11	1.12	0.156E-14
64	0.48	3.43	0.53	1.15	0.121E-14
128	1.82	14.94	1.84	1.12	0.502E-14
256	8.69	65.39	7.19	1.11	0.104E-14

Example 5: Piecewise smooth function.

$$\hat{f}(k,j) = \int_0^{2\pi}\int_0^{2\pi} \phi(x,y)\cdot s(x,y)e^{ikx}e^{ijy}dxdy,$$

N	Step 0,1,2	Step 3	Step 4	Step 5	t_{alg}/t_{FFT}	error
16	0.200	0.180	0.040	0.150	3.375	0.218E-02
32	1.060	0.840	0.110	0.840	3.235	0.683E-06
64	4.260	3.290	0.630	3.290	3.322	0.403E-13
128	18.010	14.540	1.720	14.580	3.240	0.542E-14
256	79.030	64.060	6.700	63.940	3.216	0.704E-15

All times are measured in 1/100 of a second.

Chapter 4

An algorithm for the fast Hankel transform

4.1 Introduction

Given a function $f \in c^2[0,\infty]$ and a real number $a > 0$, the Hankel transform is defined by the formula

$$g(a) = \int_0^\infty f(x) J_0(ax) dx.$$

Hankel transforms are frequently encountered in applied mathematics and computational physics. Their applications include vibrations of a circular membrane, flow of heat in a circular cylinder, wave propagation in a three-dimensional medium and many others. However, attempts to use Hankel transforms as a numerical tool (as opposed to analytical apparatus) tend to meet with serious difficulty: given a function $f : [0, A] \rightarrow \mathbf{R}$, tabulated at N nodes, it takes $O(N^2)$ operations to obtain the numerical Hankel transform

$$\int_0^A f(x) J_0(a \cdot x) dx, \tag{4.1}$$

for N values of a. In other words, unlike the Fourier series, the Chebyshev expansion or the Legendre series (see, for example [4]), the Bessel functions do not have a fast transform associated with them. Therefore, whenever possible, the Hankel transform is avoided in favor of an expansion for which a fast transform exists.

In this paper we present a procedure for the rapid evaluation of integrals of the form (4.1), to any degree of precision, expending CPU time proportional to $N \log N$. More specifically, suppose that $h = \frac{A}{N-1}$, $x_i = ih$, $a_j = \frac{\pi j}{(N-1)h}$, and $f : [0, A] \to \mathbf{R}$ is a function tabulated at N equispaced nodes $x_0, x_1, ..., x_{N-1}$. Then the integrals

$$g(a_j) = \int_0^A f(x) J_0(a_j x) dx, \tag{4.2}$$

are computed for all $j = 0, 1, 2..., N-1$, in $O(N \log N)$ operations.

Remark 4.1 Suppose that a function $f : [0, \infty] \to \mathbf{R}$ has compact support, i.e., there exist a pair of real numbers A, ϵ such that $|f(x)| \leq \epsilon$ for all $x > A$. Then, it is obvious that (4.2) approximates, to a degree of precision ϵ, the exact Hankel transform,

$$g(a) = \int_0^\infty f(x) J_0(ax) dx. \tag{4.3}$$

Our algorithm for the Hankel transform is based on well known facts from classical analysis. The Hankel transform is decomposed into two separate integral operators, each of which is evaluated independently. The first operator is rapidly computed using a fast cosine transform of a non-periodic function. Since the convergence of this procedure is slow (O(1/n)) we accelerate it by means of an appropriate end-point correction procedure. The second integral transform has a singular kernel. A trapezoidal approximation to the integral is computed rapidly using a generalized version of the one-dimensional fast multipole method (1-D FMM); a different set of end-point corrections is used to assure the rapid convergence of this second step.

In the following section, we summarize several facts from approximation theory, classical numerical analysis, and [18] to be used in the subsequent sections. Section 4.3 provides the numerical apparatus for the algorithm, and is divided into three subsections. In §4.3.1 a procedure for the accurate evaluation of a cosine transform of a non-periodic function is described. In § 4.3.2 we design a very high-order end-point corrected trapezoidal rule for integrals of the form (4.2). In §4.3.3 we present an algorithm based on the 1-D FMM which computes a trapezoidal approximation to the integrals (4.2) in $O(N)$ operations. In §4.4 we combine the results of Sections 4.3.1, 4.3.2, and 4.3.3 to present the detailed description and complexity analysis of our algorithm, for the fast Hankel transform. Finally, in §4.5 we provide results of several numerical experiments, and in §4.6 we discuss several straightforward generalizations and applications of the algorithm of this paper.

4.2 Mathematical and Numerical Preliminaries

In this section we summarize several well known results from classical analysis and approximation theory to be used in this paper.

4.2.1 An expression for the Bessel function J_0

As is well known, for any $z \in \mathbf{C}$,

$$J_0(z) = \frac{1}{\pi}\int_0^{\pi} \cos(z\cos\theta)d\theta, \tag{4.4}$$

where J_0 denotes the Bessel function of order 0 (see, for example [2]). It follows immediately from (4.4) that, for any function $f \in c^2[0,A] \to \mathbf{R}$, and any real number a,

$$\int_0^A f(x)J_0(ax)dx = \frac{1}{\pi}\int_{-a}^{a} \frac{1}{\sqrt{a^2-u^2}}\int_0^A f(x)cos(ux)dxdu. \tag{4.5}$$

4.2.2 Chebyshev Polynomials

In this section we summarize several well known facts about Chebyshev polynomials. The following three classical definitions can be found, for example, in [13].

Definition 4.1 *The n-th degree Chebyshev polynomial $T_n(x)$ is defined by the following equivalent formulae:*

$$T_n(x) = \cos(n\arccos x), \tag{4.6}$$

$$T_n(x) = \frac{1}{2}\cdot\left((x+\sqrt{x^2-1})^n + (x-\sqrt{x^2-1})^n\right). \tag{4.7}$$

The proof of the following lemma can be found in, for example, in[2].

Lemma 4.1 *Suppose that $n \geq 0$ is an integer. Then*

$$\int_{-1}^{1} \frac{T_0(u)du}{\sqrt{1-u^2}} = \pi, \tag{4.8}$$

and,

$$\int_{-1}^{1} \frac{T_n(u)du}{\sqrt{1-u^2}} = 0, \tag{4.9}$$

for all $n \geq 1$.

Definition 4.2 *The roots* $\{t_1, \ldots, t_n\}$ *of the* n*-th degree Chebyshev polynomial* T_n *lie in the interval* $[-1, 1]$ *and*

$$t_k = -\cos\left(\frac{2k-1}{n} \cdot \frac{\pi}{2}\right), \tag{4.10}$$

for $k = 1, \ldots, n$. *They are referred to as Chebyshev nodes of order* n.

Definition 4.3 *Given an integer* $n \geq 1$, *we will denote by* $u_1, \ldots, u_n$ *the set of polynomials of order* $n-1$ *defined by the formulae*

$$u_j(t) = \prod_{k \neq j}^{n} \frac{t - t_k}{t_j - t_k}, \tag{4.11}$$

for $j = 1, \ldots, n$, *where* t_k *are defined by (4.10).*

The proofs of the following two lemmas are well known from classical theory of Chebyshev approximation, and can be found, for example, in [9], [11].

Lemma 4.2 *Suppose that* $p \geq 2$, $b > 0$, *and* x_0 *are real numbers such that* $|x_0| < b$. *Suppose further that* $\{t_1, \ldots, t_p\}$ *are Chebyshev nodes on the interval* $[x_0 - b, x_0 + b]$. *Then,*

$$\left| \frac{1}{\sqrt{y^2 - x_0^2}} - \sum_{m=1}^{p} \frac{1}{\sqrt{y^2 - t_m^2}} \cdot u_m(x_0) \right| < O(\frac{1}{5^p}) \tag{4.12}$$

for all $|y| > 3b$.

Lemma 4.3 *Suppose that* $p \geq 2$, $b > 0$, *and* y_0 *are real numbers such that* $|y_0| > 3b$. *Suppose further that* $\{t_1, \ldots, t_p\}$ *are Chebyshev nodes on the interval* $[y_0 - b, y_0 + b]$. *Then,*

$$\left| \frac{1}{\sqrt{y_0^2 - x^2}} - \sum_{m=1}^{p} \frac{1}{\sqrt{t_m^2 - x^2}} \cdot u_m(y_0) \right| < O(\frac{1}{5^p}) \tag{4.13}$$

for all $|x| < b$.

4.2.3 The fast cosine transform

The following definition of the discrete cosine transform can be found for example, in [30].

Definition 4.4 *For a real sequence* $\{f_0, \ldots, f_{N-1}\}$ *the discrete cosine transform* $\{F_j^T\}$ *is defined by the formula*

$$F_j^T = \frac{\pi}{N-1}\left(\sum_{i=1}^{N-2} f_i \cos\left(\frac{\pi \cdot j \cdot i}{N-1}\right) + \frac{f_0 + (-1)^j f_{N-1}}{2}\right) \tag{4.14}$$

for all $j = 0, 1, ..., N-1$.

Remark 4.2 The fast cosine transform is an algorithm, based on the FFT (see, for example [30]) to evaluate the discrete cosine transform (DCT) in $O(N \log N)$ operations.

Remark 4.3 The discrete cosine transform is a trapezoidal approximation (see Definition 2.1) to the exact cosine transform. More specifically, suppose that $f \in c^2[0, \pi]$, $h = \pi/(N-1)$, $x_i = ih$, $f_i = f(x_i)$, and $\{F_j^T\}$ is defined in (4.14). Then

$$F_j^T \approx \int_0^{\pi} f(x) \cos(j \cdot x) dx, \tag{4.15}$$

for all $j = 0, 1, ..., N-1$.

The following well known theorem provides an error estimate for the approximation to the exact cosine transform given by the discrete cosine transform. The proof follows immediately from the combination of (4.14) and Lemma 2.1.

Theorem 4.4 *Suppose that* $f \in c^2[0, \pi]$. *Suppose further that* $h = \pi/(N-1)$, $x_i = ih$, *and* $f_i = f(x_i)$. *Then the discrete cosine transform is second order convergent, i.e., there exists some real* $c > 0$ *such that*

$$\mid F_j^T - \int_0^{\pi} f(x) \cos(j \cdot x) \mid < \frac{c}{N^2} \tag{4.16}$$

for all $j = 0, 1, ..., N-1$.

The following theorem is less widely known. The proof also follows immediately from the combination of (4.14) and Lemma 2.1; it can also be found in [11].

Theorem 4.5 *Suppose that $f \in c^m[0,\pi]$ is an even function (i.e., f(x) = f(-x)). Suppose further that $h = \pi/(N-1)$, $x_i = ih$, and $f_i = f(x_i)$. Finally, suppose that $f(\pi) = f'(\pi) = f''(\pi) = \ldots = f^m(\pi) = 0$. Then the discrete cosine transform is a rule of order m, i.e., there exists some real $c > 0$ such that*

$$| F_j^T - \int_0^\pi f(x)\cos(j \cdot x) | < \frac{c}{N^m} \tag{4.17}$$

for all $j = 0, 1, \ldots, N-1$.

4.3 Numerical apparatus

4.3.1 The corrected fast cosine transform

Remark 4.4 When a function is even, the discrete cosine transform provides a remarkably good approximation to the exact cosine transform (see Theorem 4.5). For functions that are not even, we use end-point corrections to accelerate the convergence of the DCT.

Definition 4.5 *For a finite real sequence $\{f_{-(n-1)}, \ldots, f_0, \ldots, f_{n-1}\}$ we define the corrected discrete cosine transform $\{F_j^C\}$ by the formula*

$$F_j^C = F_j^T + \frac{\pi}{N-1}\sum_{i=1}^{N-2}(-f_{-i} + f_i)\cos\left(\frac{\pi \cdot i \cdot j}{N-1}\right)\beta_i^{2N-3}, \tag{4.18}$$

for all $j = 0, 1, \ldots, N-1$, where $\{F_j^T\}$ is defined in (4.14), and β_i^{2N-3} are the correction coefficients defined in (2.52).

The following corollary provides an error estimate for the approximation to the exact cosine transform given by the corrected discrete cosine transform (CDCT). It is an immediate consequence of Theorem 2.14.

