A dissertation presented to the faculty of the Graduate School of Yale University in candidacy for the degree of Doctor of Philosophy.

Sparse Representation of Smooth Linear Operators

Bradley Keith Alpert

Research Report YALEU/DCS/RR-814
August 1990

The author was supported in part by Office of Naval Research grant N00014-86-K-0310, in part by DARPA grant DMS-9012751, and in part by IBM grant P00038437.

Approved for public release: distribution is unlimited.

Keywords: *Wavelets, Fast Numerical Algorithms, Integral Equations, Approximation Theory*

ABSTRACT
Sparse Representation of Smooth Linear Operators

Bradley Keith Alpert
Yale University
1990

A wide variety of problems in differential and integral equations require application and inversion of linear operators. For large-scale physical problems, the size of the problem n is such that the work expended by general algorithms–$O(n^2)$ for application of a transformation and $O(n^3)$ for application of its inverse–is often prohibitive. Several methods have been devised in recent years to reduce these complexities by exploiting the structure of particular problems.

This thesis puts earlier methods into a unified framework by developing new wavelet-like bases for $\mathcal{L}^2[0,1]$, and other spaces, which lead to efficient algorithms for operator application and inversion. It is based on the observation that in many cases of interest, while the matrices involved are dense, their elements vary smoothly as a function of their indices, except along a collection of bands of fixed width. Algorithms are presented for transforming such locally smooth matrices into matrices which are sparse (to a finite but arbitrarily high accuracy). The resulting sparse representations support fast algorithms for matrix application and inversion, requiring variously $O(n)$, $O(n \log n)$, and $O(n \log^2 n)$ operations.

Programs have been developed applying these techniques to the solution of second-kind integral equations with non-oscillatory kernels. Numerical results are presented to demonstrate the effectiveness of the approach.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Integral equations are a well-known mathematical tool for formulating physical problems. Historically they have achieved great popularity among mathematicians and physicists in formalizing boundary-value problems of gravitation, electrostatics, fluid dynamics, and scattering. Another application, of considerable current interest, is semiconductor modelling. While a number of physical problems give rise to integral equation formulations directly, a wide variety of partial differential equations may also be expressed as integral equations.

Integral equation formulations have several advantages (good conditioning, dimensionality reduction, and the ability to treat arbitrary regions) but have had one overriding disadvantage: the high cost of working with the associated dense matrices. For a problem requiring an n-point discretization, the inverse of a dense $n \times n$-matrix must be applied to a vector. Even to apply the matrix itself to a vector requires order $O(n^2)$ operations; application of its inverse by a direct (non-iterative) method requires order $O(n^3)$ operations. If an iterative method is employed, the number of iterations depends on the condition number of the problem and each iteration requires application of the $n \times n$ matrix. For large-scale problems, the resulting costs are often prohibitive.

In recent years a number of algorithms ([2], [8], [9], [15]) have been developed for the fast application of linear operators naturally expressible as dense matrices, the best known of which are the particle simulation algorithms developed by L. Greengard and V. Rokhlin [8]. Each algorithm of this class exploits the special structure of a particular problem by combining

1. interpolation of the function which defines the matrix elements, with

2. a divide-and-conquer strategy.

For example, the particle simulation algorithms are based on the observation that

the electrostatic or gravitational field due to a set of particles varies smoothly in regions separated from the location of the particles (see Fig. 1.1). Such a field can therefore be represented to high precision by an expansion containing only a few terms. This observation is augmented by the construction of a divide-and-conquer scheme in which virtually all particle interactions are computed in this fashion. The result is an algorithm which computes the field due to n particles at n locations in order $O(n)$ operations.

As a second example, the evaluation of Legendre expansions can be accomplished by application of a particular dense upper-triangular matrix to an arbitrary vector. Away from the main diagonal, the elements of the matrix vary smoothly as a function of their indices. Alpert and Rokhlin [2] constructed an algorithm by dividing the matrix into submatrices, each of which is separated from the diagonal, as shown in Fig. 1.2. This division ensures that each submatrix has smoothly varying elements which can be approximated by a low-order polynomial. The error of the approximation decays exponentially in the degree of the polynomial, so in practice any desired degree of accuracy may be obtained. The algorithm for application of the $n \times n$ matrix to a vector then requires $O(n)$ operations, as do the algorithms of [8], [9], [15].

Two deficiencies of these algorithms are apparent. First, though they share a

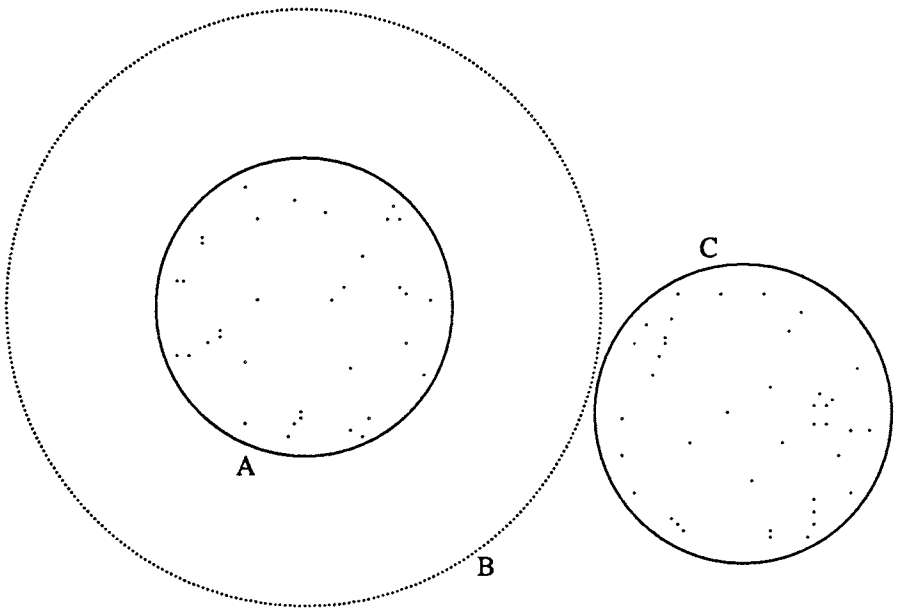

Figure 1.1: *The field due to particles inside circle* A *can be represented by a Laurent expansion about the center of* A. *The expansion converges rapidly outside* B, *which has radius twice that of* A, *so only several terms of the expansion are needed to evaluate (to high precision) the effect of particles inside* A *on particles inside* C.

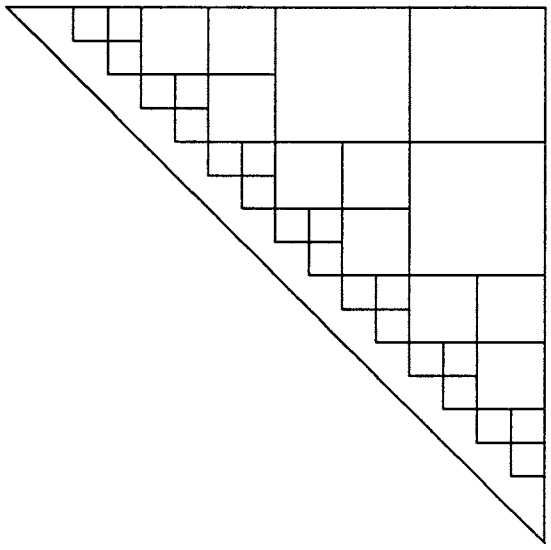

Figure 1.2: *Dense matrix connecting coefficients of Legendre and Chebyshev expansions is divided so that each submatrix can be approximated to high accuracy by a low-order polynomial. This combination of divide-and-conquer and efficient approximation results in an algorithm to apply the $n \times n$ matrix to an arbitrary vector in order $O(n)$ operations.*

common thread, each algorithm is custom-designed for its own problem and is not generally applicable to others. Second, in the case when the matrix is invertible, it is not clear from these algorithms how to efficiently invert the matrix or apply its inverse to a vector. These deficiencies have recently been addressed by Beylkin, Coifman, and Rokhlin in [3].

1.2 Wavelet Bases

1.2.1 Function Space Bases

In this thesis we construct a class of orthonormal bases of $\mathcal{L}^2[0,1]$ (and other spaces), with the property that "smooth" linear operators in these bases correspond to matrices which are sparse, to high precision. This class of operators includes those mentioned above, as well as operators resulting from a wide variety of second-kind integral equations. The matrices are sparse; furthermore, their inverses are sparse, to high precision. As a result, the Schulz method [17], which is an iterative, quadratically convergent technique for matrix inversion, becomes highly effective and produces an order $O(n \log^2 n)$ algorithm for solving a second-kind integral equation.

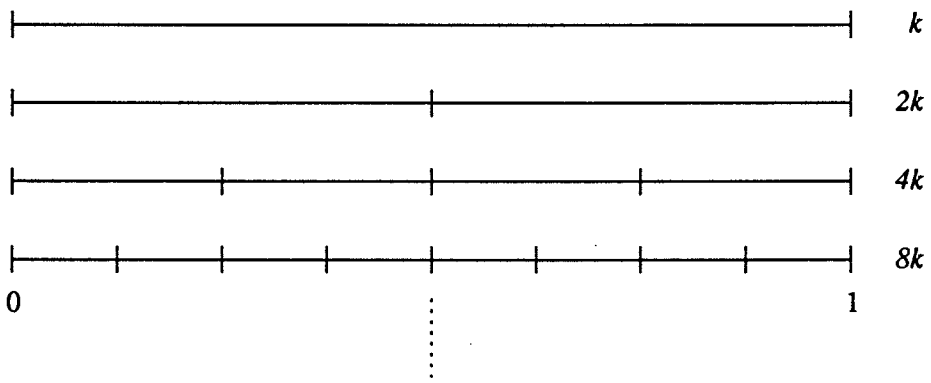

Figure 1.3: *Multi-wavelet bases of* $\mathcal{L}^2[0,1]$ *consist of functions non-zero on intervals of various scales. The basis of order* k *contains* k *functions supported on each interval of one scale, which does not overlap other intervals of that scale. The total number of functions on a single scale is shown.*

A matrix is sparse to high precision if it is a sparse matrix perturbed by a small matrix, *i.e.*, if most of the elements are negligible. In actual computations the smallest elements are discarded, according to a threshold determined by a user-specified accuracy.

What are the properties of a basis of wavelets that lead to sparse representations? The bases we construct are orthonormal and consist of functions that are (generally)

1. orthogonal to low-order polynomials, and

2. non-zero on finite intervals of various lengths.

A basis function $b: [0,1] \to \mathcal{R}$ is orthogonal to low-order polynomials if its first k moments vanish,

$$\int_0^1 b(x)\, x^j\, dx = 0, \qquad j = 0, 1, \ldots, k-1,$$

for some positive integer k. In our construction, all but k basis functions have this property. The property of support on intervals of various lengths is illustrated in Fig. 1.3. These two properties ensure that a function that is smooth, except for a finite number of singularities, will have a negligible projection on most basis functions.

1.2.2 Vector Space Bases

Numerical solution of an integral equation, as mentioned above, can be accomplished by projecting onto an n-dimensional subspace of $\mathcal{L}^2$ and solving the resulting system of n equations in n unknowns. Consider the second-kind integral equation

$$(I - \mathcal{K})f = g,$$

where $\mathcal{K}$ is an integral operator given by the formula

$$(\mathcal{K}f)(x) = \int_0^1 K(x,t)\, f(t)\, dt,$$

and suppose that $\{b_1, \ldots, b_n\}$ is a orthonormal basis for the subspace. Then it is easy to see that the projection is obtained by computing the integrals

$$\int_0^1 \int_0^1 K(x,t)\, b_i(x)\, b_j(t)\, dx\, dt, \qquad i,j = 1, \ldots, n. \tag{1.1}$$

We demonstrate this method of solution, using the new bases, in the next chapter. For many kernels, however, difficulties may arise in obtaining the integrals (1.1).

An alternative general method for solution of integral equations, developed by Nyström, approximates the integral operator by a quadrature, resulting in a matrix $A = \{w_j\, K(x_i, x_j)\}_{i,j=1,\ldots,n}$ for discretization points $\{x_1, \ldots, x_n\} \subset [0,1]$ and quadrature weights $\{w_1, \ldots, w_n\}$. The matrix A is dense, so by direct methods its application to a vector requires order $O(n^2)$ operations and its inversion requires $O(n^3)$ operations.

These time complexities can be cut considerably. Complementary to the function space bases sketched above, we also develop a vector space analogue, for the n-dimensional space of functions defined on $\{x_1, \ldots, x_n\}$. The vector space bases have properties entirely analogous to the function space bases: namely, all but k basis vectors are orthogonal to the moment vectors $\{x_1{}^j, \ldots, x_n{}^j\}$ for $j = 0, \ldots, k-1$, and the basis vectors are non-zero on various "local" subsets of $\{x_1, \ldots, x_n\}$. In solving the integral equation by the Nyström method, the matrix A is similarity-transformed by the vector basis to a sparse matrix in order $O(n \log n)$ operations. In addition, its inverse is sparse, and is obtained in order $O(n \log^2 n)$ operations via Schulz method (as with the $\mathcal{L}^2$ case). Thus the function space and vector space bases are close analogues, and both lead to fast algorithms for solving a wide class of integral equations.

1.3 Thesis Organization

In Chapter 2 we construct the function space bases and demonstrate their completeness and convergence properties. We prove that a variety of integral operators and their inverses, when expanded in these bases, are represented as sparse

matrices. We present analytical expressions for the bases and several numerical examples of their application to the solution of second-kind integral equations.

In Chapter 3 we construct quadratures for operator kernels with certain types of diagonal singularities. The quadratures have the property that they make only endpoint adjustments to the trapezoidal rule, and therefore preserve the smoothness properties of the kernel, yet they achieve high-order convergence.

In Chapter 4 we construct the vector space bases, an algorithm for their computation, and an algorithm to transform a matrix representing an integral operator with a diagonally-singular kernel to sparse form in order $O(n \log n)$ operations. We employ the quadratures of Chapter 3 to obtain the solution of several sample integral equations. We also demonstrate the effectiveness of the new bases with timing results.

Finally, in Chapter 5 we mention some other fast algorithms for the numerical solution of integral equations, we discuss extensions to the methods described in the preceding chapters, and we present further applications for the new bases.

Chapter 2

Function Space Bases

Families of functions $h_{a,b}$,

$$h_{a,b}(x) = |a|^{-1/2}\, h\left(\frac{x-b}{a}\right), \qquad a, b \in \mathcal{R},\ a \neq 0,$$

derived from a single function h by dilation and translation, which form a basis for $\mathcal{L}^2(\mathcal{R})$, are known as *wavelets* (Grossman and Morlet [10]). In recent years, these families have received study by many authors, resulting in constructions with a variety of properties. Meyer [13] constructed orthonormal wavelets for which $h \in C^\infty(\mathcal{R})$. Daubechies [5] constructed compactly supported wavelets with $h \in C^k(\mathcal{R})$ for arbitrary k, and [5] gives an overview and synthesis of the field.

In this chapter we construct a somewhat different type of basis for $\mathcal{L}^2(\mathcal{R})$ that can be readily revised to a basis for $\mathcal{L}^2[0,1]$. Each basis, which we call a *multi-wavelet* basis, is comprised of dilates and translates of a finite set of functions $h_1, \ldots, h_k$. In particular, our bases consist of orthonormal functions

$$h_{j,m}^n(x) = 2^{m/2}\, h_j(2^m x - n), \qquad j = 1, \ldots, k;\ m, n \in \mathcal{Z}, \tag{2.1}$$

where $h_1, \ldots, h_k$ are piecewise polynomial, are supported on the interval $[0,1]$, and have vanishing moments,

$$\int_0^1 h_j(x)\, x^i\, dx = 0, \qquad i = 0, 1, \ldots, k-1. \tag{2.2}$$

The properties of compact support and vanishing moments lead to bases in which a variety of integral operators are represented by sparse matrices. In particular, an integral operator whose kernel is non-oscillatory and analytic except along a finite set of curves, when expanded in one of these bases, is sparse.

In the following sections, we construct multi-wavelet bases in one and several dimensions, prove their completeness, convergence, and sparsity properties, and present several examples to illustrate their numerical usefulness.

2.1 Multi-Wavelet Bases

2.1.1 The One-Dimensional Construction

We first restrict our attention to the finite interval $[0,1] \subset \mathcal{R}$ and we construct a basis for $\mathcal{L}^2[0,1]$. We employ the multi-resolution analysis framework developed by Mallat [12] and Meyer [14], and discussed at length by Daubechies [5]. We suppose that k is a positive integer and for $m = 0, 1, 2, \ldots$ we define a space S_m^k of piecewise polynomial functions,

$$S_m^k = \{f : \text{ the restriction of } f \text{ to the interval } (2^{-m}n, 2^{-m}(n+1)) \text{ is a polynomial of degree less than } k, \text{ for } n = 0, \ldots, 2^m - 1, \text{ and } f \text{ vanishes elsewhere}\}. \tag{2.3}$$

It is apparent that the space S_m^k has dimension $2^m k$ and

$$S_0^k \subset S_1^k \subset \cdots \subset S_m^k \subset \cdots.$$

For $m = 0, 1, 2, \ldots$ we define the $2^m k$-dimensional space R_m^k to be the orthogonal complement of S_m^k in S_{m+1}^k,

$$S_m^k \oplus R_m^k = S_{m+1}^k, \qquad R_m^k \perp S_m^k,$$

so we have the decomposition

$$S_m^k = S_0^k \oplus R_0^k \oplus R_1^k \oplus \cdots \oplus R_{m-1}^k. \tag{2.4}$$

Suppose that functions $h_1, \ldots, h_k : \mathcal{R} \to \mathcal{R}$ form an orthogonal basis for R_0^k. By the orthogonality of R_0^k to S_0^k, the first k moments of $h_1, \ldots, h_k$ vanish,

$$\int_0^1 h_j(x)\, x^i\, dx = 0, \qquad i = 0, 1, \ldots, k-1.$$

The $2k$-dimensional space R_1^k is spanned by the $2k$ orthogonal functions $h_1(2x)$, $\ldots, h_k(2x)$, $h_1(2x-1), \ldots, h_k(2x-1)$, of which k are supported on the interval $[0, \frac{1}{2}]$ and k on $[\frac{1}{2}, 1]$. In general, the space R_m^k is spanned by $2^m k$ functions obtained from $h_1, \ldots, h_k$ by translation and dilation. There is some freedom in choosing the functions $h_1, \ldots, h_k$ within the constraint that they be orthogonal; by requiring normality and additional vanishing moments, we specify them uniquely, up to sign.

In preparation for the definition of $h_1, \ldots, h_k$, we construct the k functions $f_1, \ldots, f_k : \mathcal{R} \to \mathcal{R}$, supported on the interval $[-1, 1]$, with the following properties:

1. The restriction of f_i to the interval $(0,1)$ is a polynomial of degree $k-1$.

2. The function f_i is extended to the interval $(-1,0)$ as an even or odd function according to the parity of $i+k-1$.

3. The functions $f_1,\ldots,f_k$ satisfy the following orthogonality and normality conditions:

$$\int_{-1}^{1} f_i(x)\, f_j(x)\, dx \equiv \langle f_i, f_j\rangle = \delta_{ij}, \qquad i,j = 1,\ldots,k.$$

4. The function f_j has vanishing moments,

$$\int_{-1}^{1} f_j(x)\, x^i\, dx = 0, \qquad i = 0,1,\ldots,j+k-2.$$

Properties 1 and 2 imply that there are k^2 polynomial coefficients that determine the functions $f_1,\ldots,f_k$, while properties 3 and 4 provide k^2 (non-trivial) constraints. It turns out that the equations uncouple to give k nonsingular linear systems that may be solved to obtain the coefficients, yielding the functions uniquely (up to sign). Rather than prove that these systems are nonsingular, however, we now determine $f_1,\ldots,f_k$ constructively.

We start with $2k$ functions which span the space of functions that are polynomials of degree less than k on the interval $(0,1)$ and on $(-1,0)$, then orthogonalize k of them, first to the functions $1,x,\ldots,x^{k-1}$, then to the functions $x^k,x^{k+1},\ldots,x^{2k-1}$, and finally among themselves. We define $f_1^1,f_2^1,\ldots,f_k^1$ by the formula

$$f_j^1(x) = \begin{cases} x^{j-1}, & x\in(0,1),\\ -x^{j-1}, & x\in(-1,0),\\ 0, & \text{otherwise,}\end{cases}$$

and note that the $2k$ functions $1,x,\ldots,x^{k-1},f_1^1,f_2^1,\ldots,f_k^1$ are linearly independent, hence span the space of functions that are polynomials of degree less than k on $(0,1)$ and on $(-1,0)$.

1. By the Gram-Schmidt process we orthogonalize f_j^1 with respect to $1,x,\ldots,x^{k-1}$, to obtain f_j^2, for $j=1,\ldots,k$. This orthogonality is preserved by the remaining orthogonalizations, which only produce linear combinations of the f_j^2.

2. The next sequence of steps yields $k-1$ functions orthogonal to x^k, of which $k-2$ functions are orthogonal to x^{k+1}, and so forth, down to 1 function which is orthogonal to x^{2k-2}. First, if at least one of f_j^2 is not orthogonal to x^k, we reorder the functions so that it appears first, $\langle f_1^2,x^k\rangle \neq 0$. We then define $f_j^3 = f_j^2 - a_j\cdot f_0^2$ where a_j is chosen so $\langle f_j^3,x^k\rangle = 0$ for $j=2,\ldots,k$, achieving the desired orthogonality to x^k. Similarly, we orthogonalize to $x^{k+1},\ldots,x^{2k-2}$, each in turn, to obtain $f_1^2,f_2^3,f_3^4,\ldots,f_k^{k+1}$ such that $\langle f_j^{j+1},x^i\rangle = 0$ for $i\le j+k-2$.

3. Finally, we do Gram-Schmidt orthogonalization on $f_k^{k+1}, f_{k-1}^k, \ldots, f_1^2$, in that order, and normalize to obtain $f_k, f_{k-1}, \ldots, f_1$.

It is readily seen that the f_j satisfy properties 1-4 of the previous paragraph. Defining $h_1, \ldots, h_k : \mathcal{R} \to \mathcal{R}$ by the formula

$$h_i(x) = 2^{1/2} f_i(2x-1), \qquad i = 1, \ldots, k,$$

we obtain the equality

$$R_0^k = \text{linear span } \{h_i : \ i = 1, \ldots, k\},$$

and, generally,

$$R_m^k = \text{linear span } \{h_{j,m}^n : \ h_{j,m}^n(x) = 2^{m/2} h_j(2^m x - n), \quad j = 1, \ldots, k; \ n = 0, \ldots, 2^m - 1\}. \tag{2.5}$$

We will show next that dilates and translates of the piecewise polynomial functions $h_1, \ldots, h_k$ form an orthonormal basis for $\mathcal{L}^2(\mathcal{R})$. Furthermore, a subset of these dilates and translates, combined with a basis for S_0^k, forms a basis for $\mathcal{L}^2[0,1]$.

2.1.2 Completeness of One-Dimensional Construction

We define the space S^k to be the union of the S_m^k, given by the formula

$$S^k = \bigcup_{m=0}^{\infty} S_m^k, \tag{2.6}$$

and observe that $\overline{S^k} = \mathcal{L}^2[0,1]$. In particular, S^k contains the Haar basis for $\mathcal{L}^2[0,1]$, consisting of functions piecewise constant on each of the subintervals $(2^{-m}n, 2^{-m}(n+1))$. Here the closure $\overline{S^k}$ is defined with respect to the $\mathcal{L}^2$-norm,

$$\|f\| = \langle f, f \rangle^{1/2},$$

where the inner product $\langle f, g \rangle$ is defined by the formula

$$\langle f, g \rangle = \int_0^1 f(x)\, g(x)\, dx.$$

We let $u_1, \ldots, u_k$ denote an orthonormal basis for S_0^k; in view of Eqs. (2.4), (2.5), and (2.6), the orthonormal set

$$\begin{aligned} &\{u_j : \ j = 1, \ldots, k\} \\ \cup\, &\{h_{j,m}^n : \ j = 1, \ldots, k; \ m = 0, 1, 2, \ldots; \ n = 0, \ldots, 2^m - 1\} \end{aligned}$$

spans $\mathcal{L}^2[0,1]$; we refer to it as the *multi-wavelet basis of order k* for $\mathcal{L}^2[0,1]$.

Now we construct a basis for $\mathcal{L}^2(\mathcal{R})$ by defining, for $m \in \mathcal{Z}$, the space $\tilde{S}_m^k$ by the formula

$$\tilde{S}_m^k = \{f : \text{ the restriction of } f \text{ to the interval } (2^{-m}n, 2^{-m}(n+1)) \text{ is a polynomial of degree less than } k, \text{ for } n \in \mathcal{Z}\}$$

and observing that the space $\tilde{S}_{m+1}^k \backslash \tilde{S}_m^k$ is spanned by the orthonormal set

$$\{h_{j,m}^n : \ h_{j,m}^n(x) = 2^{m/2}\, h_j(2^m x - n),\ j = 1, \ldots, k;\ n \in \mathcal{Z}\}.$$

Thus $\mathcal{L}^2(\mathcal{R})$, which is contained in $\overline{\bigcup_m \tilde{S}_m^k}$, has orthonormal basis

$$\{h_{j,m}^n : \ j = 1, \ldots, k;\ m, n \in \mathcal{Z}\}.$$

2.1.3 Construction in Multiple Dimensions

The construction of our bases for $\mathcal{L}^2[0,1]$ and $\mathcal{L}^2(\mathcal{R})$ can be extended to certain other function spaces, including $\mathcal{L}^2[a,b]^d$ and $\mathcal{L}^2(\mathcal{R}^d)$, for any positive integer d. We now outline this extension by giving the basis for $\mathcal{L}^2[0,1]^2$, which is illustrative of the construction for any finite-dimensional space. We define the space $S_m^{k,2}$ by the formula

$$S_m^{k,2} = S_m^k \times S_m^k, \qquad m = 0, 1, 2, \ldots,$$

where S_m^k is defined by Eq. (2.3). We further define $R_m^{k,2}$ to be the orthogonal complement of $S_m^{k,2}$ in $S_{m+1}^{k,2}$,

$$S_m^{k,2} \oplus R_m^{k,2} = S_{m+1}^{k,2}, \qquad R_m^{k,2} \perp S_m^{k,2}.$$

Then $R_0^{k,2}$ is the space spanned by the orthonormal basis

$$\{u_i(x)h_j(y),\ h_i(x)u_j(y),\ h_i(x)h_j(y) : \ i, j = 1, \ldots, k\}.$$

Among these $3k^2$ basis elements each element $v(x,y)$ has no projection on low-order polynomials,

$$\int_0^1 \int_0^1 v(x,y)\, x^i\, y^j\, dx\, dy = 0, \qquad i, j = 0, 1, \ldots, k-1.$$

The space $R_m^{k,2}$ is spanned by dilations and translations of the $v(x,y)$ and the basis of $\mathcal{L}^2[0,1]^2$ consists of these functions and the low-order polynomials $\{u_i(x)u_j(y) : i, j = 1, \ldots, k\}$.

2.2 Convergence of the Multi-Wavelet Bases

For a function $f \in \mathcal{L}^2[0,1]$, a positive integer k, and $m = 0, 1, 2 \ldots$, we define the orthogonal projection $Q_m^k f$ of f onto S_m^k by the formula

$$(Q_m^k f)(x) = \sum_{j,n} \langle f, u_{j,m}^n \rangle \cdot u_{j,m}^n(x),$$

where $\{u_{j,m}^n\}$ is an orthonormal basis for S_m^k. The projections $Q_m^k f$ converge (in the mean) to f as $m \to \infty$. If the function f is several times differentiable, we can bound the error, as established by the following lemma.

Lemma 2.1 *Suppose that the function $f : [0,1] \to \mathcal{R}$ is k times continuously differentiable, $f \in C^k[0,1]$. Then $Q_m^k f$ approximates f with mean error bounded as follows:*

$$\|Q_m^k f - f\| \leq 2^{-mk} \frac{2}{4^k k!} \sup_{x \in [0,1]} |f^{(k)}(x)| \tag{2.7}$$

Proof. We divide the interval $[0,1]$ into subintervals on which $Q_m^k f$ is a polynomial; the restriction of $Q_m^k f$ to one such subinterval $I_{m,n}$ is the polynomial of degree less than k that approximates f with minimum mean error. We then use the maximum error estimate for the polynomial which interpolates f at Chebyshev nodes of order k on $I_{m,n}$.

We define $I_{m,n} = [2^{-m}n, 2^{-m}(n+1)]$ for $n = 0, 1, \ldots, 2^m - 1$, and obtain

$$\begin{aligned}
\|Q_m^k f - f\|^2 &= \int_0^1 \left[(Q_m^k f)(x) - f(x)\right]^2 dx \\
&= \sum_n \int_{I_{m,n}} \left[(Q_m^k f)(x) - f(x)\right]^2 dx \\
&\leq \sum_n \int_{I_{m,n}} \left[(C_{m,n}^k f)(x) - f(x)\right]^2 dx \\
&\leq \sum_n \int_{I_{m,n}} \left(\frac{2^{1-mk}}{4^k k!} \sup_{x \in I_{m,n}} |f^{(k)}(x)|\right)^2 dx \\
&\leq \left(\frac{2^{1-mk}}{4^k k!} \sup_{x \in [0,1]} |f^{(k)}(x)|\right)^2,
\end{aligned}$$

and by taking square roots we have bound (2.7). Here $C_{m,n}^k f$ denotes the polynomial of degree k which agrees with f at the Chebyshev nodes of order k on $I_{m,n}$, and we have used the well-known maximum error bound for Chebyshev interpolation (see, *e.g.*, [4]). □

The error of the approximation $Q_m^k f$ of f therefore decays like 2^{-mk} and, since S_m^k has a basis of $2^m k$ elements, we have convergence of order k. For the generalization to d dimensions, it is easily seen that the rate of convergence is of order k/d.

2.3 Second-Kind Integral Equations

A linear Fredholm integral equation of the second kind may be written in the form

$$f(x) - \int_a^b K(x,t)\, f(t)\, dt = g(x), \tag{2.8}$$

where the function K is the kernel, g is the right-hand-side, and f is the unknown. For notational simplicity, in this chapter we restrict our attention to the interval $[a,b] = [0,1]$. We use the symbol $\mathcal{K}$ to denote the integral operator of Eq. (2.8), given by the formula

$$(\mathcal{K}f)(x) = \int_0^1 K(x,t)\, f(t)\, dt,$$

for all $f \in \mathcal{L}^2[0,1]$ and $x \in [0,1]$. Suppose that $\{b_1, b_2, \ldots\}$ is an orthonormal basis for $\mathcal{L}^2[0,1]$; the expansion of K in this basis is given by the formula

$$K(x,t) = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} K_{ij}\, b_i(x)\, b_j(t), \tag{2.9}$$

where the coefficient K_{ij} is given by the expression

$$K_{ij} = \int_0^1 \int_0^1 K(x,t)\, b_i(x)\, b_j(t)\, dx\, dt, \qquad i,j = 1,2,\ldots. \tag{2.10}$$

Similarly, the functions f and g have expansions

$$f(x) = \sum_{i=1}^{\infty} f_i\, b_i(x), \qquad g(x) = \sum_{i=1}^{\infty} g_i\, b_i(x),$$

where the coefficients f_i and g_i are given by the formulae

$$f_i = \int_0^1 f(x)\, b_i(x)\, dx, \qquad g_i = \int_0^1 g(x)\, b_i(x)\, dx, \qquad i = 1,2,\ldots.$$

The integral equation (2.8) then becomes the system of equations

$$f_i - \sum_{j=1}^{\infty} K_{ij}\, f_j = g_i, \qquad i = 1,2,\ldots.$$

The expansion for K may be truncated at a finite number of terms, yielding the integral operator R defined by the formula

$$(Rf)(x) = \int_0^1 \sum_{i=0}^{n} \sum_{j=0}^{n} (K_{ij}\, b_i(x)\, b_j(t))\, f(t)\, dt, \qquad f \in \mathcal{L}^2[0,1],\ x \in [0,1],$$

which approximates $\mathcal{K}$. Integral equation (2.8) is thereby approximated by the system

$$f_i - \sum_{j=1}^{n} K_{ij} f_j = g_i, \qquad i = 1, \ldots, n, \tag{2.11}$$

a system of n equations in n unknowns. Eqs. (2.11) may be solved numerically to yield an approximate solution to Eq. (2.8), given by the expression

$$f_R(x) \doteq \sum_{i=1}^{n} f_i \, b_i(x).$$

How large is the error $e_R = f - f_R$ of the approximate solution? We follow the derivation by Delves and Mohamed in [6]. Rewriting Eqs. (2.8) and (2.11) in terms of operators $\mathcal{K}$ and R, we have

$$\begin{aligned} (I - \mathcal{K})f &= g \\ (I - R)f_R &= g, \end{aligned}$$

and combining the latter equations yields

$$(I - \mathcal{K})e_R = (\mathcal{K} - R)f_R.$$

Provided that $(I - \mathcal{K})^{-1}$ exists, we obtain the error bound

$$\|e_R\| \leq \|(I - \mathcal{K})^{-1}\| \cdot \|(\mathcal{K} - R)f_R\|. \tag{2.12}$$

The error depends, therefore, on the conditioning of the original integral equation, as is apparent from the term $\|(I - \mathcal{K})^{-1}\|$, and on the fidelity of the finite-dimensional operator R to the integral operator $\mathcal{K}$.

2.4 Sparse Representation of Integral Operators and Their Inverses

2.4.1 Representation in Multi-Wavelet Bases

We consider integral operators $\mathcal{K}$ with kernels that are analytic, except at $x = t$, where they are singular. In particular, we analyze singularities of the form $\log|x - t|$ or the form $|x - t|^{\alpha}$, with $0 < |\alpha| < 1$. An operator with such a kernel K, expanded in one of the multi-wavelet bases defined above, is represented as a sparse matrix. This sparseness is due to the smoothness of K on rectangles separated from the "diagonal".

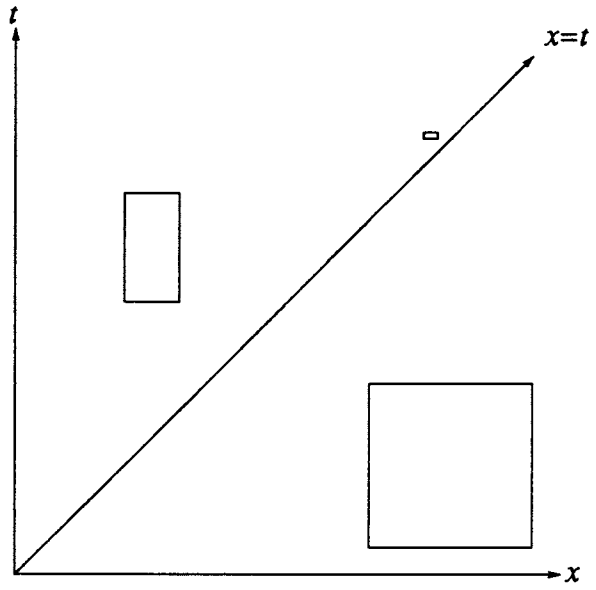

Figure 2.1: *Rectangular regions (just) separated from the diagonal.*

Definition 2.2 We say that a rectangular region oriented parallel to the coordinate axes x, t is *separated from the diagonal* if its distance in the horizontal or vertical direction from the line $x = t$ is at least the length of its longer side. In symbols, a region $[x, x+a] \times [t, t+b] \subset \mathcal{R}^2$ is separated from the diagonal if $a + \max\{a, b\} \leq t - x$ or $b + \max\{a, b\} \leq x - t$.

This definition is illustrated in Fig. 2.1.

Suppose that k is a positive integer and that $\{b_1, b_2, \ldots\}$ is the multi-wavelet basis for $\mathcal{L}^2[0,1]$ of order k, defined in Section 2.1. We let I_j denote the interval of support of b_j, and we assume that the sequence of basis functions $b_1, b_2, \ldots$ is ordered so that $I_1, I_2, \ldots$ have non-increasing lengths. For large n, the matrix $\{K_{ij}\}_{i,j=1,\ldots,n}$ is sparse, to high precision, as is proved in the following propositions.

Lemma 2.3 *Suppose that the function* $K : [0,1] \times [0,1] \rightarrow \mathcal{R}$ *is given by the formula* $K(x,t) = \log|x-t|$. *The expansion of* K *in the multi-wavelet basis of order* k *(Eq. 2.9) has coefficients* K_{ij} *which satisfy the bound*

$$|K_{ij}| \leq \frac{1}{8k \cdot 3^{k-1}} \tag{2.13}$$

whenever the rectangular region $I_i \times I_j$ *is separated from the diagonal.*

Proof. Suppose that the intervals I_i and I_j are given by the expressions $I_i = [x_0, x_0 + a]$ and $I_j = [t_0, t_0 + b]$; without loss of generality we assume (one

of two equivalent cases) that $b + \max\{a, b\} \leq x_0 - t_0$. It is immediate from this inequality that

$$\left|\frac{x_0 + a/2 - x}{x_0 + a/2 - t}\right| \leq \frac{1}{3} \tag{2.14}$$

for $(x, t) \in I_i \times I_j$.