Corollary 4.6 *Suppose that $f \in c^m[-\pi,\pi]$. Suppose further that $h = \pi/(N-1)$, $x_i = ih$, and $f_i = f(x_i)$ for all $i = 0, \pm 1, \ldots, \pm N-1$. Finally, suppose that $f(\pi) = f'(\pi) = f''(\pi) = \ldots = f^m(\pi) = 0$. Then there exists some real $c > 0$ such that*

$$| F_j^C - \int_0^\pi f(x)\cos(j \cdot x) | < \frac{c}{N^m} \tag{4.19}$$

for all $j = 0, 1, \ldots, N-1$.

Remark 4.5 The CDCT requires that the function be tabulated outside the interval of integration $[0:\pi]$. However, if a function is odd (or even) this requirement is obviated, i.e., the function needs to be tabulated only within the interval of integration.

Observation 4.6 *Suppose that $f \in c^m[0,\pi]$ is an even function (i.e., $f(x) = f(-x)$) satisfying the conditions of Corrollary 4.6. Then it follows from (4.18) that*

$$F_j^C = F_j^T, \tag{4.20}$$

for all $j = 0, 1, ..., N-1$.

Observation 4.7 *Suppose that $f \in c^m[0,\pi]$ is an odd function (i.e., $f(x) = -f(-x)$) satisfying the conditions of Corrollary 4.6. Then it follows from (4.18) that*

$$F_j^C = F_j^T - \frac{\pi}{N-1}\sum_{i=1}^{N-2}(2\cdot f_i\cdot \beta_i^{2N-3})\cos(\frac{\pi\cdot i\cdot j}{N-1}) \tag{4.21}$$

for all $j = 0, 1, ..., N-1$.

Rapid evaluation of the corrected discrete cosine transform (CDCT)

Suppose that $\{f_{-(n-1)}, \ldots, f_0, \ldots, f_{n-1}\}$ is a finite real sequence. Suppose further that we define the real sequence $\{\hat{f}_0, \ldots, \hat{f}_{n-1}\}$ by the formulae

$$\hat{f}_0 = f_0, \quad \hat{f}_{N-1} = f_{N-1}, \tag{4.22}$$

and,

$$\hat{f}_i = f_i + \beta_i^{2N-3}(-f_{-i} + f_i), \tag{4.23}$$

for all $i = 1, 2, ..., N-2$, where the real coefficients β_k^{2N-3} are defined in (2.52). Then it immediately follows from the combination of (4.18) and (4.22) that

$$F_j^C = \frac{\pi}{N-1}(\sum_{i=1}^{N-2}\hat{f}_i\cos(\frac{\pi\cdot i\cdot j}{N-1}) + \frac{\hat{f}_0 + (-1)^j\hat{f}_{N-1}}{2}) \tag{4.24}$$

for all $j = 0, 1, 2, ..., N-1$.

Remark 4.8 Given a real sequence $\{f_{-(n-1)}, \ldots, f_0, \ldots, f_{n-1}\}$, the sequence $\{\hat{f}_0, \ldots, \hat{f}_{n-1}\}$ can be computed (using (4.22)) in $O(N)$ operations. Subsequently sums of the form (4.24) can be computed in $O(N\log N)$ operations using the FCT (see Remark 4.2). Thus the corrected discrete cosine transform $\{F_j^C\}$ can be computed in $O(N\log N) + O(N)$ operations.

The approach we use for correcting the fast cosine transform for non-periodic functions (i.e., preprocessing the tabulated function with correction weights prior to the use of the FCT algorithm) can be used to evaluate the Fourier transform of functions with singularities. The details of this scheme are developed in [19].

4.3.2 End-point corrected trapezoidal quadrature rules for singular functions of the form $\frac{F(u)}{(a^2-u^2)^{1/2}}$

In this section we develop an end-point corrected quadrature formula to approximate the definite integral

$$\int_{-a}^{a} \frac{F(u)}{\sqrt{a^2-u^2}} du, \tag{4.25}$$

with $a > 0$, and $F \in c^k[-a, a]$, an even function (i.e., $F(-u) = F(u)$).

We define the corrected trapezoidal rule $T_{\nu^N}^N$ by the formula

$$T_{\nu^N}^N\left(\frac{F(u)}{\sqrt{a^2-u^2}}\right) = h \sum_{l=-(N-2)}^{N-2} \frac{F(x_l)}{\sqrt{a^2-x_l^2}} \; + \; h\sum_{i=1}^{k} \nu_i^N \frac{F(y_i)}{\sqrt{|a^2-(y_i)^2|}}, \tag{4.26}$$

where $h = a/(N-1)$, $x_l = lh$, $y_i = a - hi$ for all $1 \le i \le k/2$, and $y_i = a + h(i - k/2)$ for all $k/2 + 1 \le i \le k$. We will use the expression $T_{\nu^N}^N$ with appropriately chosen ν_i^N as a quadrature formulae to approximate integrals of the form (4.25), and the following construction provides a tool for finding ν_i^N, so that the rule is of order $2k-2$, i.e., there exists a real $c > 0$ such that

$$\left| T_{\nu^N}^N\left(\frac{F(u)}{\sqrt{a^2-u^2}}\right) - \int_{-a}^{a} \frac{F(u)}{\sqrt{a^2-u^2}} du \right| < \frac{c}{N^{2k-2}}. \tag{4.27}$$

Remark 4.9 The correction nodes $\{y_1, \ldots, y_k\}$ are equispaced nodes on both sides of the singularity a (i.e., half the correction nodes lie outside the interval of integration). The approach we use in this paper is similar to that of [18]; where a group of quadrature formulae is presented applicable to functions with end-point singularities, taking advantage of functional information outside the interval of integration.

Construction of the quadrature weights ν^N

For any pair of positive integers k, N ($k < N$), we will consider the following system of linear algebraic equations with respect to the unknowns $\{\nu_1^N, \ldots, \nu_k^N\}$:

$$\sum_{p=1}^{k} \frac{T_{2i-2}(y_p)}{\sqrt{|a^2-(y_p)^2|}}\nu_p^N = \frac{1}{h}\left(\int_{-a}^{a} \frac{T_{2i-2}(u)}{\sqrt{a^2-u^2}}du - h \sum_{l=-(N-2)}^{N-2} \frac{T_{2i-2}(x_l)}{\sqrt{a^2-x_l^2}}\right), \tag{4.28}$$

for all $i = 1, 2, ..., k$, where h, x_l, and y_i are defined in (4.26). In (4.28) T_{2i-2} is the Chebyshev polynomial defined in (4.6).

The following simple observation is crucial to the development of our algorithm for the fast Hankel transform.

Observation 4.10 *The linear system (4.28) is independent of the length of the interval a i.e., the quadrature weights $\nu_1^N, \nu_2^N, ..., \nu_k^N$ are only dependent on the number of points N used in the trapezoidal approximation to the integral (4.25).*

Thus, by substituting $a = 1$ in (4.26), the unknowns $\{\nu_1^N, \nu_2^N, ..., \nu_k^N\}$ alternatively can be determined by solving the system of equations:

$$\sum_{p=1}^{k} \frac{T_{2i-2}(y_p)}{\sqrt{|1-y_p^2|}}\nu_p^N = \frac{1}{h}\left(\int_{-1}^{1} \frac{T_{2i-2}(u)}{\sqrt{1-u^2}}du - h \sum_{l=-(N-2)}^{N-2} \frac{T_{2i-2}(x_l)}{\sqrt{1-x_l^2}}\right), \tag{4.29}$$

for all $i = 1, 2, ..., k$, where $h = \frac{1}{N-1}$, $x_l = lh$, $y_p = 1 - hp$ for all $1 \leq p \leq k/2$, and $y_p = 1 + h(p - k/2)$ for all $k/2 + 1 \leq p \leq k$.

Remark 4.11 Any polynomial basis can be chosen to construct the linear system (4.29) above. Our choice of the Chebyshev polynomials T_i as the polynomial basis is simply for the reason that the right-hand side of of the linear system (4.29) can be simplified due to Lemma 4.1.

Hence, we may alternatively solve the following system of equations to obtain the quadrature weights $\nu_1^N, \nu_2^N, ..., \nu_k^N$:

$$\sum_{p=1}^{k} \frac{T_{2i-2}(y_p)}{\sqrt{|1-y_p^2|}}\nu_p^N = \frac{1}{h}\left(\pi - h \sum_{l=-(N-2)}^{N-2} \frac{T_{2i-2}(x_l)}{\sqrt{1-x_l^2}}\right), \tag{4.30}$$

for $i = 1$, and

$$\sum_{p=1}^{k} \frac{T_{2i-2}(y_p)}{\sqrt{|1-y_p^2|}} \nu_p^N = \frac{1}{h}(-h \sum_{l=-(N-2)}^{N-2} \frac{T_{2i-2}(u_l)}{\sqrt{1-x_l^2}}), \tag{4.31}$$

for $i = 2, 3, ..., k$.

Convergence of the rule $T_{\nu^N}^N$

The use of expressions $T_{\nu^N}^N$ as quadrature formulae to approximate integrals of the form (4.25) is based on the following theorem.

Theorem 4.7 *Suppose that $k, N > 2$ are a pair of positive integers. Further, suppose that h, a are positive real numbers with $h = a/(N-1)$. Also, suppose that the systems (4.30), (4.31) have solutions $(\nu_1^N, \nu_2^N, ..., \nu_k^N)$ for all N. Finally, suppose that $F \in c^k[-a-kh, a+kh]$ is an even function. Then the rule $T_{\nu^N}^N$ is of order $2k-2$, i.e., there exists some real number c such that*

$$| T_{\nu^N}^N(\frac{F(u)}{\sqrt{a^2-u^2}}) - \int_{-a}^{a} \frac{F(u)}{\sqrt{a^2-u^2}} du |< \frac{c}{N^{2k-2}} \tag{4.32}$$

Proof. Let us define the functions

$$\begin{aligned} f(u) &= \frac{F(u)}{\sqrt{a^2-u^2}}, \\ s(u) &= \frac{1}{\sqrt{a^2-u^2}}, \end{aligned} \tag{4.33}$$

where F satisfies the conditions of Theorem 4.7. Applying the Taylor expansion to the function f at $u = 0$ we obtain

$$f(u) = P(f)(u) + R_k(F)(u)s(u), \tag{4.34}$$

where

$$P(f)(u) = s(u) \sum_{i=1}^{k} \frac{F^{(i)}(0)}{(i!)} u^{2i-2}. \tag{4.35}$$

Substituting (4.34) into (4.32), we obtain

$$\begin{aligned} | T_{\nu^N}^N(f) - \int_{-a}^{a} f(u)du | &\leq | T_{\nu^N}^N(P(f)) - \int_{-a}^{a} P(f)(u)du | + \\ & | T_{\nu^N}^N((R_k(F) \cdot s) - \int_{-a}^{a} ((R_k(F(u))s(u))(u). \end{aligned} \tag{4.36}$$

Due to (4.30), (4.31)

$$T_{\nu^N}^N(P(f)) - \int_{-a}^{a} P(f)(u)du = 0, \tag{4.37}$$

and we have

$$| T_{\nu^N}^N(f) - \int_{-a}^{a} f(u)du | \leq | T_\nu^N(s \cdot R_k(F)) - \int_{-a}^{a} (s \cdot R_k(F))(u)du | . \tag{4.38}$$

Finally, due to Lemma 2.12 there exists some real $c_1 > 0$ such that

$$| T_\nu^N(s \cdot R_k(F)) - \int_{-a}^{a} (s \cdot R_k(F))(u)du | < \frac{c_1}{N^{2k-2}}. \tag{4.39}$$

Now, the conclusion of the theorem follows from the combination of (4.39), and (4.37). □

The following corollary is an immediate consequence of Theorem 4.7.

Corollary 4.8 *Suppose that $F \in c^{2k}[-N, N]$ is an even function. Suppose further that F is tabulated at equispaced nodes on the interval $[-N, N]$. Finally, suppose that the systems (4.30), (4.31) have solutions $(\nu_1^j, \nu_2^j, ..., \nu_k^j)$ for all $j = 1, 2, 3, ..., N-1$. Then there exist real numbers c_j such that*

$$| T_{\nu^j}^j(\frac{F(u)}{\sqrt{j^2 - u^2}}) - \int_{-j}^{j} \frac{F(u)}{\sqrt{j^2 - u^2}}du | < \frac{c_j}{j^{2k-2}} \tag{4.40}$$

for all $j = 1, 2, ..., N-1$.

The following theorem easily follows from the combination of (4.18), Corollary 4.8, and Corollary 4.6.

Theorem 4.9 *Suppose that $f \in c^{2k}[-\pi, \pi]$ is an odd or even function. Suppose further that $h = \pi/(N-1)$, $x_i = ih$, and $f_i = f(x_i)$ for all $i = 0, \pm 1, \ldots, \pm N-1$. Finally, suppose that*

$$F_i^C = \frac{\pi}{N-1}(\sum_{j=0}^{N-1} \hat{f}(x_j) \cos(\frac{\pi \cdot i \cdot j}{N-1}), \tag{4.41}$$

where $\hat{f}$ is defined in (4.22). Then there exist real numbers c_j such that

$$| T_{\nu^j}^j(\frac{F^C(u)}{\sqrt{j^2 - u^2}}) - \int_0^{\pi} f(x)J_0(j \cdot x)dx | < \frac{c_j}{j^{2k-2}}, \tag{4.42}$$

for all $j = 0, \pm 1, \pm 2, \pm 3, ..., \pm N-1$.

Approximation of the Hankel tranform using the rule $T^j_{\nu_j}$

Suppose that we define the trapezoidal approximation g^T to the Hankel transform by the formula

$$g^T_j = h \sum_{l=-(j-1)}^{j-1} \frac{F^C_l}{\sqrt{j^2 - l^2}}, \tag{4.43}$$

and the correction g^C by the formula

$$g^C_j = h \sum_{i=1}^{k} \nu^j_i \frac{F^C_{y^j_i}}{\sqrt{j^2 - (y^j_i)^2}}, \tag{4.44}$$

for all $j = \pm1, \pm2, ..., \pm N-1$, where $y^j_i = j - i$ for $1 \le i \le k/2$, and $y^j_i = j + (i - k/2)$ for $k/2 + 1 \le i \le k$. It follows immediately from the combination of (4.43), (4.44), and (4.26) that

$$g^C_j + g^T_j = T^j_{\nu^j}\left(\frac{F^C_j}{\sqrt{j^2 - u^2}}\right). \tag{4.45}$$

Remark 4.12 It is obvious that g^T_j can be computed for all $j = 0, \pm1 \pm 2, ..., \pm N-1$,in $O(N^2)$ operations, and g^C_j can be computed in $O(kN)$ operations. In the following section we discuss how to compute sums of the form (4.43) in $O(N)$ operations.

Remark 4.13 The numerical stability of the scheme developed above, for the approximation of integrals of the form (4.25), is dependent on the assumption that the size of the quadrature weights $(\nu^j_1, \nu^j_2, ..., \nu^j_k)$ is small. We have observed in [18] that the size of the quadrature weights can be suppressed (for both singular and non-singular functions) by using functional information outside the interval of integration. It is observed empirically that the quadrature weights $(\nu^j_1, \nu^j_2, ..., \nu^j_k)$ are always of $O(1)$.

Remark 4.14 The authors have been unable to construct a quadrature rule which is independent of the number j of points used in the uncorrected trapezoidal rule. However, this is a minor deficiency since the weights in such cases can be precomputed and stored.

4.3.3 A fast multipole method in one-dimension for sums of the form $\sum_{k=1}^{j-1} \frac{\alpha_k}{(y_j^2-x_k^2)^{1/2}}$

In this section we consider the problem of computing the sums

$$f_j = \sum_{k=1}^{j-1} \frac{\alpha_k}{\sqrt{y_j^2 - x_k^2}} \tag{4.46}$$

for $j = 1, \ldots, N$, where $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_N\}$, $\{\alpha_1, \ldots, \alpha_N\}$, and $\{f_1, \ldots, f_N\}$ are sets of real numbers.

Remark 4.15 For the remainder of this section, we shall assume without loss of generality that $x_i, y_i \in [-1, 1]$ for $i = 1, \ldots, N$.

Remark 4.16 Sums of the form (4.46) are a simple reformulation of the trapezoidal approximation to the Hankel transform, defined in (4.43).

The fast multipole algorithm of [13] computes sums of a slightly different form than (4.46) in $O(N)$ arithmetic operations, described described by the formulae

$$f_j = \sum_{k=1}^{N} \frac{\alpha_k}{w_j - z_k} \tag{4.47}$$

for $j = 1, \ldots, N$, where $\{z_1, \ldots, z_N\}$ and $\{w_1, \ldots, w_N\}$ are sets of complex numbers. From a physical viewpoint, this corresponds to the evaluation of the electrostatic field due to N charges which lie in the plane. The two and three dimensional scenarios for the N-body problem have been discussed in some depth (see, for example, [13]). In recent years the the analysis and applications of one dimensional problems have been invesigated in [9], [10], [4]. In [9] the sums

$$f_j = \sum_{k=1}^{N} \frac{\alpha_k}{y_j - x_k} \tag{4.48}$$

for $j = 1, 2, \ldots, N$, are computed in $O(N)$ using a combination of chebyshev approximation techniques, singular value decompositon (SVD) based compression. In [4], an algorithm is constructed to evalauate the function f

$$f(t) = \sum_{j=0}^{n-1} \alpha_j \cdot P_j(t), \tag{4.49}$$

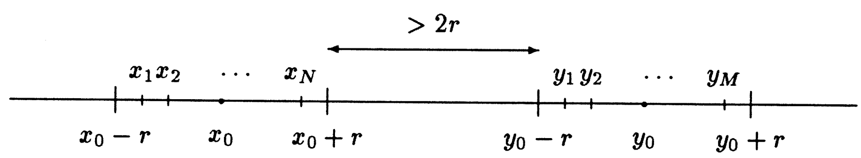

Figure 4.1: Well-separated intervals on the line.

at the nodes $t_0, t_1, ..., t_{n-1}$, in expending CPU time proportional to $O(N)$ operations. In fact, an early implementation of the FBT used a version of the algorithm developed in [4].

In this section we briefly describe an $O(N)$ algorithm for the computation of (4.46) which is based on the one-dimensional FMM of [9], and [4]. We assume that the reader is familiar with [13].

General strategy

We will illustrate by means of a simple example how Chebyshev expansions can be used to evaluate expressions of the form (4.46) more efficiently. We will also give an informal description of how the method of this simple example is used in the construction of a fast algorithm for the general case.

First we introduce a definition which formalizes the notion of well-separated intervals on the real line. This is simply the one-dimensional analog of the definition of well-separatedness in [13].

Definition 4.6 *Let* $\{x_1, \ldots, x_N\}$ *and* $\{y_1, \ldots, y_M\}$ *be two sets of points in* $\mathbf{R}$. *We say that the sets* $\{x_i\}$ *and* $\{y_i\}$ *are* well-separated *if there exist points* $x_0, y_0 \in \mathbf{R}$ *and a real* $r > 0$ *such that*

$$\begin{aligned} &|x_i - x_0| < r \quad \forall \;\; i = 1, \ldots, N, \\ &|y_i - y_0| < r \quad \forall \;\; i = 1, \ldots, M, \;\; \textit{and} \\ &|x_0 - y_0| > 4r. \end{aligned} \tag{4.50}$$

Suppose now that $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_M\}$ are well-separated sets of points in $\mathbf{R}$ (see Figure 4.1), that $\{\alpha_1, \ldots, \alpha_N\}$ is a set of complex numbers, and that we wish to compute the numbers $f(y_1), \ldots, f(y_M)$ where the function $f : \mathbf{R} \to \mathbf{C}$ is defined by the formula

$$f(x) = \sum_{k=1}^{N} \frac{\alpha_k}{\sqrt{x^2 - x_k^2}}. \tag{4.51}$$

A direct evaluation of (4.51) at the points $\{y_1, \ldots, y_M\}$ requires $O(NM)$ arithmetic operations. We will describe two different ways of speeding up this calculation based on the following two observations. The observations follow from a combination of Lemma 4.2, Lemma 4.3, and the triangle inequality.

Observation 4.17 *Suppose that $p > 2$ is an integer. Further, suppose that we define the real coefficients Φ_m (representing the far field) by the formula*

$$\Phi_m = \sum_{k=1}^{N} \alpha_k \cdot u_m(x_k), \tag{4.52}$$

for all $m = 1, 2, ..., p$, where u_m is defined in (4.11). Finally, suppose that

$$\tilde{f}_1(y_j) = \sum_{m=1}^{p} \frac{1}{\sqrt{y_j^2 - t_m^2}} \Phi_m \tag{4.53}$$

for all $j = 1, 2, ..., M$, where $t_1, t_2, ..., t_p$ are the Chebyshev coefficients defined in (4.10). Then,

$$| f(y_j) - \tilde{f}_1(y_j) | < O(\frac{1}{5^p}). \tag{4.54}$$

for all $j = 1, 2, ..., M$.

Computation of the coefficients Φ_j requires $O(Np)$ operations, and a subsequent evaluation of $\tilde{f}_1(y_1), \ldots, \tilde{f}_1(y_M)$ is an $O(Mp)$ procedure. The total computational cost of approximating (4.51) to a relative precision $1/5^p$ is then $O(Np + Mp)$ operations.

Observation 4.18 *Suppose that $p > 2$ is an integer. Further, suppose that we define the real coefficients Ψ_m (representing the local expansion) by the formula*

$$\Psi_j = \sum_{k=1}^{N} \frac{\alpha_k}{\sqrt{t_j^2 - x_k^2}} \tag{4.55}$$

for all $m = 1, 2, ..., p$, where $t_1, t_2, ..., t_p$ are the Chebyshev coefficients defined in (4.10). Finally, suppose that

$$\tilde{f}_2(y_j) = \sum_{m=1}^{p} \Psi_m u_m(j) \tag{4.56}$$

for all $j = 1, 2, ..., N$, where u_m is defined in (4.11). Then,

$$| f(y_j) - \tilde{f}_2(y_j) | < O(\frac{1}{5^p}). \tag{4.57}$$

for all $j = 1, 2, ..., M$.

Computation of the coefficients Ψ_j requires $O(Np)$ operations, and a subsequent evaluation of $\tilde{f}_2(y_1), \ldots, \tilde{f}_2(y_M)$ is an $O(Mp)$ procedure. Again the total computational cost of approximating (4.51) to a relative precision $1/5^p$ is $O(Np + Mp)$ operations.

Consider now the general case, where the points $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_M\}$ are arbitrarily distributed on the interval $[-1, 1]$ (see Remark 4.15). We use a hierarchy of grids to subdivide the computational domain $[-1, 1]$ into progressively smaller subintervals, and to subdivide the sets $\{x_i\}$ and $\{y_i\}$ according to subinterval (see Figure 4.2). A tree structure is imposed on this hierarchy, so that the two subintervals resulting from the bisection of a larger (parent) interval are referred to as its children. Two Chebyshev expansions are associated with each subinterval: a far-field expansion for the points within the subinterval, and a local expansion for the points which are well-separated from the subinterval. Interactions between pairs of well-separated subintervals can be computed via these Chebyshev expansions in the manner described above, and all other interactions at the finest level can be computed directly. Once the precision has been fixed, the computational cost of the entire procedure is $O(N)$ operations. We refer the reader to the algortithm of [9], or [4] for a detailed description.

A more efficient algorithm

Chebyshev expansions are not the most efficient means of representing interactions between well-separated intervals. All the matrix operators of the algorithm are numerically rank-deficient, and can be further compressed by a suitable change of basis. The orthogonal matrices required for this basis change are obtained via singular value decompositions

(SVDs) of appropriate matrices. We describe briefly below how SVDs can be used to make a more efficient algorithm for the evaluation of sums of the form (4.46).

Suppose that we define the matrix

$$M_{j,m} = \frac{1}{\sqrt{y_j^2 - t_m^2}} \tag{4.58}$$

for all $j = 1, 2, ..., M$, and all $m = 1, 2, 3, ..., p$. Then, the SVD of the matrix is

$$M_{j,m} = U_{j,m} \cdot \Sigma_{m,m} \cdot V_{m,m}^T \tag{4.59}$$

Now, it is observed emperically that the numerical rank of the matrix $M_{j,m}$ is $\bar{p}$ where $\bar{p} \approx \frac{p}{2}$. This observation provides us with a tool to compress the far field expansion Φ_m. We define the matrix

$$\bar{M}_{j,\bar{m}} = U_{j,\bar{m}} \cdot \Sigma_{\bar{m},\bar{m}} \cdot V_{\bar{m},\bar{m}} \tag{4.60}$$

for all $j = 1, 2, ..., M$, and all $\bar{m} = 1, 2, ..., \bar{p}$. and, the real coefficients (representing the compressed far field expansion) by $\bar{\Phi}_m$,

$$\bar{\Phi}_m = \Sigma_{\bar{m},\bar{m}} \cdot V_{\bar{m},\bar{m}} \cdot \Phi_m, \tag{4.61}$$

Now, obviously

$$\bar{f_1}_j \approx U_{j,\bar{m}} \cdot \bar{\Phi}_m. \tag{4.62}$$

for all $j = 1, 2, ...M$.