We use the Taylor expansion for the natural logarithm about $c > 0$,

$$\log(c + y) = \log(c) + (y/c) - (y/c)^2/2 + (y/c)^3/3 - (y/c)^4/4 + \cdots,$$

for $|y| < c$. We now let $c = x_0 + a/2 - t$ and $y = x - x_0 - a/2$ and for $(x, t) \in I_i \times I_j$ we obtain the formula

$$\log|x - t| = \log(x_0 + a/2 - t) - \sum_{m=1}^{\infty} \frac{1}{m}\left(\frac{x_0 + a/2 - x}{x_0 + a/2 - t}\right)^m. \tag{2.15}$$

We now apply Eqs. (2.10), (2.15), (2.2), and (2.14), each in turn, to obtain

$$\begin{aligned}
|K_{ij}| &= \left|\int_{t_0}^{t_0+b}\int_{x_0}^{x_0+a} K(x,t)\, b_i(x)\, b_j(t)\, dx\, dt\right| \\
&\leq \int_{t_0}^{t_0+b}\left|\int_{x_0}^{x_0+a} \log|x - t|\, b_i(x)\, dx\right| |b_j(t)|\, dt \\
&= \int_{t_0}^{t_0+b}\left|\int_{x_0}^{x_0+a}\left[\log(x_0 + \frac{a}{2} - t)\right.\right. \\
&\qquad \left.\left. - \sum_{m=1}^{\infty}\frac{1}{m}\left(\frac{x_0 + a/2 - x}{x_0 + a/2 - t}\right)^m\right] b_i(x)\, dx\right| |b_j(t)|\, dt \\
&\leq \int_{t_0}^{t_0+b}\left|\int_{x_0}^{x_0+a}\sum_{m=k}^{\infty}\frac{1}{m}\left(\frac{x_0 + a/2 - x}{x_0 + a/2 - t}\right)^m b_i(x)\, dx\right| |b_j(t)|\, dt \\
&\leq \int_{t_0}^{t_0+b}\int_{x_0}^{x_0+a}\frac{1}{k}\sum_{m=k}^{\infty}\left(\frac{1}{3}\right)^m |b_i(x)|\, dx\, |b_j(t)|\, dt \\
&\leq \int_{t_0}^{t_0+b}\int_{x_0}^{x_0+a}\frac{1}{2k\cdot 3^{k-1}} |b_i(x)|\, dx\, |b_j(t)|\, dt \\
&\leq \frac{1}{2k\cdot 3^{k-1}}\int_{t_0}^{t_0+b}\sqrt{\left(\int_{x_0}^{x_0+a} b_i^2(x)\, dx\right)\left(\int_{x_0}^{x_0+a} 1\, dx\right)}\, |b_j(t)|\, dt \\
&\leq \frac{\sqrt{ab}}{2k\cdot 3^{k-1}} \leq \frac{1}{8k\cdot 3^{k-1}},
\end{aligned}$$

as was to be proved. □

Lemma 2.4 *Suppose that the function $L : D \times D \to \mathcal{C}$ is given by the formula $L(z, w) = f(z, w)\log|z - w|$, where D is the closed disk of radius $\frac{3}{2}$ centered at $z = \frac{1}{2}$ and f is analytic in a domain containing $D \times D \subset \mathcal{C}^2$. Suppose further*

that the function K is the restriction of L to $[0,1] \times [0,1]$. The expansion of K in the multi-wavelet basis of order k has coefficients K_{ij} which satisfy the bound

$$|K_{ij}| \leq \left(\frac{k}{8} + \frac{3}{16}\right) \frac{1}{3^{k-1}} \sup_{z,w \in \partial D} |f(z,w)|, \tag{2.16}$$

whenever the rectangular region $I_i \times I_j$ is separated from the diagonal.

Proof. We combine the method of proof used in Lemma 2.3 with the formula for the derivative of a product,

$$\frac{\partial^m K(x,t)}{\partial x^m} = \sum_{r=0}^{m} \binom{m}{r} \frac{\partial^r f(x,t)}{\partial x^r} \cdot \frac{\partial^{m-r} \log|x-t|}{\partial x^{m-r}}.$$

By the Cauchy integral formula we obtain

$$\left|\frac{\partial^r f(x,t)}{\partial x^r}\right| \leq r! \sup_{z,w \in \partial D} |f(z,w)|$$

for $(x,t) \in [0,1] \times [0,1]$. For the logarithm, differentiation yields the formula

$$\frac{\partial^{m-r} \log|x-t|}{\partial x^{m-r}} = \frac{(-1)^{m-r-1}(m-r-1)!}{(x-t)^{m-r}},$$

for $r < m$. Combining these expressions, we obtain

$$\begin{aligned}
\left|\frac{\partial^m K(x,t)}{\partial x^m}\right| &\leq \sum_{r=0}^{m} \binom{m}{r} \left|\frac{\partial^r f(x,t)}{\partial x^r}\right| \cdot \left|\frac{\partial^{m-r} \log|x-t|}{\partial x^{m-r}}\right| \\
&\leq \sup_{z,w \in \partial D} |f(z,w)| \left(\sum_{r=0}^{m-1} \binom{m}{r} r! \frac{(m-r-1)!}{|x-t|^{m-r}} + m! \left|\log|x-t|\right|\right) \\
&\leq S_f \cdot \left(m! \frac{2+\log m}{|x-t|^m}\right) \qquad (2.17)
\end{aligned}$$

for $|x-t| \leq 1$ and $m \geq 1$, where $S_f = \sup_{z,w \in \partial D} |f(z,w)|$.

Suppose that the intervals I_i and I_j are given by the expressions $I_i = [x_0, x_0 + a]$ and $I_j = [t_0, t_0 + b]$; we assume without loss of generality that $b + \max\{a,b\} \leq x_0 - t_0$. It follows directly from this inequality that

$$\left|\frac{x_0 + a/2 - x}{x_0 + a/2 - t}\right| \leq \frac{1}{3} \tag{2.18}$$

for $(x,t) \in I_i \times I_j$. We now apply Eqs. (2.10), (2.2), (2.17), and (2.18), to obtain

$$|K_{ij}| = \left|\int_{t_0}^{t_0+b} \int_{x_0}^{x_0+a} K(x,t)\, b_i(x)\, b_j(t)\, dx\, dt\right|$$

$$
\begin{aligned}
&\leq \int_{t_0}^{t_0+b} \left| \int_{x_0}^{x_0+a} \sum_{m=0}^{\infty} \frac{(x_0 + a/2 - x)^m}{m!} \frac{\partial^m K(x_0 + a/2, t)}{\partial {x_0}^m} b_i(x)\, dx \right| |b_j(t)|\, dt \\
&\leq \int_{t_0}^{t_0+b} \int_{x_0}^{x_0+a} \sum_{m=k}^{\infty} \left| \frac{x_0 + a/2 - x}{x_0 + a/2 - t} \right|^m S_f\, (2 + \log m)\, |b_i(x)|\, dx\, |b_j(t)|\, dt \\
&\leq \int_{t_0}^{t_0+b} \int_{x_0}^{x_0+a} S_f \sum_{m=k}^{\infty} \left(\frac{1}{3}\right)^m (m+1)\, |b_i(x)|\, dx\, |b_j(t)|\, dt \\
&\leq \int_{t_0}^{t_0+b} \int_{x_0}^{x_0+a} S_f \left(\frac{k}{2} + \frac{3}{4}\right) \frac{1}{3^{k-1}}\, |b_i(x)|\, dx\, |b_j(t)|\, dt \\
&\leq S_f \left(\frac{k}{2} + \frac{3}{4}\right) \frac{1}{3^{k-1}} \int_{t_0}^{t_0+b} \sqrt{\left(\int_{x_0}^{x_0+a} {b_i}^2(x)\, dx\right) \left(\int_{x_0}^{x_0+a} 1\, dx\right)}\, |b_j(t)|\, dt \\
&\leq S_f \left(\frac{k}{2} + \frac{3}{4}\right) \frac{\sqrt{ab}}{3^{k-1}} \leq S_f \left(\frac{k}{8} + \frac{3}{16}\right) \frac{1}{3^{k-1}},
\end{aligned}
$$

which was to be proved. □

The proofs of the following two lemmas closely resemble those of Lemma 2.3 and Lemma 2.4, and are omitted.

Lemma 2.5 *Suppose that the function* $K : [0,1] \times [0,1] \to \mathcal{R}$ *is given by the formula* $K(x,t) = |x-t|^\alpha$ *with* $0 < |\alpha| < 1$. *Then the expansion coefficient* K_{ij} *of the function* K *in the multi-wavelet basis of order* k *satisfies the bound*

$$|K_{ij}| \leq \frac{1}{2 \cdot 3^{k-1}} \tag{2.19}$$

whenever the rectangular region $I_i \times I_j$ *is separated from the diagonal.*

Lemma 2.6 *Suppose that the function* $L : D \times D \to \mathcal{C}$ *is given by the formula* $L(z,w) = f(z,w)|z-w|^\alpha$, *with* $0 < |\alpha| < 1$, *where* D *is the closed disk of radius* $\frac{3}{2}$ *centered at* $z = \frac{1}{2}$ *and* f *is analytic in a domain containing* $D \times D \subset \mathcal{C}^2$. *Suppose further that the function* K *is the restriction of* L *to* $[0,1] \times [0,1]$. *The expansion of* K *in the multi-wavelet basis of order* k *has coefficients* K_{ij} *which satisfy the bound*

$$|K_{ij}| \leq \left(\frac{k}{2} + \frac{3}{4}\right) \frac{1}{3^{k-1}} \sup_{z,w \in \partial D} |f(z,w)|, \tag{2.20}$$

whenever the rectangular region $I_i \times I_j$ *is separated from the diagonal.*

The four preceding lemmas show that for a smooth kernel K with logarithm or power singularity at $x = t$, the order k of the multi-wavelet basis in which K is expanded may be chosen large enough that the expansion coefficient K_{ij} is negligible, provided $I_i \times I_j$ is separated from the diagonal. A similar statement can be proven for any kernel of the form $K(x,t) = \phi(x,t)s(|x-t|)+\psi(x,t)$, where

ϕ, ψ are entire analytic functions of two variables and s is an analytic function except at the origin (where it has a singularity), provided that s is integrable. We do not prove this statement here.

The next lemma establishes the fact that, asymptotically, most regions $I_i \times I_j$ *are* separated from the diagonal.

Lemma 2.7 *Suppose that $I_1, \ldots, I_n$ are the (non-increasing) intervals of support of the first n functions of the multi-wavelet basis of order k. Of the n^2 rectangular regions $I_i \times I_j$, we denote the number separated from the diagonal by $S(n)$ and the number "near" the diagonal by $N(n) = n^2 - S(n)$. Then $N(n)$ grows as $O(n \log n)$; in particular, for $n = 2^l k$ with $l > 0$, we have the formula*

$$N(n) = 6nlk - 15nk - 6lk^2 + 16k^2. \tag{2.21}$$

Proof. The restriction that $n = 2^l k$ ensures that the first n basis functions consist of those functions whose intervals of support have length at least 2^{1-l}. We define $S_1(p)$ to be the number of pairs (i, j) such that the rectangular region $I_i \times I_j$ is separated from the diagonal and $|I_i| = |I_j| = 2^{-p}$, and we observe that $S_1(p) = (2^p - 1)(2^p - 2)\, k^2$ for $p = 0, 1, 2, \ldots$. We further define $S_2(p, q)$ to be the number of pairs (i, j) such that $I_i \times I_j$ is separated from the diagonal and $|I_i| = 2^{-p}$, $|I_j| = 2^{-q}$, and we observe that $S_2(p, q) = S_1(\min\{p, q\}) 2^{|p-q|}$ for $p, q = 0, 1, 2, \ldots$. Finally, we combine these formulae to obtain

$$\begin{aligned}
S(n) &= \sum_{p=0}^{l-1} \left(S_1(p) + \sum_{q=p+1}^{l-1} \left(S_2(p, q) + S_2(q, p) \right) \right) \\
&= \sum_{p=0}^{l-1} S_1(p) \left(1 + 2(2^{l-p} - 2) \right) \\
&= \sum_{p=0}^{l-1} (2^p - 1)(2^p - 2)\, k^2 \, (2^{l-p+1} - 3) \\
&= (4^l - 6 \cdot 2^l l + 15 \cdot 2^l + 6l - 16)\, k^2 \\
&= n^2 - 6nlk + 15nk + 6lk^2 - 16k^2,
\end{aligned}$$

from which Eq. (2.21) follows directly. The assertion that the general growth of $N(n)$ is $O(n \log n)$ follows from Eq. (2.21) and that fact that N is a monotonic function of n. □

2.4.2 Products of Integral Operators

The previous subsection established the fact that a wide class of integral operators, when expanded in multi-wavelet coordinates, are represented to high accuracy as sparse matrices. It readily follows that a product of such integral

operators can be similarly represented. For if we define integral operators $\mathcal{K}_1, \mathcal{K}_2$ by the formulae

$$\begin{aligned}(\mathcal{K}_1 f)(x) &= \int_0^1 K_1(x,t)\, f(t)\, dt \\ (\mathcal{K}_2 f)(x) &= \int_0^1 K_2(x,t)\, f(t)\, dt,\end{aligned}$$

then the product operator $\mathcal{K}_3 = \mathcal{K}_2\mathcal{K}_1$ is given by the formula

$$\begin{aligned}(\mathcal{K}_2\mathcal{K}_1 f)(x) &= \int_0^1 \int_0^1 K_2(x,y)\, K_1(y,t)\, f(t)\, dt\, dy \\ &= \int_0^1 \left(\int_0^1 K_2(x,y)\, K_1(y,t)\, dy \right) f(t)\, dt \\ &= \int_0^1 K_3(x,t)\, f(t)\, dt,\end{aligned}$$

where the kernel K_3 of the product has the form

$$K_3(x,t) = \int_0^1 K_2(x,y)\, K_1(y,t)\, dy.$$

If kernels K_1 and K_2 are analytic except along the diagonal $x = t$, where they have integrable singularities, then the same is true of the product kernel K_3. As a result, the product operator $\mathcal{K}_3$ also has a sparse representation in a multi-wavelet basis.

2.4.3 Schulz Method of Matrix Inversion

Schulz's method [17] is an iterative, quadratically convergent algorithm for computing the inverse of a linear operator. Its performance is characterized by the following lemma.

Lemma 2.8 *Suppose that A is an invertible linear operator, X_0 is the operator given by $X_0 = A^H/\|A^H A\|$, and for $m = 0, 1, 2, \ldots$ the operator X_{m+1} is defined by the recursion*

$$X_{m+1} = 2X_m - X_m A X_m.$$

Then X_{m+1} satisfies the formula

$$I - X_{m+1}A = (I - X_m A)^2. \tag{2.22}$$

Furthermore, $X_m \to A^{-1}$ as $m \to \infty$ and for any $\epsilon > 0$ we have

$$\|I - X_m A\| < \epsilon \quad \text{provided} \quad m \geq 2\log_2 \kappa(A) + \log_2 \log(1/\epsilon), \tag{2.23}$$

where $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ is the condition number of A and the norm $\|A\| =$ (largest eigenvalue of $A^H A)^{1/2}$.

Proof. Eq. (2.22) is obtained directly from the definition of X_{m+1}. Bound (2.23) is equally straightforward. Noting that $A^H A$ is symmetric positive-definite and letting λ_0 denote the smallest and λ_1 the largest eigenvalue of $A^H A$ we have

$$\begin{aligned} \|I - X_0 A\| &= \left\| I - \frac{A^H A}{\|A^H A\|} \right\| \\ &= 1 - \lambda_0/\lambda_1 \\ &= 1 - \kappa(A)^{-2}. \end{aligned} \tag{2.24}$$

From Eq. (2.22) we obtain $I - X_m A = (I - X_0 A)^{2^m}$, which in combination with Eq. (2.24) and simple manipulation yields bound (2.23). □

The Schulz method is a notably simple scheme for matrix inversion and its convergence is extremely rapid. It is rarely used, however, because it involves matrix-matrix multiplications on each iteration; for most problem formulations, this process requires order $O(n^3)$ operations for an $n \times n$ matrix. As we have seen above, on the other hand, a discretized integral operator A represented in the basis of Section 2.1 has only order $O(n \log n)$ elements (to finite precision). In addition, $A^H A$ and $(A^H A)^m$ are similarly sparse. This property enables us to employ the Schulz algorithm to compute A^{-1} in order $O(n \log^2 n)$ operations.

2.5 Numerical Examples

2.5.1 Basis Functions

In this section we give numerical expressions for the multi-wavelet functions $f_0, f_1, \ldots, f_{k-1}$ and show their graphs for several values of k. Table 2.1 contains, for small k, the polynomials which represent the f_i on the interval $(0, 1)$, together with the reflection formula to extend the functions to $(-1, 1)$, which is their interval of support. Fig. 2.2 shows the graphs of the functions for $k = 4$ and $k = 5$.

2.5.2 Integral Operators and Their Inverses

We compute the expansion in multi-wavelet bases of the integral operator $\mathcal{K}$ defined by the formula

$$(\mathcal{K}f)(x) = \int_0^1 \log|x - t|\, f(t)\, dt, \tag{2.25}$$

which yields the matrix

$$T = \{K_{ij}\}_{i,j=1,\ldots,n}, \tag{2.26}$$

Table 2.1: *Expressions for the orthonormal, vanishing-moment functions* $f_1, \ldots, f_k$, *for various* k, *for argument* x *in the interval* $(0,1)$. *The function* f_i *is extended to the interval* $(-1,1)$ *as an odd or even function, according to the formula* $f_i(x) = (-1)^{i+k-1} f_i(-x)$ *for* $x \in (-1,0)$, *and is zero outside* $(-1,1)$. *The functions, given here for* $k = 1, \ldots, 5$, *are tabulated for larger values of* k *in the chapter appendix.*

	$k = 1$	
$f_1(x) =$	$\sqrt{\frac{1}{2}}$	
	$k = 2$	
$f_1(x) =$	$\sqrt{\frac{3}{2}}$	$(-1 + 2x)$
$f_2(x) =$	$\sqrt{\frac{1}{2}}$	$(-2 + 3x)$
	$k = 3$	
$f_1(x) =$	$\frac{1}{3}\sqrt{\frac{1}{2}}$	$(1 - 24x + 30x^2)$
$f_2(x) =$	$\frac{1}{2}\sqrt{\frac{3}{2}}$	$(3 - 16x + 15x^2)$
$f_3(x) =$	$\frac{1}{3}\sqrt{\frac{5}{2}}$	$(4 - 15x + 12x^2)$
	$k = 4$	
$f_1(x) =$	$\sqrt{\frac{15}{34}}$	$(1 + 4x - 30x^2 + 28x^3)$
$f_2(x) =$	$\sqrt{\frac{1}{42}}$	$(-4 + 105x - 300x^2 + 210x^3)$
$f_3(x) =$	$\frac{1}{2}\sqrt{\frac{35}{34}}$	$(-5 + 48x - 105x^2 + 64x^3)$
$f_4(x) =$	$\frac{1}{2}\sqrt{\frac{5}{42}}$	$(-16 + 105x - 192x^2 + 105x^3)$
	$k = 5$	
$f_1(x) =$	$\sqrt{\frac{1}{186}}$	$(1 + 30x + 210x^2 - 840x^3 + 630x^4)$
$f_2(x) =$	$\frac{1}{2}\sqrt{\frac{1}{38}}$	$(-5 - 144x + 1155x^2 - 2240x^3 + 1260x^4)$
$f_3(x) =$	$\sqrt{\frac{35}{14694}}$	$(22 - 735x + 3504x^2 - 5460x^3 + 2700x^4)$
$f_4(x) =$	$\frac{1}{8}\sqrt{\frac{21}{38}}$	$(35 - 512x + 1890x^2 - 2560x^3 + 1155x^4)$
$f_5(x) =$	$\frac{1}{2}\sqrt{\frac{7}{158}}$	$(32 - 315x + 960x^2 - 1155x^3 + 480x^4)$

where

$$K_{ij} = \int_0^1 \int_0^1 K(x,t)\, b_i(x)\, b_j(t)\, dx\, dt$$

and $\{b_1, b_2, \ldots\}$ is a multi-wavelet basis of $\mathcal{L}^2[0,1]$. This computation is done for the multi-wavelet basis of order $k = 4$, for various sizes n.