Suppose that we define the matrix

$$M_{j,k} = \frac{1}{\sqrt{t_j^2 - x_k^2}} \tag{4.63}$$

for all $j = 1, 2, ..., N$, and $k = 1, 2, 3, ..., p$. Obviously

$$\Psi_j = M_{j,k} \cdot \alpha_k = U_{j,k} \cdot \Sigma_{k,k} \cdot V_{m,m}^T \cdot \alpha_k \tag{4.64}$$

for all $j = 1, 2, ...m$, and, if the numerical rank of the matrix $M_{j,k}$ is $\bar{p}$, then

$$\Psi_{\bar{j}} = K_{\bar{j},k} \cdot \alpha_k = U_{\bar{j},k} \cdot \Sigma_{k,k} \cdot V_{k,k}^T \cdot \alpha_k \tag{4.65}$$

where $j = 1, 2, ..., \bar{p}$, then the local representation of potential is

$$\bar{\Psi}_{\bar{j}} = U_{\bar{j},k} \cdot \Phi_j \tag{4.66}$$

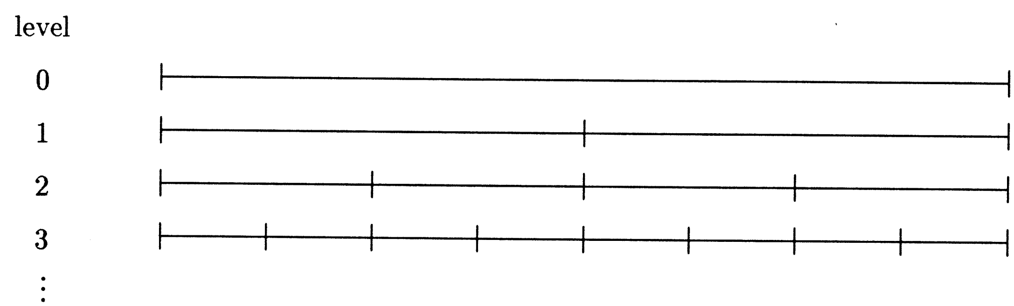

Figure 4.2: Hierarchy of subintervals.

4.4 The fast Hankel transform

4.4.1 Informal description of the algorithm

In this section we outline the procedure we use to rapidly evaluate integrals of the form

$$g(a) = \int_0^A f(x)J_0(ax)dx. \tag{4.67}$$

More specifically, suppose that $h = A/(N-1)$, $x_i = ih$, $a_j = \frac{\pi j}{Nh}$, $u_i = \frac{\pi i}{Nh}$, and, $f : [0, A] \to \mathbf{R}$ is a function tabulated at N $x_0, x_1, ..., x_{N-1}$. Then

$$g(a_j) = \int_0^A f(x)J_0(a_j x)dx \tag{4.68}$$

is computed for all $j = 0, 1, 2..., N-1$, in $O(N \log N)$ operations.

We use the (4.5) to decompose the right-hand side of equation (4.67) into two separate integrals, each of which is computed independently. If we define

$$F(u) = \int_0^A f(x)\cos(ux)dx, \tag{4.69}$$

then (4.67) is equivalent to

$$g(a) = \int_{-a}^{a} \frac{F(u)}{\sqrt{a^2 - u^2}} du. \tag{4.70}$$

Now, given a function f tabulated at n points, we evaluate the integrals $F(u_i)$, for all $i = 0, 1, 2..., N-1$, using a fast cosine transform in $O(N \log N)$ operations. Subsequently, using a high order corrected trapezoidal rule (developed in [18]), we evaluate the integrals $g(a_i)$ in two stages. First, a trapezoidal approximation to the integral is made using the formula

$$g_T(a_j) = h \sum_{l=-(j-1)}^{j-1} \frac{F(u_l)}{\sqrt{a_j^2 - u_l^2}}. \tag{4.71}$$

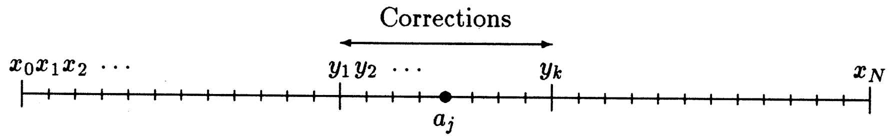

Figure 4.3: Correction nodes

The trapezoidal approximation g_T is subsequently corrected with appropriately chosen quadrature weights $\nu_1^j, \nu_2^j, ..., \nu_k^j$ using the formula

$$g_C(a_j) = h \sum_{i=1}^{k} \nu_i^j \frac{F(y_i)}{\sqrt{| a_j^2 - y_i^2 |}}, \tag{4.72}$$

for all $j = 0, 1, 2, ..., N-1$, where $y_i = a_j - hi$ for $1 \le i \le k/2$, and $y_i = a_j + h(i - k/2)$ for $k/2 + 1 \le i \le k$. In Section 4.3.2 we prove that the corrected rule $g_T + g_C$ is of order $2k - 2$, i.e., for all j there exists some real $c > 0$ such that

$$| (g_T(a_j) + g_C(a_j)) - g(a_j) | \le \frac{c}{j^{2k-2}}. \tag{4.73}$$

The trapezoidal approximation g_T is rapidly computed in $O(N)$ operations, using a generalized version of the 1-D FMM for the smoothly varying kernel $\frac{1}{\sqrt{a^2-u^2}}$. The correction to the trapezoidal approximation g_C is computed in $O(kN)$ operations using (4.72). Thus, the total asymptotic time complexity of the algorithm is $O(N \log N) + O(N) + O(pN)$.

4.4.2 Detailed description of the algorithm, and complexity analyis

This section contains the complexity analysis of the algorithm for the fast Hankel transform. More specifically, suppose that $f : [0, A] \to \mathbf{R}$. Further, suppose that $h = A/(N-1)$, $x_i = ih$, $a_j = \frac{\pi j}{Nh}$, $u_i = \frac{\pi i}{Nh}$, and f is tabulated at N points $x_0, x_1, ..., x_{N-1}$. Then

$$g(a_j) = \int_0^A f(x) J_0(a_j x) dx \tag{4.74}$$

is calculated for all $j = 0, 1, 2..., N-1$, in $O(N \log N)$ operations. The function is assumed to be tabulated at the Nyquist sampling rate.

Algorithm 4.1

Step **Complexity** **Description**

1 **Comment** [Preprocessing: Input Problem size N. Compute the correction coefficients β_k^m using the formula (2.52). Compute the correction weights ν_p^j by solving the system of equations (4.30), (4.31). These correction weights are obtained once and are stored.]

2 $O(N \log N)$ **Comment** [The fast cosine transform of an even or odd function, ie, for a function f tabulated at n points, $f(x_0)$, $f(x_1)$, ..., $f(x_{N-1})$, compute the integral $F_C(a_i) \approx \int_0^A f(x)\cos(a_i x)dx$ for all $i = 0, 1, ..., N-1$.]

If the function f is even,

$$F_C(a_i) \quad = \quad h(\sum_{j=0}^{N-1} f(x_j)\cos(\frac{\pi ij}{N})),$$

is computed using a fast cosine transform.

If the function f is odd,

The function is interpolated to a finer grid of with twice the number of nodes, using a two fast sine transforms in $O(2N \log N)$ operations. Subsequently, $\hat{f}$ is calculated using the formula (4.22) in $O(1N)$ operations. Finally,

$$F_C(a_i) \quad = \quad h(\sum_{j=0}^{N-1} \hat{f}(x_j)\cos(\frac{\pi ij}{N}))$$

is computed using a fast cosine transform.

3 $O(N)$ **Comment** [The trapezoidal approximation to the to the integral (4.74) is computed in $O(N)$ operations using a 1-D FMM.]

$$g_T(a_j) = h \sum_{l=-(j-1)}^{j-1} \frac{F(u_l)}{\sqrt{a_j^2 - u_l^2}}.$$

is computed for all $j = 0, 1, ..., N-1$, using an SVD based 1-D FMM (see, section 4.3.3).

4 $O(N)$ **Comment** [The trapezoidal approximation is corrected using an end-point corrected trapezoidal rule.]

$$g_C(a_j) = h \sum_{i=1}^{k} \nu_i^j \frac{F(y_i)}{\sqrt{a_j^2 - y_i^2}},$$

for all $j = \pm 1, \pm 2, ..., \pm \frac{N-1}{2}$, where $y_i = a_j - hi$ for $1 \leq i \leq k/2$, and $y_i = a_j + h(i - k/2)$ for $k/2 + 1 \leq i \leq k$.

4.5 Numerical Results

In this section we present numerical experiments testing the algorithms of this paper.

We consider the following integral (of an odd function),

$$\int_0^{2\pi} sin(bx) \cdot cos(bx) \cdot e^{-x^2}. \tag{4.75}$$

to illustrate the effectiveness of using the correction coefficients β^m (see, 2.52) for correcting the cosine transform of an odd function (see, (2.52)). In Table 1 we present convergence results of the standard trapezoidal rule, while in Table 2 we present convergence results using these correction coefficients. In both tables the first column contains the number of nodes discretizing the interval $[0 : 2\pi]$. In Table 1, columns 2-6 contain the relative errors of the standard trapezoidal rule used to evaluate the integral (4.75) for various values of b. In Table 2, columns 2-6 contain the relative errors of the corrected trapezoidal rule (using the mapping $P_{N,2N-3}$ defined in (4.22)) to also evaluate the integral (4.75) for various values of b. We observe empirically that, an odd function sampled at 4 points per wavelength, can be integrated to double precision accuracy using the correted trapezoidal rule for non-singular functions. Hence, it is possible to accurately evaluate the cosine transform of an odd function if the function is sampled at twice the Nyquist sampling rate (see, Step 2 of Algorithm 4.1).

The quadrature weights for the rule $T^j_{\nu j}$ (see, (4.26)) are obtained as solutions of linear systems (4.30), (4.31). The linear systems used for determining these weights are very ill-conditioned. In order to combat the high condition number, all systems were solved using the mathematical package Mathematica using 100 significant digits. We consider the following integral

$$\int_{-\pi}^{\pi} \frac{\cos(b \cdot u)}{\sqrt{\pi^2 - u^2}} du, \tag{4.76}$$

to experimentally demostrate the convergence rate of the quadrature rule $T^j_{\nu j}$. In Table 3 convergence results are presented for the rule rule $T^j_{\nu j}$ using 20 correction weights $(\nu^j_1, \nu^j_2, ..., \nu^j_{20})$. The first column contains the number of nodes discretizing the interval $[0 : 2\pi]$. Columns 2-4 contain the relative errors of rule to evaluate the integral (4.76) for various values of b. It can be observed in Table 3 that the rule $T^j_{\nu j}$ provides single precision accuracy for a function tabulated at twice the Nyquist sampling rate, and double precision accuracy for function tabulated at four times the Nyquist sampling rate.

We have written a computer program in ANSI FORTRAN for the implementation of the algorithm of this paper. This program was tested on a Sun SPARCstation 10 for a variety of input data. Four experiments are described below, and their results are summarized in Tables 4-7. These tables contain error estimates and CPU time requirements for the algorithms, with all computations performed in double precision arithmetic.

The table entries are described below

- The first column in each table contains the problem size N, which was chosen to be a power of 2, ranging from 32 to 1024.
- The second column contains the time required for the fast cosine transform. In Examples 3, and 4 this includes the time to interpolate to a finer grid.
- The third column contains the time required for the one-dimensional FMM.

- The fourth column contains the time required for the correction to the trapezoidal approximation to the Hankel transform.
- The fifth column contains the time required for the the direct implementation of the Hankel transform.
- The sixth column contains the time required for an FFT of the same size.
- The seventh column contains the relative 2-norm E_2 for each result.

Two technical details of our implementations appear worth mentioning here:

- The implementation consists of two main subroutines: the first is an initialization stage in which the elements of the various matrices employed by the algorithms are stored on disk, and the second is the evaluation stage in which these matrices are applied. Successive applications of the linear transformations to multiple vectors requires the initialization to be performed once.
- The parameters for the algorithm were chosen to retain maximum precision while minimizing the CPU time requirements. We found that by using 20 quadrature weights for the correction of the trapezoidal approximation (see, 4.43) we minimized the CPU time of the algorithm without sacrificing accuracy.

Following are the descriptions of experiments, and tables of numerical results.

Example 1. The Hankel transform for an even function The purpose of this example is to demonstrate the performance of the fast Hankel transform for evaluating the expression

$$g(a_j) = \int_0^{2\pi} f(x) J_0(a_j x) dx \tag{4.77}$$

where

$$f(x) = (\cos(b \cdot x) + \cos(\frac{b \cdot x}{2}) + \cos(\frac{b \cdot x}{3})) \cdot e^{-x^2}, \tag{4.78}$$

and $b = \frac{N}{4}$. The function f is tabulated at the equispaced nodes $x_0, x_1, ..., x_{N-1}$, where $h = \frac{2\pi}{N-1}$, and, $x_i = ih$. The Hankel transform (4.77) is computed for all $a_j = \frac{\pi j}{Nh}$ for all

$j = 0, 1, ..., N-1$. The single precision results are provided in Table 4. The algorithm is easily modified to provide double precision results, merely by zero padding the input vector by a vector of the same size (see, Table 3). The double precision results are are provided in Table 5.

Example 2. The Hankel transform for an odd function The purpose of this example is to demonstrate the performance of the fast Hankel transform for evaluating the expression

$$g(a_j) = \int_0^{2\pi} f(x) J_0(a_j x) dx \tag{4.79}$$

where

$$f(x) = x \cdot ((\cos(b \cdot x) + \cos(\frac{b \cdot x}{2}) + \cos(\frac{b \cdot x}{3})) \cdot e^{-x^2}), \tag{4.80}$$

and $b = \frac{N}{4}$. The function f is tabulated at the equispaced nodes $x_0, x_1, ..., x_{N-1}$, where $h = \frac{2\pi}{N-1}$, and, $x_i = ih$. The Hankel transform (4.77) is computed for all $a_j = \frac{\pi j}{Nh}$ for all $j = 0, 1, ..., N-1$. The single precision results are provided in Table 4. The algorithm is easily modified to provide double precision results, merely by zero padding the input vector by a vector of the same size (see, Table 3). The double precision results are are provided in Table 5. We observe that if a function g is defined as follows

$$\begin{aligned} g(u,v) &= (\cos(b \cdot \sqrt{(u^2+v^2)}) + \cos(\frac{b \cdot \sqrt{(u^2+v^2)}}{2}) \\ &+ \cos(\frac{b \cdot \sqrt{(u^2+v^2)}}{3})) \cdot e^{-(u^2+v^2)}, \end{aligned} \tag{4.81}$$

(i.e., the function is rotationally symmetric) then the Hankel transform may be used to compute the two-dimensional FFT (see,)

$$g(a_k, a_j) = \int_{-2\pi}^{2\pi} \int_{-2\pi}^{2\pi} g(u,v) \cdot e^{ia_k} \cdot e^{ia_j} \tag{4.82}$$

for all $k = 0, \pm 1, \pm 2, ..., \pm \frac{N-1}{2}$, and all $j = 0, \pm 1, \pm 2, ..., \pm \frac{N-1}{2}$. Column 10 contains the time required by a 2-D FFT to evaluate the integrals (4.82) Column 11 contains the ratio of the time taken by the 2-D FFT to the time taken by Algorithm 4.1.

Table 4.1: Accuracy of the uncorrected trapezoidal rule for evaluating integrals $\int_0^{2\pi} sin(bx) \cdot cos(bx) \cdot e^{-x^2}$

N	b=4	b=8	b=16	b=32	b=64
32	0.223E+00	0.108E+01	0.124E+02	0.330E+02	0.358E+02
64	0.519E-01	0.220E+00	0.104E+01	0.261E+02	0.677E+02
128	0.127E-01	0.524E-01	0.218E+00	0.102E+01	0.533E+02
256	0.313E-02	0.129E-01	0.523E-01	0.216E+00	0.101E+01
512	0.779E-03	0.320E-02	0.129E-01	0.521E-01	0.215E+00

Table 4.2: Accuracy of the corrected trapezoidal rule for evaluating the integrals $\int_0^{2\pi} sin(bx) \cdot cos(bx) \cdot e^{-x^2}$

N	b=4	b=8	b=16	b=32	b=64
32	0.359E-04	0.125E+01	0.126E+02	0.334E+02	0.376E+02
64	0.730E-14	0.346E-11	0.119E+01	0.262E+02	0.679E+02
128	0.708E-14	0.705E-14	0.621E-14	0.115E+01	0.533E+02
256	0.665E-14	0.705E-14	0.554E-14	0.355E-14	0.111E+01
512	0.644E-14	0.749E-14	0.598E-14	0.599E-14	0.133E-13

Table 4.3: Accuracy of the quadrature rule $T^j_{\nu_j}$ for evaluating the integrals $\int_{-\pi}^{\pi} \cos(b \cdot u)/(\pi^2 - u^2)^{\frac{1}{2}} du$

N	b=N/2	b=N/4	b=N/8
32	0.165E-01	0.829E-08	0.897E-15
64	0.109E-02	0.334E-07	0.868E-14
128	0.671E-02	0.668E-08	0.298E-13
256	0.784E-02	0.175E-07	0.107E-13
512	0.798E-02	0.242E-07	0.361E-13
1024	0.795E-02	0.257E-07	0.597E-14

Table 4.4: Numerical results for Example 1 (single precision)

N	t_{cos}	t_{fmm}	t_{corr}	t_{alg}	t_{dir}	t_{fft}	t_{alg}/t_{fft}	E_2 error
32	0.00011	0.00016	0.00016	0.00043	0.00050	0.00011	3.839	0.33351E-07
64	0.00018	0.00048	0.00034	0.00100	0.00142	0.00019	5.319	0.33688E-06
128	0.00040	0.00116	0.00080	0.00236	0.00464	0.00040	5.900	0.22484E-06
256	0.00070	0.00280	0.00170	0.00520	0.01660	0.00082	6.341	0.19274E-06
512	0.00200	0.00600	0.00350	0.01150	0.05900	0.00260	4.423	0.32146E-06
1024	0.00400	0.01200	0.00700	0.02300	0.22900	0.00760	3.026	0.62221E-06

Table 4.5: Numerical results for Example 1 (double precision)

N	t_{cos}	t_{fmm}	t_{corr}	t_{alg}	t_{dir}	t_{fft}	t_{alg}/t_{fft}	E_2 error
64	0.00020	0.00065	0.00030	0.00115	0.00140	0.00020	5.750	0.27881E-13
128	0.00030	0.00190	0.00080	0.00300	0.00470	0.00038	7.895	0.12463E-12
256	0.00060	0.00500	0.00160	0.00720	0.01660	0.00084	8.571	0.13597E-12
512	0.00150	0.01350	0.00350	0.01850	0.06300	0.00280	6.607	0.19605E-12
1024	0.00417	0.03333	0.00667	0.04417	0.23917	0.00833	5.300	0.26481E-12

Table 4.6: Numerical results for Example 2 (single precision)

N	t_{cos}	t_{fmm}	t_{corr}	t_{alg}	t_{dir}	t_{fft}	t_{alg}/t_{fft}	E_2 error	t_{2Dfft}	t_{2Dfft}/t_{alg}
32	0.00046	0.00016	0.00016	0.00078	0.00086	0.00011	7.091	0.76613E-07	0.00335	4.295
64	0.00082	0.00050	0.00038	0.00170	0.00214	0.00021	8.173	0.33767E-06	0.01332	7.835
128	0.00160	0.00116	0.00080	0.00356	0.00588	0.00040	8.900	0.17319E-05	0.05268	14.798
256	0.00350	0.00270	0.00170	0.00790	0.01890	0.00086	9.186	0.18528E-06	0.22000	27.848
512	0.00850	0.00550	0.00350	0.01750	0.06800	0.00260	6.731	0.45411E-06	1.42000	81.143
1024	0.02000	0.01300	0.00700	0.04000	0.25200	0.00740	5.405	0.6342E-06	8.58200	214.550

Table 4.7: Numerical results for Example 2 (double precision)

N	t_{cos}	t_{fmm}	t_{corr}	t_{alg}	t_{dir}	t_{fft}	t_{alg}/t_{fft}	E_2 error	t_{2Dfft}	t_{2Dfft}/t_{alg}
64	0.00080	0.00065	0.00035	0.00180	0.00205	0.00019	9.474	0.10525E-13	0.01265	7.028
128	0.00160	0.00190	0.00080	0.00430	0.00590	0.00038	11.316	0.85742E-13	0.05190	12.070
256	0.00340	0.00460	0.00180	0.00980	0.01880	0.00084	11.667	0.10078E-12	0.21880	22.327
512	0.00750	0.01400	0.00350	0.02500	0.06700	0.00280	8.929	0.90009E-12	1.55300	62.120
1024	0.02250	0.03167	0.00750	0.06167	0.25583	0.00783	7.872	0.54210E-12	8.72167	141.432

4.6 Generalizations and Conclusions

An algorithm has been presented for the rapid evaluation of the Hankel transform. The algorithm is based on a combination of high-order quadratures for singular functions, and a version of the one-dimensional fast multipole method. The algorithm can be applied to a number of situations.

1. It is is well known (see, for example [23]) that a function $f \in c^k(0,a)$ can be expanded in a series

$$f(x) = \sum_{r=1}^{\infty} A_r J_n(\lambda_r x), \tag{4.83}$$

where $n > -1$, J_n are the Bessel functions of order n, and $\lambda_1, \lambda_2, ...,$ are the positive zeros of $J_n(\lambda a)$. The coefficients A_r are given by the formula

$$\int_0^a x f(x) J_n(\lambda_r x) dx = \frac{a^2}{2} A_r (J_n'(\lambda_r a))^2. \tag{4.84}$$

Expansions of the form (4.83) are known as Fourier-Bessel expansions. Fourier-Bessel expansions are encountered in many areas of computational physics. Among the problems leading to them are the vibrations of a circular membrane, flow of heat in a circular cylinder, wave propagation in a three-dimensional layered medium and many others.

2. The Fourier Transform of a cylindrically symmetric function can be computed with a single integral instead of a double integral as we show below. We introduce polar coordinates in both the spatial and frequency domains,

$$x = r\cos(\phi) \quad \text{and} \quad y = r\sin(\phi),$$
$$u = \rho\cos(\alpha) \quad \text{and} \quad v = \rho\sin(\alpha),$$

so that $ux + vy = r\rho\cos(\phi - \alpha)$. If, $f(x,y) = g(r)$ then the transform

$$F(u,v) \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-i(ux+vy)} dx dy,$$

is just

$$G(\rho) \;=\; \int_{-\pi}^{\pi}\int_0^{\infty} f(r) e^{-ir\rho\cos(\phi-\alpha)} dr d\phi.$$

If we change the order of integration,

$$\int_{-\pi}^{\pi} e^{-ir\rho\cos(\phi-\alpha)}d\phi \quad = \quad 2\pi J_0(r\rho).$$

Thus, if $F(u,v) = G(\rho)$ then,

$$G(\rho) = 2\pi \int_0^\infty rf(r)J_0(r\rho)dr. \tag{4.85}$$

3. There are a number of applications based on the approach of this chapter to the study of diffraction techinques in optics. For example, the Fraunhofer diffraction formula is of the form

$$I(u_k, u_j, D) \approx |\frac{1}{\lambda D}\int_{\mathbb{R}^2} A(x,y)e^{-i\frac{2\pi}{\lambda D}u_x+u_y}dxdy| \tag{4.86}$$

where $A(x,y)$ is the aperture function. In many situations the aperture function is circular, simply reducing the problem to a Hankel transform.