In addition the inverse matrix $(I - T)^{-1}$ is obtained by the Schulz method. Table 2.2 displays, for various precisions ϵ, the average number of elements per row in the matrices $I - T$ and $(I - T)^{-1}$. Fig. 2.3 displays the matrices for $n = 128$

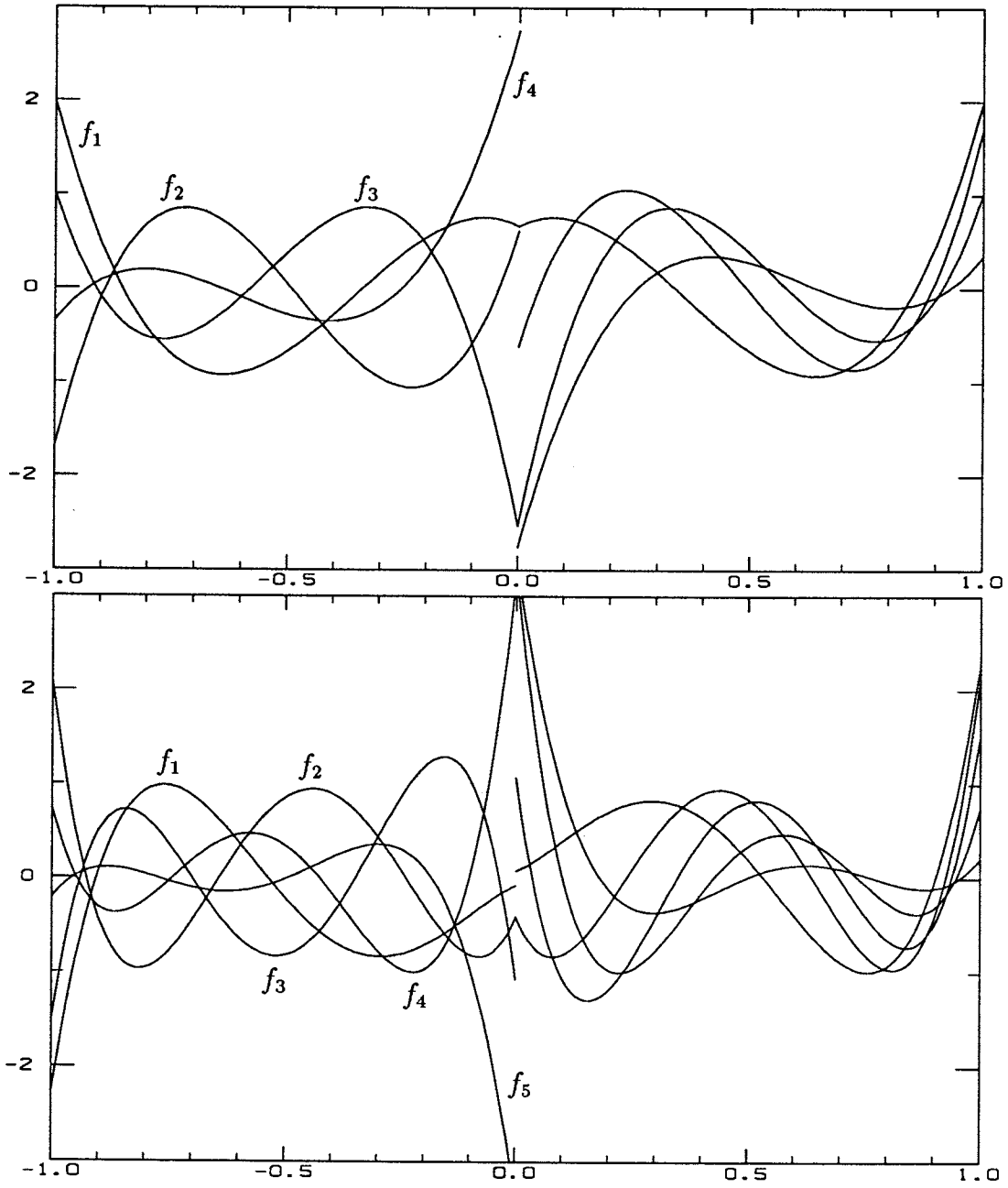

Figure 2.2: *Functions $f_1, \ldots, f_k$ are graphed for $k = 4$ (top graph) and $k = 5$ (bottom). Each function (given in Table 2.1) is a polynomial on the interval $(0, 1)$, is an odd or even function on $(-1, 1)$, and is zero elsewhere.*

and $\epsilon = 10^{-3}$.

Table 2.2: *The average number of elements per row of the matrices $I - T$ and $(I - T)^{-1}$, where T is defined in Eq. (2.26), is tabulated for various precisions ϵ and various sizes n. Here $k = 4$.*

	$\epsilon = 10^{-2}$		$\epsilon = 10^{-3}$		$\epsilon = 10^{-4}$	
n	$I-T$	$(I-T)^{-1}$	$I-T$	$(I-T)^{-1}$	$I-T$	$(I-T)^{-1}$
32	8.8	9.7	19.3	19.6	22.8	23.6
64	9.3	10.0	25.8	26.0	31.9	32.6
128	9.9	10.1	29.2	29.4	38.2	38.8
256	11.8	11.8	30.1	30.3	41.9	42.7

2.6 Discussion

The results of the previous subsection demonstrate, for a particular integral operator, that the multi-wavelet representations are sparse. The matrix has a peculiar structure in which the non-negligible elements are contained in blocks lying along rays emanating from one corner of the matrix. Furthermore, the inverse matrix shares that structure. This property is a general characteristic of integral operators with non-oscillatory kernels that possess diagonal singularities, and further examples will be given in Chapter 4.

The kernel $K(x,t) = \log|x-t|$ of the previous subsection was chosen, however, because the projections K_{ij} could be computed analytically, thereby avoiding use of quadratures. The difficulty here with quadratures is that they would be required for each element K_{ij} (in two dimensions), and would have to cope with the singularity of the logarithm. It was felt that the analytical computation would be more efficient. In fact, the analytical computation, which requires integrating monomials x^j $(0 \le j < k)$ against the logarithm and combining the results with large coefficients, is a very poorly-conditioned procedure. The computations described above required quadruple-precision arithmetic to obtain single-precision accuracy for n as small as 64. This procedure is not recommended.

The fault lies, of course, not with the idea of projection to the multi-wavelet basis, but with the method of projection. The integration should be performed numerically, with quadratures. As mentioned above, such a procedure would require use of quadratures for each matrix element K_{ij}, or potentially order $O(n \log n)$ times. A more efficient procedure is to use the Nyström method, in which only n quadrature applications are required. Numerical quadratures and a vector-space analogue of the multi-wavelet bases are developed in Chapters 3 and 4; these tools enable efficient solution of integral equations using Nyström's method.

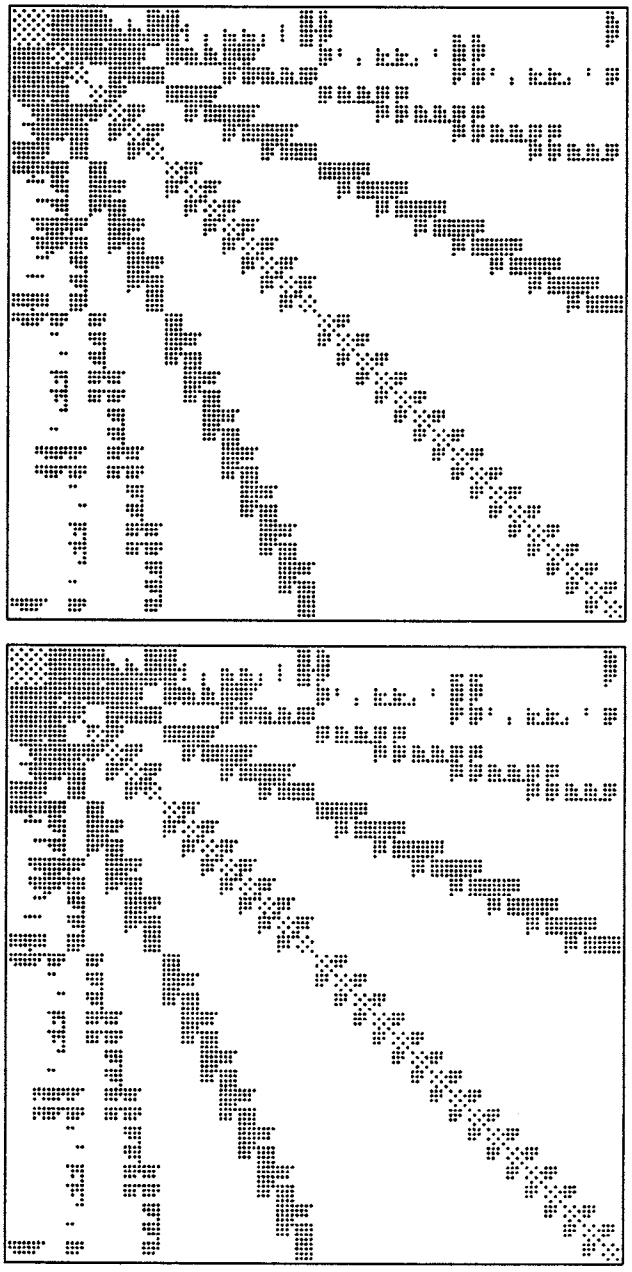

Figure 2.3: *Matrices representing the operators* $1 - \mathcal{K}$ *(top) and* $(1 - \mathcal{K})^{-1}$ *(bottom), with* $\mathcal{K}$ *defined by Eq. (2.25), expanded in the multi-wavelet basis of order* $k = 4$*, for* $n = 128$*. The dots represent elements above a threshold, which is determined so as to bound the relative truncation error at* $\epsilon = 10^{-3}$*.*

2.7 Appendix: Tabulation of Basis Functions

In the following table we present the functions $f_1, \ldots, f_k$, expressed in expansions of orthonormal polynomials for the interval $(0, 1)$, and tabulated for $k = 4, 8, 12$.

Table 2.3: *Coefficients for the orthonormal, vanishing-moment functions* $f_1, \ldots, f_k$, *for various* k, *for argument* x *in the interval* $(0,1)$. *Each function is given in the form* $f(x) = \sum_{j=0}^{k-1} c_j \cdot p_j(x)$, *where* $p_j(x) = P_j(2x-1)\sqrt{2j+1}$, *the Legendre polynomial of degree* j, *shifted to the interval* $(0,1)$ *and normalized. The function* f_i *is extended to the interval* $(-1,1)$ *as an odd or even function, according to the formula* $f_i(x) = (-1)^{i+k-1} f_i(-x)$ *for* $x \in (-1,0)$, *and is zero outside the interval* $(-1,1)$.

$k = 4$

	f_1	f_2	f_3	f_4
c_0	.0000000000000000	−.1543033499620919	.0000000000000000	.2156454872944857
c_1	−.1533929977694741	.2672612419124244	.0878668779193509	−.3735089404169980
c_2	.5940885257860046	−.1725163898355886	−.3403069548648863	.4436221311410142
c_3	.3514675116774037	−.6123724356957945	.6135719910778963	−.3423265984407288

$k = 8$

	f_1	f_2	f_3	f_4
c_0	.0000000000000000	−.0843312289034136	.0000000000000000	.0742292692888546
c_1	−.0394514921576236	.1460659731254334	.0258430814883384	−.1285688658170082
c_2	.1527949721095086	−.1616317375321096	−.1000898242190157	.1503045161172336
c_3	−.3013157484854446	.0637484169813985	.2127004256180621	−.1036447629483221
c_4	.2049959665542275	.2168517314659207	−.2558883246467202	−.0669121025632302
c_5	.5288069924640356	−.3995643491174248	−.0103051686103246	.3099911232623849
c_6	.2463744895581894	−.4560908549012228	.4962827560403292	−.2333502563584798
c_7	.0441081091391231	−.1633067225540632	.3635809208901253	−.5381471079157126

	f_5	f_6	f_7	f_8
c_0	.0000000000000000	−.0871939722326493	.0000000000000000	.1309942892720891
c_1	−.0213434791213834	.1510243900206985	.0192854509208144	−.2268887645206330
c_2	.0826629391872435	−.1841022341475768	−.0746922302404146	.2899079084461328
c_3	−.1829592311019578	.1663892400581053	.1723717448166373	−.3288050657844448
c_4	.2692181887254343	−.0520293149261530	−.2992501225820789	.3336898392328423
c_5	−.1968270025380475	−.1836375913703662	.4050432626343038	−.2900996008838325
c_6	−.1805321245713286	.4401344681347232	−.4003612351829521	.1945752095162505
c_7	.5616280477927965	−.4217364916434134	.2245825563246441	−.0763964570434043

Table 2.3: (continued)

$k = 12$

	f_1	f_2	f_3	f_4
c_0	.0000000000000000	−.0592042739052351	.0000000000000000	.0516869543201819
c_1	−.0179405992366968	.1025448104290915	.0132868156385064	−.0895244309710467
c_2	.0694836420647080	−.1233585459417166	−.0514596156920636	.1096491458548548
c_3	−.1530096755279723	.1032398584967429	.1156254051056246	−.1016897863958198
c_4	.2201077082230140	−.0064287469139523	−.1813181292375394	.0408535191494144
c_5	−.1431401336550512	−.1680946588317245	.1765547047788799	.0858956775115070
c_6	−.1805072879652185	.2749158159062862	−.0058679070010733	−.2211070510482991
c_7	.3717453015731024	−.0081064654950250	−.2736941647216635	.1853011503121983
c_8	.4413074075372503	−.4549238393986653	.1993616369248909	.1593931614774968
c_9	.1881229915369518	−.3636376752880584	.4671192165219669	−.3452248228872576
c_{10}	.0395553032931457	−.1281176981334961	.2822961661345380	−.4313594000122254
c_{11}	.0041396058259610	−.0236611435924639	.0847279654920732	−.2098115229292681

	f_5	f_6	f_7	f_8
c_0	.0000000000000000	−.0490852233087548	.0000000000000000	.0549972807548283
c_1	−.0104273737468520	.0850181006716274	.0094614910169722	−.0952580845454926
c_2	.0403850448662383	−.1054116332841276	−.0366441971390245	.1194424834832239
c_3	−.0918988661694854	.1041544568835931	.0842044896200951	−.1245947511300136
c_4	.1514809211499543	−.0626055685259494	−.1439433554135965	.0954606130089946
c_5	−.1754801708549531	−.0348056650511157	.1860088057424346	−.0149295180909320
c_6	.0929109878353303	.1662387148811842	−.1548304378705184	−.1154462360082784
c_7	.1216196841160863	−.2280521421975709	.0001986646990547	.2421338205360575
c_8	−.2844627881902686	.0647562432345634	.2270020705291812	−.2425421502237302
c_9	.0147751774362919	.2752581716809581	−.2763185483542785	.0010478959001006
c_{10}	.4359353132957034	−.2219439549151931	−.1129177018240620	.3713168669707026
c_{11}	.3761307967348913	−.5043899953764185	.5211079830677339	−.4243990923037588

	f_9	f_{10}	f_{11}	f_{12}
c_0	.0000000000000000	−.0663210971285699	.0000000000000000	.0971353300365081
c_1	−.0090771349437703	.1148715098403935	.0078798389563931	−.1682433268332032
c_2	.0351555924684996	−.1458392917293345	−.0305184850489067	.2165192505829722
c_3	−.0816263402165753	.1609203666648590	.0716769100690038	−.2529614538466701
c_4	.1447831048236192	−.1500812540366636	−.1321728306705118	.2777662732296559
c_5	−.2066628313763642	.0983488566098737	.2074571615484287	−.2875975640399399
c_6	.2287027892690286	.0060095635684675	−.2845256425672342	.2777726278131828
c_7	−.1579200668486585	−.1566209531327825	.3405574275289650	−.2447881538876456
c_8	−.0366328605538905	.3115846050759926	−.3475763522283202	.1897944657285511
c_9	.2951802756507381	−.3921104938549786	.2872585567373639	−.1218486998229358
c_{10}	−.4314237351640850	.3270789009891432	−.1727059110267771	.0578048706132345
c_{11}	.2765929192982552	−.1437437082969879	.0572056487761727	−.0152658200454800

Chapter 3

Numerical Quadratures

An integral equation to be solved numerically must be converted into a finite-dimensional problem. The equation's solution, completely characterized by its infinite expansion in a chosen basis, is approximated by a finite truncation of that expansion. Alternatively, the solution may be represented by its values at a finite set of points.

In the previous chapter, a projection method was used, in which the infinite expansion in a multi-wavelet basis was approximated by a truncated expansion containing a finite number of terms. By contrast, in this chapter and the next, we use the method of Nyström, in which the integral is approximated by a quadrature (a weighted average of values of the integrand at selected points). The quadrature's rate of convergence to the integral, as the number of points increases, affects the amount of computation needed to achieve a given accuracy in the solution to the integral equation. The convergence rate depends on the behavior of the integrand and in particular on whether the integrand is singular in the interval of integration. In this chapter we develop rapidly-convergent quadratures for several types of singularities encountered in physical problems.

3.1 Nyström Method

A linear Fredholm integral equation of the second kind may be written in the form

$$f(x) - p(x) \int_a^b K(x,t)\, f(t)\, dt = g(x), \tag{3.1}$$

where the function p is the coefficient, K is the kernel, g is the right hand side, and f is the unknown. The coefficient p is often included as part of the kernel K, but we separate them, for in the following development we allow p considerable freedom (including oscillatory behavior) while K is more restricted. We use the symbol $\mathcal{K}$ to denote the integral operator of Eq. (3.1), which is given by the

formula

$$(\mathcal{K}f)(x) = p(x) \int_a^b K(x,t)\, f(t)\, dt,$$

for all $f \in C[a,b]$ and $x \in [a,b]$. Then Eq. (3.1) written in operator form is

$$(I - \mathcal{K})f = g. \tag{3.2}$$

The Nyström, or *quadrature*, method for numerical solution of integral equations approximates the integral operator $\mathcal{K}$ by the finite-dimensional operator R, characterized by weights $w_1, w_2, \ldots, w_n \in \mathcal{R}$ and points $x_1, x_2, \ldots, x_n \in [a,b]$, and given by the formula

$$(Rf)(x) = p(x) \sum_{j=1}^{n} w_j\, K(x, x_j)\, f(x_j),$$

for all $f \in C[a,b]$ and $x \in [a,b]$. Substitution of R for $\mathcal{K}$ in Eq. (3.2), plus the requirement that the resulting equation hold for $x = x_1, x_2, \ldots, x_n$, yields the following system of n equations in the n unknowns $f_1, f_2, \ldots, f_n$:

$$f_i - p(x_i) \sum_{j=1}^{n} w_j\, K(x_i, x_j)\, f_j = g(x_i), \qquad i = 1, \ldots, n.$$

The approximation $\langle f_1, \ldots, f_n \rangle$ to the solution f of Eq. (3.1) may be extended to all $x \in [a,b]$ by the natural formula

$$f_R(x) = g(x) + p(x) \sum_{i=1}^{n} w_j\, K(x, x_j)\, f_j, \tag{3.3}$$

which satisfies $f_R(x_i) = f_i$ for $i = 1, \ldots, n$. A bound on the error $e_R = f - f_R$ of the Nyström solution is given in Chapter 2 by (2.12),

$$\|e_R\| \leq \|(I - \mathcal{K})^{-1}\| \cdot \|(\mathcal{K} - R) f_R\|.$$

The error depends, therefore, on the conditioning of the original integral equation, as is apparent from the term $\|(I - \mathcal{K})^{-1}\|$, and on the fidelity of the quadrature R to the integral operator $\mathcal{K}$. It is not necessary that $\|\mathcal{K} - R\|$ be small, rather merely that R approximate $\mathcal{K}$ well near the solution f. In this chapter we develop quadrature rules that have this property; nevertheless, they are of a somewhat different form than R and are defined only on the mesh points $x_1, \ldots, x_n$ rather than the whole interval $[a,b]$.