Chapter 5

Appendix

5.1 Correction weights for Non-singular and Singular functions

5.1.1 Quadrature Weights β_k^m for Non-singular Functions

$$\begin{aligned}\int_a^b f(x)dx &\approx T_{\beta^m}^n(f) \\ &= T_n(f) + h\sum_{k=1}^{\frac{m-1}{2}}(f(b+kh) - f(b-kh) - f(a+kh) + f(a-kh))\beta_k^m.\end{aligned}$$

m = 3	m = 5	m = 7
0.4166666666666667D-01	0.5694444444444444D-01	0.6483961640211640D-01
	-0.7638888888888889D-02	-0.1395502645502646D-01
		0.1579034391534392D-02

m = 9	m = 11	m = 13
0.6965636022927690D-01	0.7289995064734647D-01	0.7523240913673701D-01
-0.1877177028218695D-01	-0.2247873075998076D-01	-0.2539430387171893D-01
0.3643353174603175D-02	0.5728518443362193D-02	0.7672233851187638D-02
-0.3440531305114638D-03	-0.9618798768104324D-03	-0.1739366039940610D-02
	0.7722834328737106D-04	0.2539297439987751D-03
		-0.1767014007114040D-04

m = 15	m = 17	m = 19
0.7699017460749256D-01	0.7836226334784643D-01	0.7946301859082432D-01
-0.2773799116605967D-01	-0.2965891540255508D-01	-0.3126001393779562D-01
0.9429999321943197D-02	0.1100166460634853D-01	0.1240262582468400D-01
-0.2591615965155427D-02	-0.3464763345380610D-02	-0.4326893325894750D-02
0.5202578456284055D-03	0.8560837610996297D-03	0.1240963216686299D-02
-0.6683840498737985D-04	-0.1531936403942661D-03	-0.2763550661820004D-03
0.4097355409686621D-05	0.1753039202853559D-04	0.4447195391960246D-04
	-0.9595026156320693D-06	-0.4581897491741901D-05
		0.2263996797568645D-06

m = 21	m = 23	m = 25
0.8036566134581083D-01	0.8111924751518991D-01	0.8175787507251367D-01
-0.3261397807027540D-01	-0.3377334140778168D-01	-0.3477689899786187D-01
0.1365243887004996D-01	0.1477039637407387D-01	0.1577395396415406D-01
-0.5160102022805384D-02	-0.5955094025666833D-02	-0.6707762218226974D-02
0.1657567565141616D-02	0.2092328816706471D-02	0.2535074812330083D-02
-0.4325816968527443D-03	-0.6167158739860944D-03	-0.8233306719437802D-03
0.8735769567235570D-04	0.1470308086322377D-03	0.2231520499850693D-03
-0.1275061020655204D-04	-0.2710805091870410D-04	-0.4885697701951313D-04
0.1193747238089644D-05	0.3616565358265304D-05	0.8277049522724384D-05
-0.5374153101848776D-07	-0.3101244008783459D-06	-0.1016258365190328D-05
	0.1281914349299291D-07	0.8036239225326941D-07
		-0.3070147670921659D-08

m = 27	m = 29	m = 31
0.8230598039728972D-01	0.8278153337505391D-01	0.8319804338077547D-01
-0.3565386751750355D-01	-0.3642664110637037D-01	-0.3711265758638233D-01
0.1667832775003454D-01	0.1749655860883470D-01	0.1823974312884766D-01
-0.7417074991466566D-02	-0.8083781617155592D-02	-0.8709621212955971D-02
0.2978395295604828D-02	0.3417018075663398D-02	0.3847282797776159D-02
-0.1047324179282599D-02	-0.1284180480514226D-02	-0.1530046036007233D-02
0.3146160654817536D-03	0.4198855326958103D-03	0.5372304569083815D-03
-0.7872277799802227D-04	-0.1170025842576793D-03	-0.1636490137583287D-03
0.1591319181836592D-04	0.2714748278587396D-04	0.4245334246577455D-04
-0.2491841417488210D-05	-0.5092371734040995D-05	-0.9173934315347822D-05
0.2832550619442283D-06	0.7409483976575184D-06	0.1604355866780116D-05
-0.2077714429849625D-07	-0.7838889284981948D-07	-0.2179294939201383D-06
	0.5360956533353892D-08	0.2155763344330162D-07
	-0.1778140387386520D-09	-0.1380750254862090D-08
		0.4296200771869423D-10

m = 33	m = 35	m = 37
0.8356586223906441D-01	0.8389305571446765D-01	0.8418600148964681D-01
-0.3772568901686392D-01	-0.3827675171227989D-01	-0.3877475953008444D-01
0.1891730418359046D-01	0.1953724971593344D-01	0.2010640150771007D-01
-0.9296840793733073D-02	-0.9847903489149048D-02	-0.1036531420894598D-01
0.4266725355474089D-02	0.4673760300951798D-02	0.5067442370362510D-02
-0.1781711570625991D-02	-0.2036550840838121D-02	-0.2292444185955085D-02
0.6648868875120992D-03	0.8011551083894191D-03	0.9444553816549185D-03
-0.2183589125884934D-03	-0.2806529564181254D-03	-0.3499410006344108D-03
0.6214890604463385D-04	0.8640764426675015D-04	0.1152776626902024D-03
-0.1506576957398094D-04	-0.2305218544957479D-04	-0.3336290631509345D-04
0.3044582263334879D-05	0.5240846629123186D-05	0.8369617098659884D-05
-0.4984930776645727D-06	-0.9942016492531560D-06	-0.1790615950589770D-05
0.6348092756603319D-07	0.1529838641028607D-06	0.3199739595444088D-06
-0.5895566545002414D-08	-0.1833269916550451D-07	-0.4643199407153423D-07
0.3550460830740161D-09	0.1604311636472664D-08	0.5253570715177823D-08
-0.1040280251184406D-10	-0.9116340394367584D-10	-0.4346230819394555D-09
	0.2523768794744743D-11	0.2337667781591708D-10
		-0.6133208535638922D-12

m = 39	m = 41	m = 43
0.8444980879146044D-01	0.8468861878191560D-01	0.8490582345073519D-01
-0.3922700061890783D-01	-0.3963949060242128D-01	-0.4001723785254232D-01
0.2063059004248263D-01	0.2111481741443320D-01	0.2156339227395194D-01
-0.1085151806728575D-01	-0.1130884391857241D-01	-0.1173947578371039D-01
0.5447289134690455D-02	0.5813149815719777D-02	0.6165108551649863D-02
-0.2547701211583463D-02	-0.2800989375372994D-02	-0.3051271143145499D-02
0.1093355313271473D-02	0.1246579017292300D-02	0.1403005122150116D-02
-0.4255727119317083D-03	-0.5068750854937796D-03	-0.5931791433463679D-03
0.1487041779510615D-03	0.1865518346092672D-03	0.2286250628124040D-03
-0.4617000028477128D-04	-0.6158941596033656D-04	-0.7968542809071795D-04
0.1259595810865357D-04	0.1806736367095093D-04	0.2490991825775139D-04
-0.2980436293579194D-05	-0.4659163000193157D-05	-0.6921164516490830D-05
0.6019365929090900D-06	0.1042814313838009D-05	0.1691476513364548D-05
-0.1016414607443390D-06	-0.1993926296380813D-06	-0.3590633249061523D-06
0.1395254130438025D-07	0.3190683763180230D-07	0.6517156581265044D-07
-0.1495069020432704D-08	-0.4154964772643378D-08	-0.9908863701222516D-08
0.1172703286200068D-09	0.4227988947523140D-09	0.1227209106806963D-08
-0.5987202298631028D-11	-0.3152674188244618D-10	-0.1188835069924533D-09
0.1492744845851982D-12	0.1531756684278896D-11	0.8447500588821128D-11
	-0.3638111051825521D-13	-0.3914899117784468D-12
		0.8877720031504791D-14

5.1.2 Quadrature Weights γ_j^k for Singular Functions

$$\begin{aligned}\int_0^b f(x)dx &\approx T^n_{\gamma^k\beta^m}(f)\\ &= T^n_{R\beta^m}(f) + h\sum_{j=-k,j\neq 0}^{k}\gamma_j f(x_j)\end{aligned}$$

	$s(x) = log(x)$	$s(x) = x^{\frac{1}{2}}$	$s(x) = x^{-\frac{1}{2}}$
	k= 2		
-1	0.7518812338640025D+00	0.4911169802967502D+00	0.1635135941723353D+01
-2	-0.6032109664493744D+00	-0.3176980828356269D+00	-0.1533115151360971D+01
1	0.1073866830872157D+01	0.7141080571189234D+00	0.2143719446940490D+01
2	-0.7225370982867850D+00	-0.3875269545800468D+00	-0.1745740237302873D+01

k= 4			
-1	0.1420113571035790D+01	0.8951854542876017D+00	0.3192416400365587D+01
-2	-0.3125287797178819D+01	-0.1631355661694529D+01	-0.8349519005997507D+01
-3	0.2592853861401367D+01	0.1216528022899115D+01	0.7653118908743808D+01
-4	-0.7648698789584314D+00	-0.3318968291168987D+00	-0.2415721426013858D+01
1	0.2027726083620572D+01	0.1323278097869649D+01	0.4127731944814846D+01
2	-0.3730238148796624D+01	-0.1996997843341944D+01	-0.9431538570036398D+01
3	0.2914105643150046D+01	0.1392513231112159D+01	0.8285519053356245D+01
4	-0.8344033342739005D+00	-0.3672544720151524D+00	-0.2562007305232722D+01

k= 6			
-1	0.2051970990601252D+01	0.1265469280121926D+01	0.4710262208645700D+01
-2	-0.7407035584542865D+01	-0.3802563634358600D+01	-0.2025763995934342D+02
-3	0.1219590847580216D+02	0.5639024206133662D+01	0.3690977699143199D+02
-4	-0.1064623987147282D+02	-0.4569107975444730D+01	-0.3458675005305701D+02
-5	0.4799117710681772D+01	0.1943368974038607D+01	0.1646218520818186D+02
-6	-0.8837770983721025D+00	-0.3411137981342110D+00	-0.3167334195084358D+01
1	0.2915391987686506D+01	0.1878261417316043D+01	0.6026290938505443D+01
2	-0.8797979464048396D+01	-0.4649333971499730D+01	-0.2274216675280301D+02
3	0.1365562914252423D+02	0.6444550155059975D+01	0.3978973181300623D+02
4	-0.1157975479644601D+02	-0.5048462684259424D+01	-0.3656337403895339D+02
5	0.5130987287355766D+01	0.2104363245869803D+01	0.1720419649716102D+02
6	-0.9342187797694916D+00	-0.3644552148433214D+00	-0.3285178657691059D+01

k= 8			
-1	0.2661829001135098D+01	0.1616169645940613D+01	0.6202998068889192D+01
-2	-0.1336900704886964D+02	-0.6771767050468779D+01	-0.3714709770899691D+02
-3	0.3292331764210170D+02	0.1503196947284841D+02	0.1012860584122768D+03
-4	-0.4773939140223472D+02	-0.2024989176835058D+02	-0.1577736812789053D+03
-5	0.4288580615706955D+02	0.1717995995110646D+02	0.1497778690096803D+03
-6	-0.2359187584186291D+02	-0.9018058251167396D+01	-0.8617211496827355D+02
-7	0.7312948709041004D+01	0.2686335493243228D+01	0.2773685303768452D+02
-8	-0.9817367313018633D+00	-0.3483500116200692D+00	-0.3846246456428401D+01
1	0.3760781014023317D+01	0.2398992474897278D+01	0.7870429343373961D+01
2	-0.1580903864167977D+02	-0.8260181779465771D+01	-0.4150717430533848D+02
3	0.3674321491528176D+02	0.1714292235263991D+02	0.1088399244984859D+03
4	-0.5179306469244793D+02	-0.2233476105127601D+02	-0.1663887812447046D+03
5	0.4575621781632506D+02	0.1857536706216344D+02	0.1562272759566466D+03
6	-0.2489478606121209D+02	-0.9622728690582360D+01	-0.8923488368760573D+02
7	0.7656685336983747D+01	0.2839683305088209D+01	0.2857613653609836D+02
8	-0.1021900172352320D+01	-0.3656611549965858D+00	-0.3947565212882627D+01

k= 10			
-1	0.3256353919777872D+01	0.1953545360705999D+01	0.7677722423353747D+01
-2	-0.2096116396850468D+02	-0.1050311310076629D+02	-0.5894517227637276D+02
-3	0.6872858265408605D+02	0.3105516048922884D+02	0.2140398605114418D+03
-4	-0.1393153744796911D+03	-0.5850644296241638D+02	-0.4662332548976578D+03
-5	0.1874446431742073D+03	0.7437254291687940D+02	0.6631353162140867D+03
-6	-0.1715855846429547D+03	-0.6498918498319249D+02	-0.6351002576675097D+03
-7	0.1061953812152787D+03	0.3866979933460322D+02	0.4083227672169233D+03
-8	-0.4269031893958787D+02	-0.1502289586232686D+02	-0.1696285390723725D+03
-9	0.1009036069527147D+02	0.3445119980743215D+01	0.4126838241810020D+02
-10	-0.1066655310499552D+01	-0.3544413204640886D+00	-0.4476202232026015D+01
1	0.4576078100790908D+01	0.2895451608911961D+01	0.9675787330957780D+01
2	-0.2469045273524281D+02	-0.1277820188943208D+02	-0.6561769910673283D+02
3	0.7648830198138171D+02	0.3534092272477722D+02	0.2294242274362024D+03
4	-0.1508194558089468D+03	-0.6441908403427060D+02	-0.4907643918974356D+03
5	0.1996415730837827D+03	0.8029833065236247D+02	0.6906485447124722D+03
6	-0.1807965537141134D+03	-0.6926226351772149D+02	-0.6568499770824342D+03
7	0.1110467735366555D+03	0.4083390088012690D+02	0.4202275815793937D+03
8	-0.4438764193424203D+02	-0.1575467189373152D+02	-0.1739340651258045D+03
9	0.1044548196545488D+02	0.3593677332216888D+01	0.4219582451243715D+02
10	-0.1100328792904271D+01	-0.3681517162342983D+00	-0.4566454997023116D+01

	$s(x) = x^{\frac{1}{3}}$	$s(x) = x^{-\frac{1}{3}}$	$s(x) = x^{-\frac{9}{10}}$
k= 2			
-1	0.5534091724301567D+00	0.1181425202719417D+01	0.9469239981678674D+01
-2	-0.3866961728429464D+00	-0.1060178333186577D+01	-0.9440762908621185D+01
1	0.8032238407479816D+00	0.1613104391254726D+01	0.1027199835611538D+02
2	-0.4699368403351921D+00	-0.1234351260787565D+01	-0.9800475429172870D+01
k= 4			
-1	0.1020832071388625D+01	0.2282486199885223D+01	0.1887299140127902D+02
-2	-0.1983186102544885D+01	-0.5650813876770368D+01	-0.5533585657332243D+02
-3	0.1533381243224831D+01	0.5015176492677874D+01	0.5446669568113802D+02
-4	-0.4298392181270347D+00	-0.1549698701260824D+01	-0.1798256931439028D+02
1	0.1498331817034082D+01	0.3083643341213459D+01	0.2031702799746152D+02
2	-0.2412699293646369D+01	-0.6532393248536034D+01	-0.5722348457297536D+02
3	0.1747071103625803D+01	0.5514329562635926D+01	0.5565641900092846D+02
4	-0.4738916209550530D+00	-0.1662729769845256D+01	-0.1827122362011895D+02

	k= 6		
-1	0.1453673785846622D+01	0.3345150279872830D+01	0.2823540877500425D+02
-2	-0.4645097217879781D+01	-0.1359274618599862D+02	-0.1374572775395727D+03
-3	0.7134681085431341D+01	0.2395745376553084D+02	0.2701409207262959D+03
-4	-0.5928340311544518D+01	-0.2193610831631847D+02	-0.2668390934875709D+03
-5	0.2571788299099303D+01	0.1025817941642386D+02	0.1321842198719930D+03
-6	-0.4588965281604129D+00	-0.1946039989983884D+01	-0.2624585302793210D+02
1	0.2135823382632839D+01	0.4476264293482232D+01	0.3025125337714398D+02
2	-0.5636852769577275D+01	-0.1561643865091390D+02	-0.1418090842954654D+03
3	0.8109443112743051D+01	0.2622632514046215D+02	0.2756109143783281D+03
4	-0.6522597930471829D+01	-0.2345727234258157D+02	-0.2708086499560764D+03
5	0.2775248991033700D+01	0.1081909216811830D+02	0.1337381187138368D+03
6	-0.4888738991530396D+00	-0.2033859578093758D+01	-0.2650087753598454D+02
	$s(x) = x^{\frac{1}{3}}$	$s(x) = x^{-\frac{1}{3}}$	$s(x) = x^{-\frac{9}{10}}$
	k= 8		
-1	0.1866196808675184D+01	0.4383819645513359D+01	0.3756991225931813D+02
-2	-0.8307135206229368D+01	-0.2478753951112947D+02	-0.2556568200490798D+03
-3	0.1909144688191794D+02	0.6535630997043997D+02	0.7531560640068682D+03
-4	-0.2636153756127239D+02	-0.9943323854718145D+02	-0.1239060653686887D+04
-5	0.2279974850816623D+02	0.9269762740745608D+02	0.1226707735965091D+04
-6	-0.1215894117169713D+02	-0.5255448820422732D+02	-0.7300983324043684D+03
-7	0.3671126621978929D+01	0.1670932578919686D+02	0.2417364444480294D+03
-8	-0.4816958588438264D+00	-0.2292774564229746D+01	-0.3433771074231191D+02
1	0.2736714477854559D+01	0.5819143597164960D+01	0.4011575033483495D+02
2	-0.1004886340865174D+02	-0.2833732911493431D+02	-0.2633133438634569D+03
3	0.2164374286136325D+02	0.7130006863921132D+02	0.7675751360752552D+03
4	-0.2894336017656429D+02	-0.1060508801269256D+03	-0.1256487025699375D+04
5	0.2456068654847544D+02	0.9756101768474124D+02	0.1240341295810031D+04
6	-0.1293400965739323D+02	-0.5482984235409941D+02	-0.7368055436813910D+03
7	0.3870327870200545D+01	0.1732510108672746D+02	0.2436290214530279D+03
8	-0.5044475379801205D+00	-0.2366321397723911D+01	-0.3457193022558553D+02

	k= 10		
-1	0.2264788460960479D+01	0.5405454633516052D+01	0.4688376828974556D+02
-2	-0.1292939169279749D+02	-0.3917131575943378D+02	-0.4098330186123370D+03
-3	0.3957101757244672D+02	0.1375220761115852D+03	0.1609252630383255D+04
-4	-0.7639785056816069D+02	-0.2925218664728695D+03	-0.3705300977598581D+04
-5	0.9898237584503627D+02	0.4085024389395215D+03	0.5500852053333245D+04
-6	-0.8785457780822613D+02	-0.3854463863750540D+03	-0.5454835999290140D+04
-7	0.5297228202020211D+02	0.2447281804852179D+03	0.3611006247703252D+04
-8	-0.2081786990425939D+02	-0.1005754937354166D+03	-0.1538237727142339D+04
-9	0.4823072180507942D+01	0.2423820803863685D+02	0.3825346496620566D+03
-10	-0.5007874588446741D+00	-0.2606987282704575D+01	-0.4230611481851793D+02
1	0.3311620888787288D+01	0.7126578020279918D+01	0.4993063308066224D+02
2	-0.1559126529305869D+02	-0.4460041915515531D+02	-0.4215785445878369D+03
3	0.4475277794263923D+02	0.1496138805763186D+03	0.1638730173971623D+04
4	-0.8371954827689160D+02	-0.3113382992063869D+03	-0.3755162938068600D+04
5	0.1064593309138400D+03	0.4292142311074982D+03	0.5559352058138100D+04
6	-0.9333016061951656D+02	-0.4015725721227424D+03	-0.5502789929350287D+04
7	0.5578209928784346D+02	0.2534431424358416D+03	0.3638060903851112D+04
8	-0.2177894078450347D+02	-0.1036930201058000D+03	-0.1548279140995316D+04
9	0.5020172381264958D+01	0.2490333391795860D+02	0.3847470160018897D+03
10	-0.5191450872697767D+00	-0.2671164050811178D+01	-0.4252574395098780D+02

5.1.3 Quadrature Weights μ_j^k for Singular Functions

$$\begin{aligned}\int_{-b}^{b} f(x)dx &\approx T^n_{\mu_l\beta^m}(f)\\ &= T^n_{R\beta^m}(f) + T^n_{L\beta^m}(f) + h\sum_{j=1}^{l}\mu_j(f(x_j)+f(x_{-j}))\end{aligned}$$

	$s(x) = log(x)$	$s(x) = x^{\frac{1}{2}}$	$s(x) = x^{-\frac{1}{2}}$
	k= 1		
1	0.1825748064736159D+01	0.1205225037415674D+01	0.3778855388663843D+01
2	-0.1325748064736159D+01	-0.7052250374156737D+00	-0.3278855388663843D+01
	k= 2		
1	0.3447839654656362D+01	0.2218463552157251D+01	0.7320148345180434D+01
2	-0.6855525945975443D+01	-0.3628353505036473D+01	-0.1778105757603391D+02
3	0.5506959504551413D+01	0.2609041254011273D+01	0.1593863796210005D+02
4	-0.1599273213232332D+01	-0.6991513011320512D+00	-0.4977728731246580D+01

	k= 3		
1	0.4967362978287758D+01	0.3143730697437969D+01	0.1073655314715114D+02
2	-0.1620501504859126D+02	-0.8451897605858329D+01	-0.4299980671214643D+02
3	0.2585153761832639D+02	0.1208357436119364D+02	0.7669950880443822D+02
4	-0.2222599466791883D+02	-0.9617570659704153D+01	-0.7115012409201039D+02
5	0.9930104998037539D+01	0.4047732219908410D+01	0.3366638170534288D+02
6	-0.1817995878141594D+01	-0.7055690129775324D+00	-0.6452512852775417D+01
	k= 4		
1	0.6422610015158415D+01	0.4015162120837891D+01	0.1407342741226315D+02
2	-0.2917804569054941D+02	-0.1503194882993455D+02	-0.7865427201433540D+02
3	0.6966653255738346D+02	0.3217489182548832D+02	0.2101259829107628D+03
4	-0.9953245609468264D+02	-0.4258465281962659D+02	-0.3241624625236100D+03
5	0.8864202397339461D+02	0.3575532701326990D+02	0.3060051449663269D+03
6	-0.4848666190307500D+02	-0.1864078694174975D+02	-0.1754069986558793D+03
7	0.1496963404602475D+02	0.5526018798331437D+01	0.5631298957378288D+02
8	-0.2003636903654183D+01	-0.7140111666166550D+00	-0.7793811669311027D+01
	k= 5		
1	0.7832432020568779D+01	0.4848996969617959D+01	0.1735350975431153D+02
2	-0.4565161670374749D+02	-0.2328131499019837D+02	-0.1245628713831056D+03
3	0.1452168846354678D+03	0.6639608321400605D+02	0.4434640879476441D+03
4	-0.2901348302886379D+03	-0.1229255269966870D+03	-0.9569976467950934D+03
5	0.3870862162579900D+03	0.1546708735692419D+03	0.1353783860926559D+04
6	-0.3523821383570680D+03	-0.1342514485009140D+03	-0.1291950234749944D+04
7	0.2172421547519342D+03	0.7950370021473013D+02	0.8285503487963169D+03
8	-0.8707796087382989D+02	-0.3077756775605837D+02	-0.3435626041981771D+03
9	0.2053584266072635D+02	0.7038797312960103D+01	0.8346420693053734D+02
10	-0.2166984103403823D+01	-0.7225930366983869D+00	-0.9042657229049132D+01

	$s(x) = x^{\frac{1}{3}}$	$s(x) = x^{-\frac{1}{3}}$
	k= 1	
1	0.1356633013178138D+01	0.2794529593974142D+01
2	-0.8566330131781384D+00	-0.2294529593974142D+01
	k= 2	
1	0.2519163888422707D+01	0.5366129541098682D+01
2	-0.4395885396191253D+01	-0.1218320712530640D+02
3	0.3280452346850634D+01	0.1052950605531380D+02
4	-0.9037308390820877D+00	-0.3212428471106080D+01

k= 3		
1	0.3589497168479460D+01	0.7821414573355062D+01
2	-0.1028194998745706D+02	-0.2920918483691252D+02
3	0.1524412419817439D+02	0.5018377890599299D+02
4	-0.1245093824201635D+02	-0.4539338065890004D+02
5	0.5347037290133003D+01	0.2107727158454216D+02
6	-0.9477704273134525D+00	-0.3979899568077642D+01
k= 4		
1	0.4602911286529744D+01	0.1020296324267832D+02
2	-0.1835599861488110D+02	-0.5312486862606378D+02
3	0.4073518974328119D+02	0.1366563786096513D+03
4	-0.5530489773783668D+02	-0.2054841186741071D+03
5	0.4736043505664168D+02	0.1902586450921973D+03
6	-0.2509295082909035D+02	-0.1073843305583267D+03
7	0.7541454492179474D+01	0.3403442687592432D+02
8	-0.9861433968239469D+00	-0.4659095961953657D+01
k= 5		
1	0.5576409349747767D+01	0.1253203265379597D+02
2	-0.2852065698585619D+02	-0.8377173491458909D+02
3	0.8432379551508595D+02	0.2871359566879038D+03
4	-0.1601173988450523D+03	-0.6038601656792564D+03
5	0.2054417067588763D+03	0.8377166700470196D+03
6	-0.1811847384277427D+03	-0.7870189584977965D+03
7	0.1087543813080456D+03	0.4981713229210595D+03
8	-0.4259681068876286D+02	-0.2042685138412166D+03
9	0.9843244561772901D+01	0.4914154195659545D+02
10	-0.1019932546114451D+01	-0.5278151333515754D+01