3.2 Corrected Trapezoidal Rules for Singular Functions

The quadrature method we develop arises from a method developed by Rokhlin [16]. Some changes have been made with the aim of solving non-periodic integral equations; in addition, one improvement permits higher order quadratures in practice. These differences from [16] are noted as they are presented.

It is well known that the trapezoidal rule for integration can be modified at the ends via the Euler-Maclaurin summation formula to a rapidly convergent rule, provided the integrand is sufficiently differentiable. We will suppose, instead, that the integrand is singular at one end of the interval and the form of the singularity is known. In this case a modification *at that end* may be determined so that the corrected trapezoidal rule is rapidly convergent.

3.2.1 Differentiable Integrands

We begin with the assumption that the integrand is differentiable throughout the interval of integration. For positive integers l, n, and $m = 2l + 2$, and a function $f \in C^m[a, b]$, the Euler-Maclaurin summation formula is given by the equation (see, *e.g.*, [18])

$$\int_a^b f(x)\,dx = T_n(f) + D_n^m(f, a) - D_n^m(f, b) + E_n^m(f), \tag{3.4}$$

where

$$\begin{aligned} T_n(f) &= h\left(\frac{1}{2}f(a) + f(a+h) + \cdots + f(b-h) + \frac{1}{2}f(b)\right) \\ D_n^m(f,x) &= \sum_{i=1}^{m/2-1} h^{2i}\frac{B_{2i}}{(2i)!}f^{(2i-1)}(x) \\ E_n^m(f) &= h^m \sum_{i=0}^{n-1} \int_{a+hi}^{a+h(i+1)} \frac{B_m((x-a)/h - i) - B_m}{m!}\, f^{(m)}(x)\,dx \\ &= -h^m \frac{B_m}{m!}(b-a)\, f^{(m)}(\xi) \end{aligned}$$

and $h = (b-a)/n$ and $a < \xi < b$. Here B_i are the Bernoulli numbers,

$$B_2 = \frac{1}{6}, \quad B_4 = -\frac{1}{30}, \quad B_6 = \frac{1}{42}, \quad B_8 = -\frac{1}{30}, \ldots.$$

The derivatives which appear in $D_n^m(f, a)$ and $D_n^m(f, b)$ may be approximated by finite differences, to obtain high-order quadrature rules which depend only on

the values of f at equispaced points. Using the Taylor expansion for f about the point a we have, for $i = 1, 2, \ldots, m-2$,

$$f(a + i\,h) = f(a) + \sum_{j=1}^{m-2} f^{(j)}(a)\,\frac{(i\,h)^j}{j!} + f^{(m-1)}(\nu_i)\,\frac{(i\,h)^{m-1}}{(m-1)!} \tag{3.5}$$

with $a < \nu_i < a + ih$. Eqs. (3.5) can be considered to be a system of equations in the unknowns $hf^{(1)}(a)$, $h^2 f^{(2)}(a), \ldots, h^{m-2} f^{(m-2)}(a)$, and the matrix A^m of this system, given by the formula

$$A^m = \begin{pmatrix} 1 & \frac{1}{2!} & \cdots & \frac{1}{(m-2)!} \\ 2 & \frac{2^2}{2!} & \cdots & \frac{2^{m-2}}{(m-2)!} \\ \vdots & & & \vdots \\ m-2 & \frac{(m-2)^2}{2!} & \cdots & \frac{(m-2)^{m-2}}{(m-2)!} \end{pmatrix},$$

is non-singular, since the functions $x^j/j!$ for $j = 1, 2, \ldots$ form a Chebyshev system. We define the vector $v_h^m = \langle v_{h,1}^m, \ldots, v_{h,m-2}^m \rangle^T$ of finite differences by the expression

$$v_h^m = (A^m)^{-1} \begin{pmatrix} f(a+h) - f(a) \\ f(a+2h) - f(a) \\ \vdots \\ f(a + (m-2)h) - f(a) \end{pmatrix}$$

and from Eqs. (3.5) we obtain, for $j = 1, \ldots, m-2$, the error bound

$$\left| v_{h,j}^m - h^j f^{(j)}(a) \right| < e_m\, h^{m-1} \sup_{\xi \in [a, a+(m-2)h]} \left| f^{(m-1)}(\xi) \right|, \tag{3.6}$$

where e_m is independent of $[a, b]$, h, and the function f.

The expressions $v_{h,j}^m$ for the derivatives are used to define the left-end correction ${}_1D_n^m(f, a)$ by the formula

$${}_1D_n^m(f, a) = \sum_{i=1}^{m/2-1} h\,\frac{B_{2i}}{(2i)!}\,v_{h,2i-1}^m. \tag{3.7}$$

Similarly, in Eqs. (3.5) we replace a by b and i by $-i$ to obtain finite-difference expressions $w_h^m = \langle w_{h,1}^m, \ldots, w_{h,m-2}^m \rangle^T$ for the derivatives at the right endpoint $\langle hf'(b), h^2 f^{(2)}(b), \ldots, h^{m-2} f^{(m-2)}(b) \rangle^T$, and we define the right-end correction ${}_1D_n^m(f, b)$ by the formula

$${}_1D_n^m(f, b) = \sum_{i=1}^{m/2-1} h\,\frac{B_{2i}}{(2i)!}\,w_{h,2i-1}^m. \tag{3.8}$$

We thus obtain the corrected trapezoidal rule

$$ {}_1T_n^m(f) = T_n(f) + {}_1D_n^m(f,a) - {}_1D_n^m(f,b) \tag{3.9} $$

that depends only on the values of f at the equispaced points $a, a+h, \ldots, b-h, b$. Combination of the Euler-Maclaurin formula (Eq. 3.4) with the finite differences error bound (3.6) yields the bound

$$ \left| {}_1T_n^m(f) - \int_a^b f(x)\,dx \right| < {}_1e_m\,(b-a)\,h^m \sup_{\xi\in[a,b]} \left| f^{(m)}(\xi) \right|, $$

where ${}_1e_m$ is independent of $[a,b]$, h, and the function f.

Alternatively, the corrections to the trapezoidal rule may be concentrated in the subintervals $[a, a+h)$ and $(b-h, b]$ by replacing h by $h' = h/(m-1)$ in Eqs. (3.5) to define revised endpoint corrections ${}_2D_n^m(f,a)$ and ${}_2D_n^m(f,b)$ by the formulae

$$ {}_2D_n^m(f,a) = \sum_{i=1}^{m/2-1} h\,\frac{B_{2i}}{(2i)!}\,v^m_{h',2i-1} \tag{3.10} $$

$$ {}_2D_n^m(f,b) = \sum_{i=1}^{m/2-1} h\,\frac{B_{2i}}{(2i)!}\,w^m_{h',2i-1}. \tag{3.11} $$

These endpoint corrections give us the "crowded" corrected trapezoidal rule

$$ {}_2T_n^m(f) = T_n(f) + {}_2D_n^m(f,a) - {}_2D_n^m(f,b). \tag{3.12} $$

For the quadrature ${}_2T_n^m$ we obtain the error bound (as for ${}_1T_n^m$)

$$ \left| {}_2T_n^m(f) - \int_a^b f(x)\,dx \right| < {}_2e_m\,(b-a)\,h^m \sup_{\xi\in[a,b]} \left| f^{(m)}(\xi) \right|, $$

where ${}_2e_m$ is independent of $[a,b]$, h, and the function f. This rule has the advantage over the equispaced rule that the constant of the error term is smaller; it has the disadvantage that the coefficients are larger and hence produce larger round-off errors. Independently of these two characteristics, the crowded rule is a suitable starting point for construction of rules for integral equations with singular kernels, as we shall see below.

Coefficients of finite-difference expressions for the derivatives, equispaced corrected trapezoidal rules, and crowded corrected trapezoidal rules are given in the chapter appendix.

3.2.2 Singular Integrands

We now consider integrands of the form

$$f(x) = \phi(x)\, s(x) + \psi(x) \tag{3.13}$$

for all $x \in (0, b]$, where $\phi, \psi \in C^m[0, b]$ and the function s contains the singular part of f. In particular, we construct quadratures for f, where s is given by the formula $s(x) = \log(x)$ or the formula $s(x) = x^\alpha$ with $0 < |\alpha| < 1$.

We define T_n' to be the trapezoidal rule on $[0, b]$ minus the left-end, by the formula

$$T_n'(f, b) = h\left(f(h) + \cdots + f(b-h) + \frac{1}{2} f(b)\right),$$

where $h = b/n$, and define the right-end corrected rule T_n'' by the formula

$$T_n''(f, b) = T_n'(f, b) - {}_2D_n^m(f, b), \tag{3.14}$$

where the term ${}_2D_n^m(f, b)$ is the finite-difference approximation to the Euler-Maclaurin correction, as defined in Eq. (3.11). Given a positive integer l and two finite sequences $\beta = \langle \beta_1, \beta_2, \ldots, \beta_l \rangle$ and $\chi = \langle \chi_1, \chi_2, \ldots, \chi_l \rangle$, with $0 < \chi_1 < \chi_2 < \cdots < \chi_l \leq 1$, we define the left-end correction L_h^β by the formula

$$L_h^\beta(f) = \sum_{i=1}^{l} \beta_i\, f(\chi_i\, h).$$

We now define the linear mapping $T_n^\beta : \mathcal{L}^1[0, b] \to \mathcal{R}$ by the formula

$$T_n^\beta(f, b) = T_n''(f, b) + L_h^\beta(f). \tag{3.15}$$

Although L_h^β and T_n^β depend on χ, this sequence is generally implied (see Remark 3.1 below) and we omit the symbol χ from our notation. The mapping T_n^β with suitable choice of β and χ will be used as a quadrature formula for functions f of the form given in Eq. (3.13). We now show how β may be chosen, given χ.

For positive integers k and n we consider the following system of linear equations with respect to the unknowns $\beta^n = \langle \beta_1^n, \ldots, \beta_{2k}^n \rangle$:

$$\begin{aligned} \sum_{j=1}^{2k} \beta_j^n\, (\chi_j\, h)^i &= \int_0^b x^i\, dx - T_n''(x^i, b) \\ \sum_{j=1}^{2k} \beta_j^n\, (\chi_j\, h)^i\, s(\chi_j\, h) &= \int_0^b x^i\, s(x)\, dx - T_n''(x^i\, s(x), b), \end{aligned} \tag{3.16}$$

for $i = 0, 1, \ldots, k-1$. We will see below that this system has a unique solution β^n and for functions f of the form given in Eq. (3.13) the rule $T_n^{\beta^n}$ is a quadrature with convergence of order at least $k - 1$.

Remark 3.1 In Rokhlin [16], the points $\chi_1, \ldots, \chi_{2k}$ are chosen to be equispaced, namely $\chi_i = i/(2k)$. While this choice leads to satisfactory quadratures for $k \leq 3$, larger values of k correspond to coefficients $\beta_1^n, \ldots, \beta_{2k}^n$ which are large enough to introduce substantial round-off errors to fixed-precision computations. We have found that alternative spacing of the points can be chosen so as to delay the growth in the coefficients. Letting $\chi_1, \ldots, \chi_{2k}$ be *half-Chebyshev* points

$$\chi_i = 1 - \cos\left(\frac{2i-1}{4k} \cdot \frac{\pi}{2}\right) \tag{3.17}$$

results in $\beta_1^n, \ldots, \beta_{2k}^n$ which are satisfactory for $k \leq 5$. Coefficients for both equispaced points and half-Chebyshev points are given in the chapter appendix.

Our purpose in this chapter is the development of quadratures for integral operators with diagonally-singular kernels. We consider kernels of the form

$$K(x,t) = \phi(x,t)\, s(|x-t|) + \psi(x,t),$$

where $\phi, \psi \in C^m([a,b] \times [a,b])$ and s is as introduced above. To compute the value

$$(\mathcal{K}f)(x) = p(x) \int_a^b K(x,t)\, f(t)\, dt, \tag{3.18}$$

the interval $[a,b]$ is divided into $[a,x]$ and $[x,b]$. The integrand is singular on one end of each of these intervals, so the quadrature $T_n^{\beta^n}$ is applicable. We augment our notation by defining the quadrature ${}_LT_n^{\beta^n}$ for an interval $[c,b]$ with a left-end singularity by the formula

$${}_LT_n^{\beta^n}(f(x), c, b) = T_n^{\beta^n}(f(x+c), b-c).$$

Similarly, we define the quadrature ${}_RT_n^{\beta^n}$ for an interval $[a,c]$ with a right-end singularity by the formula

$${}_RT_n^{\beta^n}(f(x), a, c) = T_n^{\beta^n}(f(c-x), c-a).$$

Now we define a quadrature for the integral in Eq. (3.18) by defining the mapping $T_{ni}^k : \mathcal{L}^1([a,b] \times [a,b]) \to \mathcal{R}$ by the formula

$$T_{ni}^k(G) = {}_RT_i^{\beta^i}(G_i, a, x_i) + {}_LT_{n-i}^{\beta^{n-i}}(G_i, x_i, b), \tag{3.19}$$

and further define $T_{n0}^k(G) = {}_LT_n^{\beta^n}(G_0, a, b)$ and $T_{nn}^k(G) = {}_RT_n^{\beta^n}(G_n, a, b)$. Here $G : [a,b] \times [a,b] \to \mathcal{R}$ and $G_i : [a,b] \to \mathcal{R}$ is the restriction of G defined by the formula $G_i(t) = G(x_i, t)$ where $x_i = a + i\,h$ for $i = 0, 1, \ldots, n$, and $h = (b-a)/n$. The quadrature T_{ni}^k has convergence of order $k-1$, uniformly in i, as we will show in the next section.

3.3 Analytical Properties of the Corrected Rules

The following lemma is a restatement of a classical result (see, *e.g.*, [11]).

Lemma 3.2 *Suppose that the function* $s : (0, b] \to \mathcal{R}$ *is given by the formula* $s(x) = \log(x)$ *or the formula* $s(x) = x^\alpha$ *with* $0 < |\alpha| < 1$. *Then the system of Eqs. (3.16) has a unique solution* $\beta^n = \langle \beta_1^n, \ldots, \beta_{2k}^n \rangle$.

The next two lemmas are proven in [16]. Lemma 3.3 states that the coefficients $\beta_1^n, \ldots, \beta_{2k}^n$ have limiting values as $n \to \infty$ and that the rate of convergence to these values depends on the difference between the orders of correction k on the left end and m on the right end. Lemmas 3.3, 3.4, and 3.5 will be used in the proof of Theorem 3.6, below.

Lemma 3.3 (Rokhlin) *Suppose that* k *and* m *are positive integers with* $k < m$, *the function* s *is as specified in Lemma 3.2, and for each positive integer* n, *the coefficients* $\beta^n = \langle \beta_1^n, \ldots, \beta_{2k}^n \rangle$ *are the solution to the linear system of Eqs. (3.16). Then there exist coefficients* $\beta = \langle \beta_1, \ldots, \beta_{2k} \rangle$ *and a constant* $c > 0$ *such that*

$$|\beta_i^n - \beta_i| < \frac{c}{n^{m-k}}$$

for $i = 1, \ldots, 2k$ *and all* n.

Lemma 3.4 (Rokhlin) *Suppose that the function* s *is one of the singular functions specified in Lemma 3.2,* k *is a positive integer, and* $\varphi \in C^k[0, b]$ *satisfies*

$$0 = \varphi(0) = \varphi'(0) = \cdots = \varphi^{(k)}(0).$$

Then the function $w = \varphi \cdot s$ *is defined on the closed interval* $[0, b]$ *and*

$$0 = w(0) = w'(0) = \cdots = w^{(k-1)}(0).$$

For the function $w = \varphi \cdot s$, whose kth derivative is unbounded near 0, the simplified Euler-Maclaurin error expression

$$E_n^k(w) = -h^k \frac{B_k}{k!} (b - a) w^{(k)}(\xi),$$

with $0 < \xi < b$, is not useful. In this case we substitute the following bound.

Lemma 3.5 *Suppose that* k *and* n *are positive integers and the function* $w : [0, b] \to \mathcal{R}$ *is as specified in Lemma 3.4, above. Then the error* $E_n^k(w)$ *of the Euler-Maclaurin formula (Eq. 3.4) has the bound*

$$E_n^k(w) < h^k \sup_{x \in [0,1]} \left| \frac{B_k(x) - B_k}{k!} \right| \cdot \int_0^b \left| w^{(k)}(x) \right| dx. \tag{3.20}$$

Thus, there is a constant $c_k > 0$, independent of n, b, and the function w, such that

$$\left| T_n''(w,b) - \int_0^b w(x)\,dx \right| < h^k\, c_k \int_0^b \left| w^{(k)}(x) \right| dx, \tag{3.21}$$

where $h = b/n$.

Proof. The bound (3.20) is immediate from the definition of E_n^k given in Eq. (3.4). The error bound (3.21) then follows from the combination of the Euler-Maclaurin formula, the observation that $D_n^k(w,0) = 0$, the definition of T_k'' (Eq. 3.14), and bound (3.6). □

The following theorem is the foundation for the corrected trapezoidal rules for singular functions. It is a slight generalization of a theorem found in Rokhlin [16], in that the error bound is established for integrals taken on subintervals $[0, \frac{i}{n}b]$, for $i = 1, \ldots, n$, of the interval $[0, b]$, provided that the trapezoidal points' spacing $h = b/n$ is preserved.

Theorem 3.6 *Suppose that k and m are positive integers with $2 \le k < m$ and the function $f : (0, b] \to \mathcal{R}$ is given by the formula $f = \phi \cdot s + \psi$, where $\phi, \psi \in C^m[0,b]$ and the function s is as specified in Lemma 3.2. Further suppose that for each positive integer n, β^n is the solution of Eqs. (3.16) and $T_n^{\beta^n}$ is defined by Eq. (3.15). Then there exists $c > 0$ independent of b and the function f such that*

$$\left| T_i^{\beta^i}(f, i/n \cdot b) - \int_0^{\frac{i}{n}b} f(x)\,dx \right| < c\, b\, h^{k-1} \sup_{\xi \in [0,b]} \left(\left| \phi^{(k)}(\xi) \right| + \left| \psi^{(k)}(\xi) \right| \right) \tag{3.22}$$

for all n and $i = 1, \ldots, n$.

Proof. We write the function f as a sum of two functions; one is integrated exactly by the quadrature and the other has several zero derivatives at 0.

Let $P(\varphi)$ denote the k-term Taylor expansion of a function $\varphi \in C^m[0,b]$ about $x = 0$:

$$P(\varphi)(x) = \sum_{j=0}^{k-1} \frac{\varphi^{(j)}(0)}{j!}\, x^j.$$

Now we define $\phi_r = \phi - P(\phi)$ and $\psi_r = \psi - P(\psi)$ so we have

$$0 = \phi_r(0) = \psi_r(0) = \ldots = \phi_r^{(k-1)}(0) = \psi_r^{(k-1)}(0). \tag{3.23}$$

We further define $f_p = P(\phi) \cdot s + P(\psi)$ and $f_r = \phi_r \cdot s + \psi_r$, so that $f = f_p + f_r$, and we let $b_i = \frac{i}{n}b$. Now we bound the error by the inequality

$$\begin{aligned} \left| T_i^{\beta^i}(f, b_i) - \int_0^{b_i} f(x)\,dx \right| \;<\; & \left| T_i^{\beta^i}(f_p, b_i) - \int_0^{b_i} f_p(x)\,dx \right| \\ & + \left| T_i^{\beta^i}(f_r, b_i) - \int_0^{b_i} f_r(x)\,dx \right|, \end{aligned}$$

where by Eqs. (3.16) the first term on the right vanishes. By the definition of $T_i^{\beta^i}$ (Eq. 3.15), the second term satisfies

$$\left|T_i^{\beta^i}(f_r, b_i) - \int_0^{b_i} f_r(x)\,dx\right| < \left|T_i''(f_r, b_i) - \int_0^{b_i} f_r(x)\,dx\right| + \left|\sum_{j=1}^{2k} \beta_j^i\, f_r(\chi_j h)\right|.$$

By Eq. (3.23) and Lemma 3.4 the function $f_r = \phi_r \cdot s + \psi_r$ vanishes at 0 and has vanishing derivatives,

$$0 = f_r(0) = f_r'(0) = \ldots = f_r^{(k-2)}(0),$$

from which, in combination with error bound (3.21), we obtain

$$\left|T_i''(f_r, b_i) - \int_0^{b_i} f_r(x)\,dx\right| < h^{k-1}\, c_k \int_0^b \left|f_r^{(k-1)}(x)\right| dx \tag{3.24}$$

for some constant c_k independent of n, b, i, and the function f_r. For the remaining term, we define $M_\phi = k! \sup\left|\phi^{(k)}(x)\right|$ and $M_\psi = k! \sup\left|\psi^{(k)}(x)\right|$, both suprema taken for $x \in [0, b]$, therefore finite, and $M_\beta = \sup \beta_j^i$, taken over positive integers i and $j = 1, \ldots, 2k$. M_β is finite by Lemma 3.3, and we obtain

$$\begin{aligned}\left|\sum_{j=1}^{2k} \beta_j^i\, f_r(\chi_j\, h)\right| &\leq M_\beta \sum_{j=1}^{2k} |f_r(\chi_j\, h)| \\ &\leq M_\beta \sum_{j=1}^{2k} \left(M_\phi\, (\chi_j\, h)^k\, |s(\chi_j\, h)| + M_\psi\, (\chi_j\, h)^k\right) \\ &< M_\beta\, (2k)\ (M_\phi + M_\psi)\ h^{k-1}, \end{aligned} \tag{3.25}$$

for sufficiently large n. Combining bounds (3.24) and (3.25) yields (3.22). □

Note that Theorem 3.6 establishes that the quadrature $T_n^{\beta^n}(f, b)$ converges to the integral of f on the fixed interval $[0, b]$, with order of convergence at least $k - 1$. Additionally, however, it establishes the same convergence on subintervals $[0, \frac{i}{n} b]$ of $[0, b]$ with correspondingly fewer quadrature points. This characteristic is essential for proper treatment of non-periodic integral operators; in fact, it assures uniform convergence of the quadratures T_{ni}^k defined in Eq. (3.19).

Corollary 3.7 *Suppose that k and m are positive integers with $k < m$ and that the kernel $K : [a, b] \times [a, b] \to \mathcal{R}$ is given by the formula*

$$K(x, t) = \phi(x, t)\, s(|x - t|) + \psi(x, t)$$

where $\phi, \psi \in C^m([a, b] \times [a, b])$ and s is as specified in Lemma 3.2. We further suppose $f \in C^m[a, b]$ and define the function $G : [a, b] \times [a, b] \to \mathcal{R}$ by the formula $G(x, t) = K(x, t)\, f(t)$. There exists $c_k > 0$ such that

$$\left|T_{ni}^k(G) - \int_{x_i}^b G(x, t)\, f(t)\, dt\right| < \frac{c_k}{n^{k-1}} \tag{3.26}$$

for all n and $i = 0, 1, \ldots, n$.

Proof. We observe that M_ϕ and M_ψ defined by the formulae

$$M_\phi = \sup_{x,t\in[a,b]} \frac{\partial^k \phi(x,t)}{\partial t^k}$$
$$M_\psi = \sup_{x,t\in[a,b]} \frac{\partial^k \psi(x,t)}{\partial t^k}$$

are finite, then apply Theorem 3.6. □

The quadratures T_{ni}^k achieve uniform convergence and thus, combined with the kernel K and the coefficient p, represent an operator R which approximates the integral operator $\mathcal{K}$ defined above. An issue that arises in computing Rf for a function f, however, is that T_{ni}^k requires the values of f at non-equispaced points in the interval $[a,b]$ in addition to the points $a, a+h, \ldots, b-h, b$. This issue can be handled by using k-point interpolation, for $f \in C^m[a,b]$. We have the estimate

$$\left| f(\xi) - \sum_{i=1}^{k} c_i(\xi)\, f(a + (i + j_\xi)\, h) \right| < \frac{c}{n^k},$$

where $c_1(\xi), \ldots, c_k(\xi)$ are Lagrange interpolation coefficients and j_ξ is chosen such that $\xi \in [a + (1 + j_\xi)\, h, a + (k + j_\xi)\, h]$. Replacing $f(\xi)$ in $T_{ni}^k(Kf)$ by the interpolation preserves our error estimates.

3.4 Numerical Examples

In this section we present numerical examples of the corrected quadrature rules applied to differentiable and singular integrands. FORTRAN routines were written which incorporate the corrections developed in Section 3.2, and the quadratures were computed in double-precision arithmetic on a Sun Sparcstation 1. The correction weights themselves, which are given in the chapter appendix, were computed exactly with Maple, and in quadruple-precision arithmetic in FORTRAN running on a DEC microVAX.

Table 3.1 shows the errors in using quadratures of order 4, 8, and 12 to approximate the integral of a smooth function, comparing the equispaced corrected trapezoidal rule to the "crowded" corrected trapezoidal rule. Errors from the uncorrected trapezoidal rule are also shown for comparison. We make several observations:

1. Full single or double precision accuracy is easily achievable with the higher-order rules.

2. The observed rate of convergence matches that expected quite closely, as can be seen by comparing the errors for various n.

Table 3.1: *The quadrature rules ${}_1T_n^m$ of Eq. (3.9) and ${}_2T_n^m$ of Eq. (3.12) are used to compute $\int_0^1[\cos(21x)+\sin(22x)]\,dx$ and relative errors are shown for various values of n, m.*

	Trapez.	Equispaced rule ${}_1T_n^m$		
n	Rule	$m=4$	$m=8$	$m=12$
10	−0.397E+00	−0.235E+00	0.128E+01	−0.499E+01
20	−0.936E−01	0.840E−02	−0.119E−01	0.385E−02
40	−0.231E−01	0.137E−02	−0.181E−04	−0.226E−05
80	−0.575E−02	0.108E−03	0.863E−07	−0.847E−10
160	−0.144E−02	0.733E−05	0.617E−09	0.676E−13
320	−0.359E−03	0.474E−06	0.286E−11	−0.189E−15
640	−0.897E−04	0.301E−07	0.966E−14	−0.227E−14
1280	−0.224E−04	0.190E−08	−0.170E−14	−0.151E−14

	"Crowded" rule ${}_2T_n^m$		
n	$m=4$	$m=8$	$m=12$
10	0.425E−02	0.103E−04	0.119E−06
20	0.133E−02	0.866E−06	0.653E−09
40	0.109E−03	0.462E−08	−0.420E−11
80	0.748E−05	0.201E−10	0.121E−11
160	0.486E−06	0.793E−13	0.810E−12
320	0.309E−07	0.000E+00	0.863E−13
640	0.195E−08	−0.227E−14	−0.172E−12
1280	0.122E−09	−0.170E−14	0.130E−12

3. The "crowded" rules get a substantial jump on the equispaced rules and show small errors almost as soon as the integrand is resolved by the quadrature points.

4. The crowded rules ultimately achieve somewhat less precision than the equispaced rules, due to roundoff error resulting from the large correction weights, but this effect is important only if full double-precision accuracy is required.

In summary, the corrected rules should be preferred to the trapezoidal rule whenever high accuracy is desirable. The "crowded" corrected trapezoidal rule of order 12 performs very well.

Table 3.2 shows the errors from using the corrected trapezoidal rule for singular functions to compute the integral of an integrand with logarithm singularity. The rule was applied for various numbers of quadrature points n and various k, and for equispaced and half-Chebyshev $\chi_1, \ldots, \chi_{2k}$. In each case the correction

Table 3.2: *The quadrature rule* T_n^β *of Eq. (3.15) used to compute* $\int_0^1[\cos(21x) + \sin(22x)+\log(x)(\cos(23x)+\sin(24x))]\,dx$ *and relative errors are shown for various values of* n, k. *The right end is corrected to eighth order* $(m = 8)$.

	Equispaced points $\chi_1, \ldots, \chi_{2k}$			
n	$k = 2$	$k = 3$	$k = 4$	$k = 5$
10	−0.548E+00	−0.219E−01	0.429E−01	−0.608E−02
20	0.386E−02	−0.657E−02	0.506E−03	0.394E−04
40	0.403E−02	−0.242E−03	−0.169E−05	0.614E−06
80	0.663E−03	−0.341E−05	−0.236E−06	0.482E−08
160	0.964E−04	0.381E−06	−0.971E−08	0.809E−11
320	0.139E−04	0.545E−07	−0.363E−09	−0.593E−12
640	0.202E−05	0.509E−08	−0.135E−10	0.258E−14
1280	0.292E−06	0.414E−09	−0.497E−12	0.861E−15

	Half-Chebyshev points $\chi_1, \ldots, \chi_{2k}$			
n	$k = 2$	$k = 3$	$k = 4$	$k = 5$
10	−0.510E−01	−0.667E−02	−0.503E−02	−0.110E−02
20	0.143E−02	0.599E−04	−0.166E−04	0.553E−05
40	0.484E−03	0.615E−05	0.732E−06	0.877E−07
80	0.750E−04	0.349E−06	0.351E−07	0.653E−09
160	0.108E−04	0.203E−07	0.132E−08	−0.312E−12
320	0.156E−05	0.125E−08	0.492E−10	−0.148E−12
640	0.227E−06	0.796E−10	0.183E−11	−0.445E−14
1280	0.327E−07	0.519E−11	0.670E−13	−0.100E−14

weights $\beta = \langle\beta_1, \ldots, \beta_{2k}\rangle$ were chosen to be the limiting values (of Lemma 3.3), given in the chapter appendix. Note that Theorem 3.6 assures us of convergence of order at least $k - 1$. We observe:

1. Although we have proven that the order of convergence is at least $k - 1$, the actual order of convergence appears to lie between k and $k + 1$ and the convergence pattern is somewhat irregular.

2. The correction weights from equispaced points $\chi_1, \ldots, \chi_{2k}$ are much larger than those from half-Chebyshev points; nevertheless, both versions of the quadrature rule perform well and lead to nearly full precision accuracy in practice.

The uncorrected trapezoidal rule, with the left end omitted, gives very slow convergence for this problem and is not practical for achieving high accuracy.

Our final examples of the corrected trapezoidal rules, in Table 3.3, demonstrate the uniform convergence of the quadratures for various integral operators.

Table 3.3: *The quadrature rule T^k_{ni} of Eq. (3.19) is used to compute $F(x) = \int_0^1 [\cos(21xt) + \sin(22xt) + s(|x-t|)(\cos(23xt) + \sin(24xt))]\,dt$, with $x = i/n$ for $i = 0, 1, \ldots, n$. The function s is singular and has one of three forms given below. The relative $\mathcal{L}^2$ errors for the quadratures with $k = 4$ are shown.*

	Three Choices of Integrand $f(x,t)$		
n	$s(x) = \log(x)$	$s(x) = x^{-1/2}$	$s(x) = x^{1/2}$
10	0.302E−03	0.134E−03	0.482E−04
20	0.527E−05	0.187E−05	0.579E−06
40	0.681E−07	0.487E−07	0.206E−07
80	0.573E−08	0.266E−08	0.611E−09
160	0.283E−09	0.132E−09	0.181E−10
320	0.118E−10	0.613E−11	0.547E−12
640	0.454E−12	0.300E−12	0.169E−13
1280	0.216E−13	0.155E−12	0.271E−14

In these examples, we approximate the integral $F_i = \int_0^1 f(x_i, t)\,dt$ for $x_i = i/n$ and $i = 0, 1, \ldots, n$, by the quadrature $T^k_{ni}(f)$, defined in Eq. (3.19). The table shows the relative $\mathcal{L}^2$ error of the approximations, defined by the formula

$$e_{\mathcal{L}^2} = \left(\frac{\sum_{i=0}^n [T^k_{ni}(f) - F_i]^2}{\sum_{i=0}^n F_i^2} \right)^{1/2}. \tag{3.27}$$

By taking ratios of the errors for different n, it can be seen that (for $k = 4$) the rate of convergence is uniformly of order at least 4. For the logarithm singularity, the order of convergence is about 4.5, and for the square root singularity, it is nearly 5. In each case, $n = 20$ produces roughly single-precision accuracy.

3.5 Appendix: Quadrature Weights

This appendix contains various weights and correction coefficients used in the quadratures of the chapter. The coefficients in the finite difference expressions for odd-numbered derivatives, which appear in the Euler-Maclaurin formula, as well as the corresponding correction coefficients are tabulated (Tables 3.4 and 3.5 and 3.6). The limiting values of the correction weights used for the singularities $s(x) = \log(x)$ and $s(x) = x^\alpha$ for $\alpha = \pm\frac{1}{2}$ are shown for equispaced correction points (Table 3.7) and for half-Chebyshev correction points. The correction weights for finite n, which are very numerous, are not shown.

Table 3.4: *Finite-difference coefficients for odd-numbered derivatives to various orders of approximation are shown. The table entries are coefficients from the formula* $h^j f^{(j)}(x) = (1/d)\sum_{i=0}^{m-2} c_i\, f(x + i\,h) + \mathrm{O}(h^m)$, *for* $j = 1, 3, 5, 7, 9$ *and* $m = 4, 6, 8, 10, 12$.

	$hf^{(1)}(x)$					$h^5 f^{(5)}(x)$		
	$\mathrm{O}(h^4)$	$\mathrm{O}(h^6)$	$\mathrm{O}(h^8)$	$\mathrm{O}(h^{10})$	$\mathrm{O}(h^{12})$	$\mathrm{O}(h^8)$	$\mathrm{O}(h^{10})$	$\mathrm{O}(h^{12})$
d	2	12	60	840	2520	2	6	288
c_0	−3	−25	−147	−2283	−7381	−7	−81	−8591
c_1	4	48	360	6720	25200	40	575	72492
c_2	−1	−36	−450	−11760	−56700	−95	−1790	−278313
c_3		16	400	15680	100800	120	3195	640752
c_4		−3	−225	−14700	−132300	−85	−3580	−979878
c_5			72	9408	127008	32	2581	1039656
c_6			−10	−3920	−88200	−5	−1170	−774402
c_7				960	43200		305	399408
c_8				−105	−14175		−35	−136347
c_9					2800			27788
c_{10}					−252			−2565

	$h^3 f^{(3)}(x)$				$h^7 f^{(7)}(x)$		$h^9 f^{(9)}(x)$
	$\mathrm{O}(h^6)$	$\mathrm{O}(h^8)$	$\mathrm{O}(h^{10})$	$\mathrm{O}(h^{12})$	$\mathrm{O}(h^{10})$	$\mathrm{O}(h^{12})$	$\mathrm{O}(h^{12})$
d	2	8	240	30240	2	24	2
c_0	−5	−49	−2403	−420475	−9	−605	−11
c_1	18	232	13960	2876868	70	5628	108
c_2	−24	−461	−36706	−9389763	−238	−23583	−477
c_3	14	496	57384	19227792	462	58632	1248
c_4	−3	−307	−58280	−27098442	−560	−95802	−2142
c_5		104	39128	27147960	434	107520	2520
c_6		−15	−16830	−19395138	−210	−83958	−2058
c_7			4216	9693648	58	45048	1152
c_8			−469	−3229227	−7	−15897	−423
c_9				645412		3332	92
c_{10}				−58635		−315	−9

Table 3.5: *Equispaced endpoint corrections transform the trapezoidal rule into a high-order quadrature for functions with several continuous derivatives. The quadrature rules are given by the formula* $\int_a^b f(x)\,dx = T_n(f) + (h/d)\sum_{i=0}^{m-2} c_i\,[f(a+i\,h) + f(b-i\,h)] + \mathrm{O}(h^m)$, *where* $h = (b-a)/n$ *and* $T_n(f) = h\,[\frac{1}{2}f(a) + f(a+h) + \cdots + f(b-h) + \frac{1}{2}f(b)]$.