5.1.4 Quadrature Weights ρ_j^k for Singular Functions

$$\int_{-b}^{b} f(x)dx \approx T^n_{\rho^p\beta^m}(f)$$
$$= T^n_{R\beta^m}(f) + T^n_{L\beta^m}(f) + h\sum_{j=0}^{l} \rho_j(\phi(x_j) + \phi(x_{-j}))$$

Note: ρ_0 is given for $h = 0.01$, For any other h, the following formula is used to calculate ρ_0.

$$\rho_0 = (-.918938533204672741780 3 + 0.5log(h)) - \sum_{j=1}^{p} \rho_j \tag{5.1}$$

m = 3	m = 5	m = 7
-0.3221523626198730D+01	-0.3191075169140337D+01	-0.3181467102013171D+01
	-0.3044845705839327D-01	-0.4325921322794724D-01
		0.3202689042388491D-02

m = 9	m = 11	m = 13
-0.3176811195217937D+01	-0.3174071153542312D+01	-0.3172268092036274D+01
-0.5024307342079858D-01	-0.5462714010179898D-01	-0.5763224261186158D-01
0.5996233119529027D-02	0.8188266460029230D-02	0.9905467894350714D-02
-0.4655906795234226D-03	-0.1091885919666338D-02	-0.1735836457536894D-02
	0.7828690501786438D-04	0.2213870245446548D-03
		-0.1431001195267904D-04

m = 15	m = 17	m = 19
-0.3170992165916170D+01	-0.3170041916020681D+01	-0.3169306861514305D+01
-0.5981954453204053D-01	-0.6148248184914681D-01	-0.6278924541603596D-01
0.1127253159446256D-01	0.1238115647253341D-01	0.1329589096935583D-01
-0.2343420324253269D-02	-0.2897732763288696D-02	-0.3396678852464559D-02
0.4036621845595672D-03	0.6052303442088134D-03	0.8131245480320894D-03
-0.4745095013720858D-04	-0.9784299004952013D-04	-0.1618104373797589D-03
0.2761744848710794D-05	0.1051436637368180D-04	0.2422167651587582D-04
	-0.5537586803550720D-06	-0.2381400032647607D-05
		0.1142275845182835D-06

m = 21	m = 23	m = 25
-0.3168721384920306D+01	-0.3168244070581230D+01	-0.3167847493678468D+01
-0.6384310328523506D-01	-0.6471094753809971D-01	-0.6543800519316396D-01
0.1406233305604607D-01	0.1471321624569456D-01	0.1527249136497475D-01
-0.3843770069700536D-02	-0.4244313571022683D-02	-0.4603847576274231D-02
0.1019474340602541D-02	0.1219746091263614D-02	0.1411497560731106D-02
-0.2355067918692057D-03	-0.3156154921336350D-03	-0.3995067600256630D-03
0.4387403771306165D-04	0.6890800654569580D-04	0.9851668933111745D-04
-0.6066217757119951D-05	-0.1195656336479857D-04	-0.2018119747186014D-04
0.5477355521032650D-06	0.1529459820049702D-05	0.3260961737325821D-05
-0.2408377597694342D-07	-0.1274231726028842D-06	-0.3871484601943021D-06
	0.5166969831297037D-08	0.2990271150667017D-07
		-0.1124351894335142D-08

m = 27	m = 29	m = 31
-0.3167512774719844D+01	-0.3167226492889164D+01	-0.3166978846772530D+01
-0.6605594788600917D-01	-0.6658761414298577D-01	-0.6704988689403577D-01
0.1575801776649599D-01	0.1618335077207727D-01	0.1655894738230538D-01
-0.4927531843955059D-02	-0.5219948285292188D-02	-0.5485075304276741D-02
0.1593569961301572D-02	0.1765579632676354D-02	0.1927601699833581D-02
-0.4851878897058822D-03	-0.5711927253932730D-03	-0.6564674975812873D-03
0.1318371286512027D-03	0.1680496910458935D-03	0.2064233385304999D-03
-0.3070344146767653D-04	-0.4337783830581833D-04	-0.5799637068090648D-04
0.5891522736279919D-05	0.9512778975749005D-05	0.1416413018600433D-04
-0.8882076980903206D-06	-0.1711220479787840D-05	-0.2924616447680532D-05
0.9822897121976360D-07	0.2413616289062888D-06	0.4941524555505996D-06
-0.7065765782430224D-08	-0.2495734799324587D-07	-0.6540388025633560D-07
0.2475589120039617D-09	0.1678885488869214D-08	0.6345793057687260D-08
	-0.5505102218712507D-10	-0.4007478791366100D-09
		0.1234631631962446D-10

m = 33	m = 35	m = 37
-0.3166762511619708D+01	-0.3166571903740287D+01	-0.3166402692325675D+01
-0.6745551530557688D-01	-0.6781430660801560D-01	-0.6813392816895060D-01
0.1689299430945688D-01	0.1719198706148916D-01	0.1746114206017127D-01
-0.5726331418330602D-02	-0.5946641867196491D-02	-0.6148508116208072D-02
0.2079973982393914D-02	0.2223175774156742D-02	0.2357753273497796D-02
-0.7402722529894703D-03	-0.8221018482825148D-03	-0.9016249160749561D-03
0.2463303649153491D-03	0.2872451625618713D-03	0.3287354588014059D-03
-0.7432197238379932D-04	-0.9211101483880897D-04	-0.1111274006152623D-03
0.1984260034353227D-04	0.2651349126416089D-04	0.3412004557474223D-04
-0.4580836910292848D-05	-0.6715522004894009D-05	-0.9348560035479858D-05
0.8916453665775555D-06	0.1466368276662483D-05	0.2246527693132364D-05
-0.1418448246845963D-06	-0.2695610269256914D-06	-0.4646008810431617D-06
0.1767037741742959D-07	0.4047684210333943D-07	0.8082991536902293D-07
-0.1614096203394717D-08	-0.4759815470416764D-08	-0.1148532768136401D-07
0.9602551109604562D-10	0.4105974377982503D-09	0.1278405465017250D-08
-0.2789306492547372D-11	-0.2308426950559283D-10	-0.1044412720573741D-09
	0.6342175941576707D-12	0.5564945021538352D-11
		-0.1450213949229612D-12

m = 39	m = 41	m = 43
-0.3166251466775081D+01	-0.3166115504658825D+01	-0.3166115504658825D+01
-0.6842046079112800D-01	-0.6867878881201404D-01	-0.6867878881201404D-01
0.1770469478902206D-01	0.1792611880692437D-01	0.1792611880692437D-01
-0.6334072100094389D-02	-0.6505172477564357D-02	-0.6505172477564357D-02
0.2484274171602103D-02	0.2603300521146428D-02	0.2603300521146428D-02
-0.9786376366601862D-03	-0.1053029105125390D-02	-0.1053029105125390D-02
0.3704506824517389D-03	0.4121099047922528D-03	0.4121099047922528D-03
-0.1311507079674222D-03	-0.1519803191376791D-03	-0.1519803191376791D-03
0.4259144483911754D-04	0.5184904980367620D-04	0.5184904980367620D-04
-0.1248611531858182D-04	-0.1612303155465844D-04	-0.1612303155465844D-04
0.3255027605557997D-05	0.4509136652480968D-05	0.4509136652480968D-05
-0.7428077534364396D-06	-0.1119040467513331D-05	-0.1119040467513331D-05
0.1457448522607878D-06	0.2428371655709533D-06	0.2428371655709533D-06
-0.2404950901525398D-07	-0.4528845255185268D-07	-0.4528845255185268D-07
0.3241558798437558D-08	0.7103184896000958D-08	0.7103184896000958D-08
-0.3423992518658962D-09	-0.9102854426840433D-09	-0.9102854426840433D-09
0.2656123735758442D-10	0.9146251630822981D-10	0.9146251630822981D-10
-0.1344809528411308D-11	-0.6753249440965090D-11	-0.6753249440965090D-11
0.3332744815245408D-13	0.3256755515337396D-12	0.3256755515337396D-12
	-0.7693371141612778D-14	-0.7693371141612778D-14

Bibliography

[1] B. ALPERT, *PhD Thesis*, Yale University, 1989.

[2] M. ABRAMOVITZ AND I. STEGUN, *Handbook of Mathematical Functions.* National Bureau of Standards, Washington, DC., 1964.

[3] K. E. ATKINSON, *A Survey of Numerical methods for the Solution of Fredholm Integral Equations of the Second Kind*, SIAM, Philadelphia, PA (1976)

[4] BRADLEY K. ALPERT, VLADIMIR ROKHLIN, *A Fast Algorithm for the Evaluation of Legendre Expansions*, Siam J, Sci. Stat Comput, Vol. 12, No. 1., (1991)

[5] G. BEYLKIN *On Fast Fourier Transforms of Functions with Singularities.* University of Colorado.

[6] BRIGHAM, E. *The fast Fourier transform and its applications.* Prentice Hill, 1988

[7] BRACEWELL, E. *The Fourier transform and its applications.* McGraw Hill, 1988

[8] A. J. CHORIN *Numerical study of slightly viscous flow.* Journal of Fluid Mechanics.

[9] ALOK DUTT, MING GU, VLADIMIR ROKHLIN, *Fast Algorithms for Polynomial Interpolation, Integration and Differention* Technical Report 977, Yale Computer Science Department, New Haven, CT, (1994)

[10] ALOK DUTT, *Ph.D. Thesis, Fast Fourier Transforms for Non-Equispaced Data* Technical Report 981, Yale Computer Science Department, New Haven, CT, (1993)

[11] G. DAHLQUIST, A. BJORK, *Numerical Methods*, Prentice Hall, NJ (1974)

[12] I. S. GRADSHTEYN AND I. M. RYZHIK, *Table of Integrals, Series and Products*, Academic Press Inc., 1980.

[13] L. GREENGARD AND V. ROKHLIN, *A Fast Algorithm for Particle Simulations*, J. Comp. Phys., 73 (1987), pp. 325–348.

[14] JOHN STRAIN, *Locally-Corrected Multidimensional Quadrature Rules for Singular Functions*, Lawrence Berkeley Laboratory, Berkley, CA, (1993)

[15] JOHN STRAIN, *A boundry integral approach to unstable solidification.* Journal of computational physics. (1991). CA, (1993)

[16] M.A JASWON, G.T. SYMM, *Integral equation methods in potential theory*, Academic press, (1976)

[17] KLAUS. B, HORN, P., *Robot Vison*, The MIT press, (1986)

[18] SHARAD KAPUR, VLADIMIR ROKHLIN, *High-Order Corrected Trapezoidal Rules for Singular Functions*, Technical Report 1042, Yale Computer Science Department, New Haven, CT, (1994)

[19] SHARAD KAPUR, VLADIMIR ROKHLIN, *The Fast Bessel Transform*, Technical Report 1045, Yale Computer Science Department, New Haven, CT, (1994)

[20] S. KARLIN AND W.L. STUDDEN, *Tchebycheff Systems with Applications in Analysis and Statistics*, Wiley, NY (1966)

[21] E. KELLER AND E. ISAACSON, *Analysis of numerical methods* , Dover, NY (1994)

[22] J. MA, *Interpolation on Arbitrary Regions of the Complex Plane*, UCLA, LA, CA, (1994)

[23] T. M. MACROBERT, *An Elementary Treatise on Harmonic Functions with Applications.* Pergammon Press, Oxford, London, 1967.

[24] GREGORY MATVIEYNKO, *On the Evaluation of Bessel functions* Technical Report 903, Yale Computer Science Department, CT, (1992)

[25] ALLAN V. OPPENHEIM, ALAN S. WILLSKY, *Signals and Systems.* Prentice-Hall, Inc. Englewood Cliffs, NJ, (1983)

[26] V. ROKHLIN, *End-point corrected trapezoidal quadrature rules for singular functions*, Computers and Mathematics with Applications, 1990.

[27] V. ROKHLIN, A. DUTT, T. HRYCAK, N. YARVIN Personal Communication.

[28] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer Verlag, New York, 1980.

[29] E. SORETS, *Fast Fourier Transforms of Piecewise Constant Functions.* Yale Technical Report YALEU/DCS/RR-986.

[30] J. WALKER, *Fast Fourier Transforms*, CRC Press, Ann Arbor, 1991.