	$\mathrm{O}(h^2)$	$\mathrm{O}(h^4)$	$\mathrm{O}(h^6)$	$\mathrm{O}(h^8)$	$\mathrm{O}(h^{10})$	$\mathrm{O}(h^{12})$
d	1	24	1440	120960	7257600	958003200
c_0	0	−3	−245	−23681	−1546047	−216254335
c_1		4	462	55688	4274870	679543284
c_2		−1	−336	−66109	−6996434	−1412947389
c_3			146	57024	9005886	2415881496
c_4			−27	−31523	−8277760	−3103579086
c_5				9976	5232322	2939942400
c_6				−1375	−2161710	−2023224114
c_7					526154	984515304
c_8					−57281	−321455811
c_9						63253516
c_{10}						−5675265

Table 3.6: *Corrections crowded into the intervals* $[a, a+h)$ *and* $(b-h, b]$ *convert the trapezoidal rule into a high-order quadrature for functions with several continuous derivatives. The quadrature rules have the form* $\int_a^b f(x)\,dx = T_n(f) + (h/d)\sum_{i=0}^{m-2} c_i\,[f(a+\frac{i}{m-1}h) + f(b-\frac{i}{m-1}h)] + \mathrm{O}(h^m)$, *where* $h = (b-a)/n$ *and* $T_n(f) = h\,[\frac{1}{2}f(a) + f(a+h) + \cdots + f(b-h) + \frac{1}{2}f(b)]$. *These rules have smaller error constants, but larger coefficients, than the equispaced corrected trapezoidal rules.*

	$\mathrm{O}(h^2)$	$\mathrm{O}(h^4)$	$\mathrm{O}(h^6)$	$\mathrm{O}(h^8)$	$\mathrm{O}(h^{10})$	$\mathrm{O}(h^{12})$
d	1	8	288	17280	89600	87091200
c_0	0	−3	−125	−7889	−41943	−41374135
c_1		4	30	13832	−372570	4717178004
c_2		−1	240	−57421	2898654	−43825028709
c_3			−190	133056	−9112466	180245487576
c_4			45	−130067	15663360	−428859839166
c_5				58744	−15657582	649926182400
c_6				−10255	9131810	−649910688834
c_7					−2897574	428850243624
c_8					388311	−180220260891
c_9						43821791596
c_{10}						−4703691465

Table 3.7: *The limiting values of the correction weights, at equispaced points, for the corrected trapezoidal rules for singular functions are tabulated. The corrected quadrature rules are given by the formula* $\int_a^b f(x)\,dx \doteq T_n'(f) + h\sum_{i=1}^{2k}\beta_i f(a+\frac{i}{2k}h) + (h/d)\sum_{i=0}^{m-2} c_i f(b-\frac{i}{m-1}h)$, *where* $h=(b-a)/n$, $T_n'(f) = h\,[f(a+h)+f(a+2h)+\cdots+f(b-h)+\frac{1}{2}f(b)]$, *and the coefficients* d *and* $c_0,\ldots,c_{m-2}$ *are given by Table 3.6.*

Singularity $s(x) = \log(x-a)$

	$k=2$	$k=3$
β_1	0.1601298415357170E+01	0.2228766018460087E+01
β_2	−0.3382558521919485E+01	−0.1231212070062612E+02
β_3	0.3627888464434128E+01	0.3157965997308673E+02
β_4	−0.1346628357871812E+01	−0.3840391590010434E+02
β_5		0.2267350459115253E+02
β_6		−0.5265893981968890E+01

	$k=4$	$k=5$
β_1	0.3093483401777122E+01	0.4335882754006495E+01
β_2	−0.3101788376740780E+02	−0.6931374375731333E+02
β_3	0.1362059155903270E+03	0.4590379369965628E+03
β_4	−0.3147474808724213E+03	−0.1630223863113166E+04
β_5	0.4215054127612634E+03	0.3509093316535128E+04
β_6	−0.3287854038787327E+03	−0.4819809483739626E+04
β_7	0.1388011671370668E+03	0.4271587513841620E+04
β_8	−0.2455521037187227E+02	−0.2374001155274844E+04
β_9		0.7547424129498612E+03
β_{10}		−0.1049488171922296E+03

Singularity $s(x) = (x-a)^{-1/2}$

	$k=2$	$k=3$
β_1	0.3338954623777353E+01	0.5156862384200115E+01
β_2	−0.1036918555513964E+02	−0.3718025418572306E+02
β_3	0.1238817390561390E+02	0.1050278047880726E+03
β_4	−0.4857942974251604E+01	−0.1388205130918806E+03
β_5		0.8799431664143500E+02
β_6		−0.2167821653610409E+02

Table 3.7: (continued)

Singularity $s(x) = (x-a)^{-1/2}$ (continued)

	$k=4$	$k=5$
β_1	0.7889576157976986E+01	0.1191640221424121E+02
β_2	−0.1014839102693306E+03	−0.2431211398143441E+03
β_3	0.4982052353339497E+03	0.1823927314310889E+04
β_4	−0.1241778604543411E+04	−0.7083666085060807E+04
β_5	0.1751093993580452E+04	0.1635686640624526E+05
β_6	−0.1419085152097947E+04	−0.2381679753707451E+05
β_7	0.6179863268019096E+03	0.2217597560756524E+05
β_8	−0.1123274649636003E+03	−0.1285030321889555E+05
β_9		0.4231931207608981E+04
β_{10}		−0.6062289570994023E+03

Singularity $s(x) = (x-a)^{1/2}$

	$k=2$	$k=3$
β_1	0.1076226369733505E+01	0.1403733895743621E+01
β_2	−0.1472473730142097E+01	−0.6106269754659710E+01
β_3	0.1382935017750346E+01	0.1458214411894674E+02
β_4	−0.4866876573417537E+00	−0.1639615278631818E+02
β_5		0.8952282755716858E+01
β_6		−0.1935738229429337E+01

	$k=4$	$k=5$
β_1	0.1761384695584808E+01	0.2301960768321394E+01
β_2	−0.1382118344852977E+02	−0.2856244752621036E+02
β_3	0.5459150117813370E+02	0.1642204330046712E+03
β_4	−0.1173574845498706E+03	−0.5199121092937581E+03
β_5	0.1507790199321616E+03	0.1011593120489479E+04
β_6	−0.1147784911579322E+03	−0.1264688737067177E+04
β_7	0.4762309598361213E+02	0.1027260908792144E+04
β_8	−0.8297842633159577E+01	−0.5280676861522366E+03
β_9		0.1570202164406574E+03
β_{10}		−0.2066565945589198E+02

Table 3.8: *The limiting values of the correction weights, at half-Chebyshev points, for the corrected trapezoidal rules for singular functions are tabulated. The quadratures are given by* $\int_a^b f(x)\,dx \doteq T_n'(f) + h\sum_{i=1}^{2k}\beta_i f(a+\chi_i h) + (h/d)\sum_{i=0}^{m-2} c_i f(b-\frac{i}{m-1}h)$, *where* $h=(b-a)/n$, $\chi_i = 1-\cos((2i-1)\pi/(8k))$, $T_n'(f) = h\,[f(a+h)+f(a+2h)+\cdots+f(b-h)+\frac{1}{2}f(b)]$, *and the coefficients* d *and* $c_0,\ldots,c_{m-2}$ *are given by Table 3.6.*

Singularity $s(x) = \log(x-a)$

	$k=2$	$k=3$
β_1	0.5603992216960789E−01	0.2395413023703419E−01
β_2	0.2870630583396921E+00	0.1743092878178258E+00
β_3	0.2563533589170907E+00	−0.1631724711422584E+00
β_4	−0.9945633942639063E−01	0.9387069242218954E+00
β_5		−0.5880276430135620E+00
β_6		0.1142297718790650E+00

	$k=4$	$k=5$
β_1	0.3338537800974016E−01	0.1916264301337224E−02
β_2	−0.1737066212781738E+00	0.1567268574189051E+00
β_3	0.1198836762516400E+01	−0.4334786391079948E+00
β_4	−0.2591374963545573E+01	0.7215818540402889E+00
β_5	0.3883697343902998E+01	0.1482966328206821E+01
β_6	−0.2430220427598129E+01	−0.6595938685246877E+01
β_7	0.6270755476376530E+00	0.1169258889232862E+02
β_8	−0.4769301964491491E−01	−0.9958531304701892E+01
β_9		0.4103424553775595E+01
β_{10}		−0.6712561210148048E+00

Singularity $s(x) = (x-a)^{-1/2}$

	$k=2$	$k=3$
β_1	0.8353164416920675E−01	0.3888639530245085E−01
β_2	0.1911747004918826E+00	0.8640672385266692E−01
β_3	0.3657110693973134E+00	0.3880300543933000E−01
β_4	−0.1404174140584028E+00	0.7069014304227469E+00
β_5		−0.4537263116382184E+00
β_6		0.8272875662102377E−01

Table 3.8: (continued)

Singularity $s(x) = (x-a)^{-1/2}$ (continued)

	$k=4$	$k=5$
β_1	0.2823063071751857E−01	0.1604517803043481E−01
β_2	−0.1016030472473191E+00	−0.2672763418512212E−01
β_3	0.8957498450168363E+00	0.4952499866825667E+00
β_4	−0.1944630011805141E+01	−0.1916835699721412E+01
β_5	0.3079540564910779E+01	0.6247645777407226E+01
β_6	−0.1835281366554477E+01	−0.1233093115169777E+02
β_7	0.3822511312525950E+00	0.1631924962150504E+02
β_8	−0.4257746290791726E−02	−0.1237865585994330E+02
β_9		0.4849695950284900E+01
β_{10}		−0.7747361683625560E+00

Singularity $s(x) = (x-a)^{1/2}$

	$k=2$	$k=3$
β_1	0.3685550676579945E−01	0.1729631990952830E−01
β_2	0.3411680628277778E+00	0.2024441114244492E+00
β_3	0.2026521530973060E+00	−0.2155157764926719E+00
β_4	−0.8067572269088319E−01	0.9910326020047757E+00
β_5		−0.6154646588009935E+00
β_6		0.1202074019549121E+00

	$k=4$	$k=5$
β_1	0.7512196681423955E−01	−0.6317627640571476E−01
β_2	−0.4538891994788910E+00	0.7370068868212822E+00
β_3	0.2050275204711021E+01	−0.2828700163634487E+01
β_4	−0.4084723895152133E+01	0.6657003574957570E+01
β_5	0.5503263764388441E+01	−0.8199869710972154E+01
β_6	−0.3512451256619964E+01	0.4165258132856930E+01
β_7	0.1038747521068656E+01	0.3554455470298683E+01
β_8	−0.1163441057313696E+00	−0.5923097194990581E+01
β_9		0.2913478385611404E+01
β_{10}		−0.5123591045429318E+00

Chapter 4

Vector Space Bases

In this chapter we construct a class of bases, analogous to the multi-wavelet bases of Chapter 2, that transform the dense $n \times n$-matrices resulting from the discretization of integral equations into sparse matrices with order $O(n \log n)$ non-zero elements (to arbitrary finite precision). In these bases, the inverse matrices are also sparse, and are obtained in order $O(n \log^2 n)$ operations by the classical Schulz method.

A recent paper [3] introduces the use of wavelets for the application of an integral operator to a function in $O(n \log n)$ operations, where n is the number of points in the discretization of the function. In the present chapter, rather than employ a wavelet basis for $\mathcal{L}^2(\mathcal{R})$, we construct bases in which we represent the operators discretized by Nyström's method. If $S = \{x_1, \ldots, x_n\}$ denotes the points of the discretization, we define a class of wavelet-like bases for the finite-dimensional space of functions defined on S.

4.1 Wavelet Bases

4.1.1 Properties of the Bases

Given a set of n distinct points $S = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{R}$ (the discretization) we construct an orthonormal basis for the n-dimensional space of functions defined on S. For simplicity, we assume that $n = k \cdot 2^l$, where k and l are natural numbers, and that $x_1 < x_2 < \cdots < x_n$. The basis has two fundamental properties:

1. All but k basis vectors have k vanishing moments, and

2. The basis vectors are non-zero on different scales.

Fig. 4.1 illustrates a matrix of basis vectors for $n = 128$ and $k = 4$. Each row represents one basis vector, with the dots depicting non-zero elements. The first k basis vectors are non-zero on $x_1, \ldots, x_{2k}$, the next k are non-zero on

$x_{2k+1}, \ldots, x_{4k}$, and so forth. In all, one half of the basis vectors are non-zero on $2k$ points from S, one fourth are non-zero on $4k$ points, one eighth are non-zero on $8k$ points, etc. Each of these $n/2 + n/4 + n/8 + \cdots + k = n - k$ basis vectors has k zero moments, *i.e.*, if $b = \langle b_1, \ldots, b_n \rangle$ is one of these vectors, then

$$\sum_{i=1}^{n} b_i \cdot {x_i}^j = 0, \qquad j = 0, 1, \ldots, k-1.$$

The final k vectors result from orthogonalization of the moments $\langle {x_1}^j, {x_2}^j, \ldots, {x_n}^j \rangle$ for $j = 0, 1, \ldots, k-1$.

These properties of local support and vanishing moments lead to efficient representation of functions which are smooth except at a finite set of singularities. The projection of such a function on a vector of this basis will be negligible unless the vector is non-zero near one of the singularities. As a simple example, we consider the function $f(x) = \log(x)$ on the interval $[0, 1]$ with the uniform discretization $x_i = i/n$. A hand calculation shows that for any $c > 0$, f may be interpolated on the interval $[c, 2c]$ by a polynomial of degree 7 with error bounded by 4^{-9}, or roughly single precision accuracy. If we choose $k = 8$ in constructing the basis, f will be represented to this accuracy by the k basis vectors non-zero on $x_1, \ldots, x_{2k}$, the k basis vectors non-zero on $x_1, \ldots, x_{4k}$, and so forth, down to

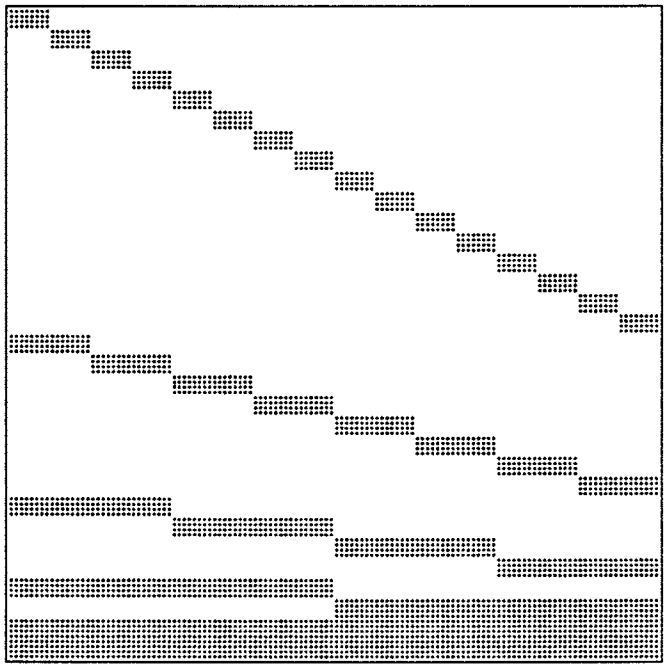

Figure 4.1: *The matrix represents a wavelet basis for a discretization with* 128 *points, for* $k = 4$. *Each row denotes one basis vector, with the dots depicting non-zero elements. All but the final* k *rows have* k *vanishing moments.*

the k basis vectors non-zero on $x_1, \ldots, x_n$, in addition to the k orthogonalized moment vectors. The number of non-negligible coefficients in the expansion of f in this basis grows logarithmically in n, the number of points of the discretization. Although this example is idealized, its behavior is representative of the general behavior of an analytic function near a singularity.

4.1.2 Construction of the Bases

The conditions of "local" support and zero moments determine the basis vectors uniquely (up to sign) if we require somewhat more moments to vanish. Namely, out of the k vectors non-zero on $x_1, \ldots, x_{2k}$, we require that one have k vanishing moments, a second have $k+1$, a third have $k+2$, and so forth, and the kth have $2k-1$ vanishing moments. We place the same condition on the k basis vectors non-zero on $x_{2k+1}, \ldots, x_{4k}$, and so on, for each block of k basis vectors among the $n-k$ basis vectors with zero moments.

We construct the basis by construction of a finite sequence of bases (shown in Fig. 4.2), each obtained by a number of orthogonalizations. The first basis results from $n/(2k)$ Gram-Schmidt orthogonalizations of $2k$ vectors each. In particular, the vectors $\langle {x_1}^j, \ldots, {x_{2k}}^j \rangle$ for $j = 0, \ldots, 2k-1$ are orthogonalized, the vectors $\langle {x_{2k+1}}^j, \ldots, {x_{4k}}^j \rangle$ for $j = 0, \ldots, 2k-1$ are orthogonalized, and so forth, up to the vectors $\langle {x_{n-2k+1}}^j, \ldots, {x_n}^j \rangle$ for $j = 0, \ldots, 2k-1$ which are orthogonalized.

Half of the n vectors of the first basis have at least k zero moments; in forming the second basis, these vectors are retained; the remaining $n/2$ basis vectors are transformed by an orthogonal transformation into basis vectors, each of which is non-zero on $4k$ of the points $x_1, \ldots, x_n$, and half of which have at least k vanishing moments. The orthogonal transformation results from $n/(4k)$ Gram-Schmidt orthogonalizations of $2k$ vectors each. Similarly, the third basis is obtained from the second basis by an orthogonal transformation that itself results from $n/(8k)$ Gram-Schmidt orthogonalizations of $2k$ vectors each. Before we can specify these orthogonalizations, we require some additional notation.

Suppose V is a matrix whose columns $v_1, \ldots, v_{2k}$ are linearly independent. We define W=Orth(V) to be the matrix which results from the column-by-column Gram-Schmidt orthogonalization of V. Namely, denoting the columns of W by $w_1, \ldots, w_{2k}$, we have

$$\begin{array}{l} \text{linear span}\{w_1, \ldots, w_i\} = \text{linear span}\{v_1, \ldots, v_i\} \\ {w_i}^T w_j = \delta_{ij} \end{array} \qquad i, j = 1, \ldots, 2k.$$

For a $2k \times 2k$-matrix V we let V^U and V^L denote two $k \times 2k$-matrices, V^U consisting of the upper k rows and V^L the lower k rows of V,

$$V = \begin{pmatrix} V^U \\ V^L \end{pmatrix}.$$

Now we proceed to the definition of the basis matrices. Given the set of points $S = \{x_1, \ldots, x_n\} \subset \mathcal{R}$ with $x_1 < \cdots < x_n$, where $n = k \cdot 2^l$, we define the $2k \times 2k$ moments matrices $M_{1,i}$ for $i = 1, \ldots, n/(2k)$ by the formula

$$M_{1,i} = \begin{pmatrix} 1 & x_{s_i+1} & \cdots & {x_{s_i+1}}^{2k-1} \\ 1 & x_{s_i+2} & \cdots & {x_{s_i+2}}^{2k-1} \\ \vdots & & & \vdots \\ 1 & x_{s_i+2k} & \cdots & {x_{s_i+2k}}^{2k-1} \end{pmatrix}, \tag{4.1}$$

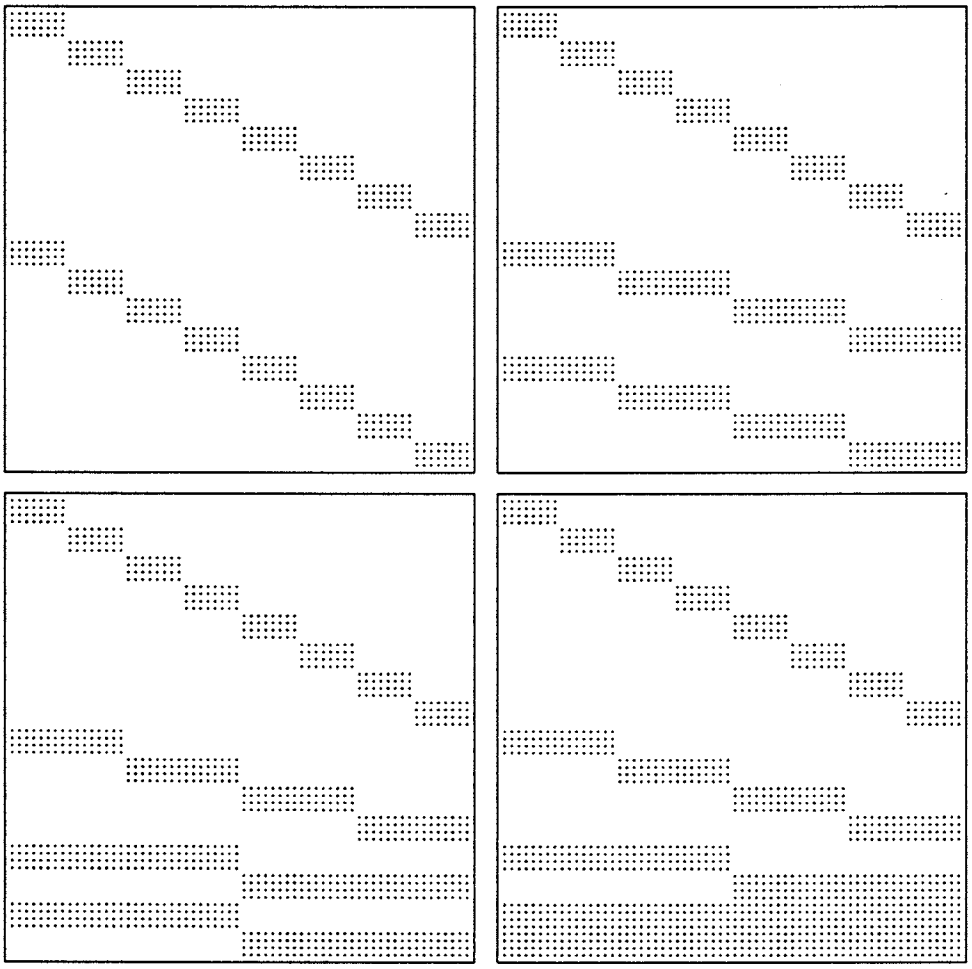

Figure 4.2: *Each of the four matrices represents one basis, as in Fig. 1. The upper-left matrix is formed by orthogonalizing moment vectors on blocks of* $2k$ *points. The upper-right matrix is obtained from the upper-left matrix by premultiplying by an orthogonal matrix which is the identity on the upper half. Similarly, the lower matrices are obtained by further orthogonal transformations. The lower-right matrix represents the wavelet basis for* $n = 64$, $k = 4$.

where $s_i = (i-1)2k$. The first basis matrix U_1 is given by the formula

$$U_1 = \begin{pmatrix} {U_{1,1}}^L & & & \\ & {U_{1,2}}^L & & \\ & & \ddots & \\ & & & {U_{1,n_1}}^L \\ {U_{1,1}}^U & & & \\ & {U_{1,2}}^U & & \\ & & \ddots & \\ & & & {U_{1,n_1}}^U \end{pmatrix},$$

where ${U_{1,i}}^T = \mathrm{Orth}(M_{1,i})$ and $n_1 = n/(2k)$. The second basis matrix is $U_2 U_1$, with U_2 defined by the formula

$$U_2 = \begin{pmatrix} I_{n/2} & \\ & U_2' \end{pmatrix}$$

where I_m is the $m \times m$ identity matrix and U_2' is given by the formula

$$U_2' = \begin{pmatrix} {U_{2,1}}^L & & & \\ & {U_{2,2}}^L & & \\ & & \ddots & \\ & & & {U_{2,n_2}}^L \\ {U_{2,1}}^U & & & \\ & {U_{2,2}}^U & & \\ & & \ddots & \\ & & & {U_{2,n_2}}^U \end{pmatrix},$$

where $n_2 = n/(4k)$, ${U_{2,i}}^T = \mathrm{Orth}(M_{2,i})$, and $M_{2,i}$ is given by

$$M_{2,i} = \begin{pmatrix} {U_{1,2i-1}}^U M_{1,2i-1} \\ {U_{1,2i}}^U M_{1,2i} \end{pmatrix}.$$

In general, the jth basis matrix, for $j = 2, \ldots, \log_2(n/k)$, is $U_j \cdots U_1$, with U_j defined by the formula

$$U_j = \begin{pmatrix} I_{n-n/2^{j-1}} & \\ & U_j' \end{pmatrix}$$

where U_j' is given by the formula

$$U_j' = \begin{pmatrix} {U_{j,1}}^L & & & \\ & {U_{j,2}}^L & & \\ & & \ddots & \\ & & & {U_{j,n_j}}^L \\ {U_{j,1}}^U & & & \\ & {U_{j,2}}^U & & \\ & & \ddots & \\ & & & {U_{j,n_j}}^U \end{pmatrix},$$

where $n_j = n/(2^j k)$, $U_{j,i}$ is given by

$$U_{j,i}{}^T = \mathrm{Orth}(M_{j,i}), \tag{4.2}$$

and $M_{j,i}$ is given by

$$M_{j,i} = \begin{pmatrix} {U_{j-1,2i-1}}^U M_{j-1,2i-1} \\ {U_{j-1,2i}}^U M_{j-1,2i} \end{pmatrix}. \tag{4.3}$$

The final basis matrix $U = U_l \cdots U_1$, where $l = \log_2(n/k)$, represents the wavelet basis of parameter k on $x_1, \ldots, x_n$.

Remark 4.1 The definitions given for the basis matrices are mathematical definitions only; in a numerical procedure, considerable roundoff error would be introduced by the orthogonalizations defined above. In the actual implementation, the matrices $M_{j,i}$ are shifted and scaled, resulting in a numerically stable procedure that is equivalent to the above definitions (in exact arithmetic). Details of this procedure are provided in Section 4.3.

It is apparent that the application of the matrix U to an arbitrary vector of length n may be accomplished in order $\mathrm{O}(n)$ operations by the application of $U_1, \ldots, U_l$ in turn. Similarly, $U^{-1} = U^T$ may be applied to a vector in order $\mathrm{O}(n)$ operations. Certain dense matrices, in particular those arising from integral operators, are sparse in the basis of U and their similarity transformations can be computed in $\mathrm{O}(n \log n)$ operations. These techniques are developed in the following sections.

We conclude this section with an illustration of the vectors of one basis from this class, in Fig. 4.3.

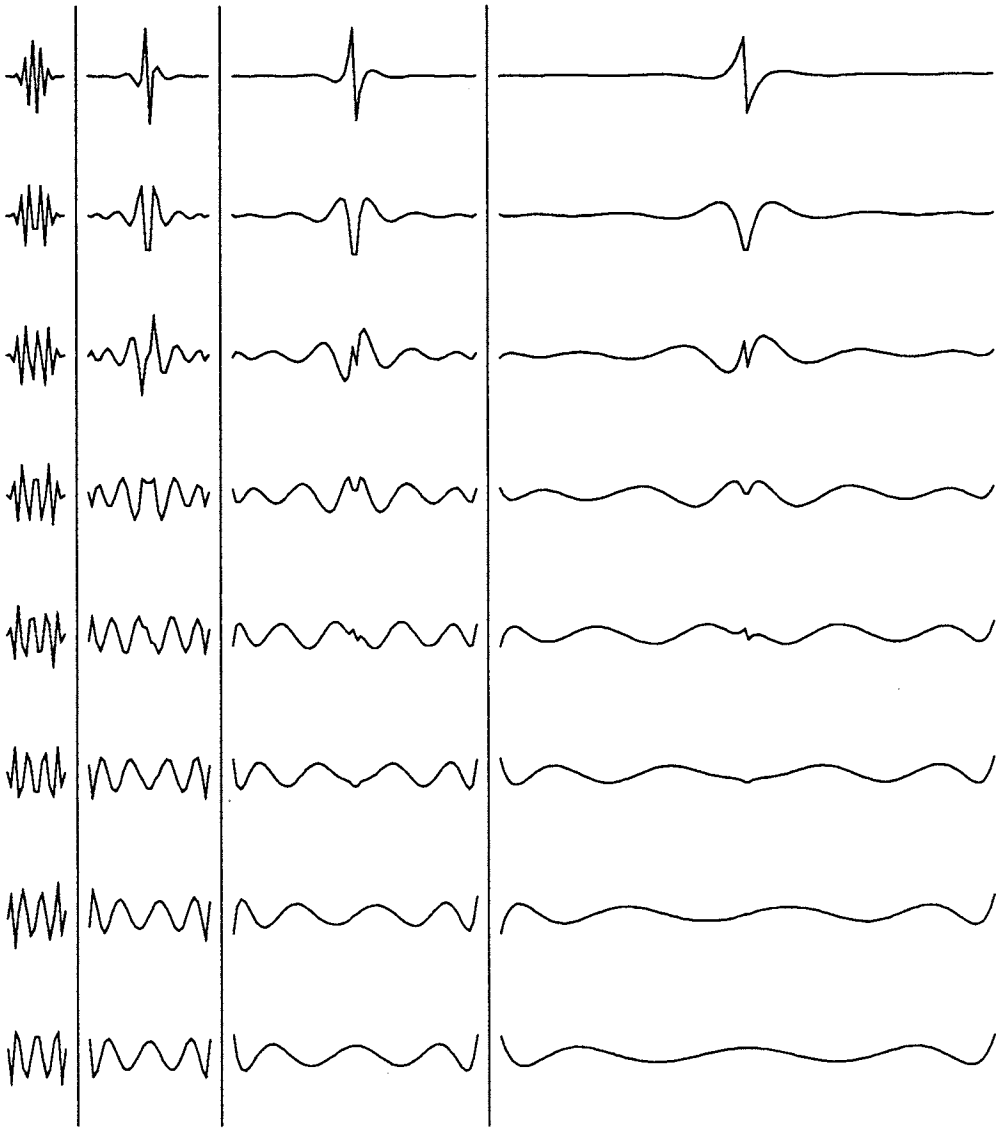

Figure 4.3: *Basis vectors on four scales are shown for the basis where* $n = 128$, *points* $x_1, \ldots, x_n$ *are equispaced, and* $k = 8$. *The first column of vectors consists of rows* 1–8 *of* U, *the second column consists of rows* 65–72, *etc. Note that half of the vectors are odd and half are even functions, and that the odd ones are generally discontinuous at their center.*

4.2 Second-Kind Integral Equations

4.2.1 Sparse Representation of Integral Operators

As in Chapters 2 and 3, we concern ourselves here with kernels $K = K(x,t)$ which are analytic except at $x = t$, where they may possess an integrable singularity. We discretize the integral operator $\mathcal{K}$,

$$(\mathcal{K}f)(x) = \int_a^b K(x,t)\, f(t)\, dt,$$

by Nyström's method, using an equispaced quadrature. Given $n \geq 2$, we define points $x_1, \ldots, x_n$ to be equispaced on the interval $[a,b]$,

$$x_i = a + (i-1)(b-a)/(n-1), \tag{4.4}$$

and define the elements T_{ij} of the $n \times n$-matrix T by the formula

$$T_{ij} = \begin{cases} \frac{1}{n-1} K(x_i, x_j) & i \neq j \\ 0 & i = j. \end{cases} \tag{4.5}$$

Note that the matrix $T = T(n)$ corresponds to a primitive, trapezoid-like quadrature discretization of the integral operator $\mathcal{K}$. The matrix T possesses the same smoothness properties as the kernel $K(x,t)$. Transformation of T by the bases developed in Section 4.1 produces a matrix that is sparse, to high precision. The number of elements is effectively bounded by order $O(n \log n)$.

When the matrix representing the quadrature corrections developed in Section 3.2 is added to T, producing high-order convergence to the integral operator, this bound remains valid.

The matrix T, transformed by the orthogonal $n \times n$-matrix U, can be decomposed into the sum of a sparse matrix and a matrix with small norm. That is, given $\epsilon > 0$, there exists $c_\epsilon > 0$, independent of n, such that the transformed matrix can be written in the form

$$UTU^T = V + E,$$

where the number of elements in $V = V(n)$ is bounded by $c_\epsilon\, n \log n$ and $E = E(n)$ is small: $\|E\| < \epsilon \|T\|$. This assertion can be proven very similarly to Lemmas 2.3 and 2.4; in fact, substitution of the finite sums which determine the elements of UTU^T for the integrals in those lemmas yields the assertion.

4.2.2 Solution via Schulz Method

The sparse matrix representing the integral operator also has a sparse inverse, as was the case for the multi-wavelet bases developed in Chapter 2. We invert

the matrix by the Schulz method, described in Section 2.4.3. Each iteration requires two matrix-matrix multiplications, where the multiplicands are sparse, and yields a sparse product. The method is quadratically convergent; the number of iterations, given by bound (2.23, grows only as the logarithm of the condition number.

4.2.3 Oscillatory Coefficients

We now consider a somewhat more general class of integral equations, in which the integral operator is given by the formula

$$(D\mathcal{K}f)(x) = p(x)\int_a^b K(x,t)\,f(t)\,dt,$$

where the kernel K is assumed to be smooth, but the coefficient function p can be oscillatory. In particular, we only restrict p to be positive. In terms of generality, these problems lie between the problems with smooth kernels (and constant coefficient) and those with arbitrary oscillatory kernels.

Writing the corresponding integral equation in operator form, we obtain the equation

$$(I - D\mathcal{K})f = g. \tag{4.6}$$

Although D is a diagonal operator, and $\mathcal{K}$ is smooth, it is clear that the discretization of the operator $D\mathcal{K}$ will not be a sparse matrix in wavelet coordinates. In this framework, it would appear that the constructions of Chapter 2 and this chapter are inapplicable. If we instead consider the operator $D^{1/2}\mathcal{K}D^{1/2}$, in which oscillations in the rows match those in the columns, it becomes clear that the construction of Section 4.1 can be revised. Rather than constructing basis functions orthogonal to low-order polynomials x^j, we can construct them to be orthogonal to $p(x)^{1/2}\,x^j$. The sole revision in our definition of basis matrices $U_1,\ldots,U_l$ is to replace the definition (4.1) of the moments matrices $M_{1,i}$ for $i = 1,\ldots,n/(2k)$, by the new definition

$$M_{1,i} = \begin{pmatrix} p_{s_i+1} & 0 & \cdots & 0 \\ 0 & p_{s_i+2} & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & 0 & p_{s_i+2k} \end{pmatrix} \begin{pmatrix} 1 & x_{s_i+1} & \cdots & {x_{s_i+1}}^{2k-1} \\ 1 & x_{s_i+2} & \cdots & {x_{s_i+2}}^{2k-1} \\ \vdots & & & \vdots \\ 1 & x_{s_i+2k} & \cdots & {x_{s_i+2k}}^{2k-1} \end{pmatrix},$$

where $s_i = (i-1)2k$ and $p_j = p(x_j)^{1/2}$.

Now the integral equation (4.6) can be transformed to the equation

$$(I - D^{1/2}\mathcal{K}D^{1/2})(D^{-1/2}f) = (D^{-1/2}g),$$

which is discretized to a system that is sparse in the revised wavelet coordinates. The inverse matrix is also sparse.

4.3 Numerical Algorithms

In Section 4.1 we defined a class of bases for functions defined on $\{x_1, \ldots, x_n\}$ and in Section 4.2 we showed that, to finite precision, second-kind integral operators and their inverses are asymptotically sparse in these bases. In this section we present procedures for computation of the bases, discretized integral operators in these bases, and the inverses of these operators. In Section 4.4 we give some numerical examples based on our implementations of these procedures.

The computation of the wavelet bases is discussed next, followed by a discussion of the transformation of the integral operators to the wavelet bases. We defer discussion of the computation of the inverses, sketched above, to subsection 4.3.3, which contains detailed descriptions of all of the algorithms. Finally, subsection 4.3.4 gives the complexity analysis for the algorithms.

4.3.1 Computation of Wavelet Bases

It was mentioned in Section 4.1 that the mathematical definition of $U_1, \ldots, U_l$, if used directly, would result in a numerical procedure that would create large roundoff errors. A correct procedure is obtained by shifting and scaling the matrices $M_{j,i}$ defined there.

For a pair of numbers $(\mu, \sigma) \in \mathcal{R} \times (\mathcal{R} \backslash \{0\})$ we define a $2k \times 2k$-matrix $S(\mu, \sigma)$ whose (i,j)th element is the binomial term

$$S(\mu, \sigma)_{i,j} = \binom{j-1}{i-1} \frac{(-\mu)^{j-i}}{\sigma^{j-1}} \tag{4.7}$$

for $i \leq j$, and $S(\mu, \sigma)_{i,j} = 0$ otherwise. The matrix $S(\mu, \sigma)$ is upper-triangular and non-singular, and its inverse is given by the formula

$$S(\mu, \sigma)^{-1} = S(-\mu/\sigma, 1/\sigma). \tag{4.8}$$

Furthermore, the product formula

$$S(\mu_1, \sigma_1) S(\mu_2, \sigma_2) = S(\mu_1 + \mu_2 \sigma_1, \sigma_1 \sigma_2) \tag{4.9}$$

is easily verified.

We define $M'_{j,i}$ for $j = 1, \ldots, l$ and $i = 1, \ldots, n/(2^j k)$ by the formula

$$M'_{j,i} = M_{j,i} S(\mu_{j,i}, \sigma_{j,i}), \tag{4.10}$$

where $\mu_{j,i} = (x_{1+(i-1)k2^j} + x_{ik2^j})/2$, $\sigma_{j,i} = (x_{ik2^j} - x_{1+(i-1)k2^j})/2$ and the matrix $M_{j,i}$ is defined by Eq. (4.1) and Eq. (4.3) in Section 4.1. The matrix $U_{j,i}$ is given by the formula

$${U_{j,i}}^T = \mathrm{Orth}(M'_{j,i}), \tag{4.11}$$

which is equivalent to the definition given by Eq. (4.2). This equivalence immediately follows from the fact that $S(\mu, \sigma)$ is upper-triangular and non-singular.

The matrices $M'_{1,i}$ for $i = 1, \ldots, n/(2k)$ are actually computed by the formula

$$M'_{1,i} = \begin{pmatrix} 1 & \frac{x_{s_i+1}-\mu_{1,i}}{\sigma_{1,i}} & \cdots & \left(\frac{x_{s_i+1}-\mu_{1,i}}{\sigma_{1,i}}\right)^{2k-1} \\ 1 & \frac{x_{s_i+2}-\mu_{1,i}}{\sigma_{1,i}} & \cdots & \left(\frac{x_{s_i+2}-\mu_{1,i}}{\sigma_{1,i}}\right)^{2k-1} \\ \vdots & & & \vdots \\ 1 & \frac{x_{s_i+2k}-\mu_{1,i}}{\sigma_{1,i}} & \cdots & \left(\frac{x_{s_i+2k}-\mu_{1,i}}{\sigma_{1,i}}\right)^{2k-1} \end{pmatrix}, \tag{4.12}$$

where $s_i = (i-1)2k$. Likewise, the matrices $M'_{j,i}$ for $j = 2, \ldots, l$ and $i = 1, \ldots, n/(2^j k)$ are computed by the formula

$$M'_{j,i} = \begin{pmatrix} {U_{j-1,2i-1}}^U M'_{j-1,2i-1} S^1_{j,i} \\ {U_{j-1,2i}}^U M'_{j-1,2i} S^2_{j,i} \end{pmatrix}, \tag{4.13}$$

where $S^1_{j,i}$ and $S^2_{j,i}$ are defined by the formulae

$$S^1_{j,i} = S(\mu_{j-1,2i-1}, \sigma_{j-1,2i-1})^{-1} S(\mu_{j,i}, \sigma_{j,i}) \tag{4.14}$$

$$S^2_{j,i} = S(\mu_{j-1,2i}, \sigma_{j-1,2i})^{-1} S(\mu_{j,i}, \sigma_{j,i}). \tag{4.15}$$

Application of the inverse and product rules given in Eq. (4.8) and Eq. (4.9) to Eq. (4.14) and Eq. (4.15) yields formulae by which $S^1_{j,i}$ and $S^2_{j,i}$ can be computed:

$$S^1_{j,i} = S((\mu_{j,i} - \mu_{j-1,2i-1})/\sigma_{j-1,2i-1},\ \sigma_{j,i}/\sigma_{j-1,2i-1}) \tag{4.16}$$

$$S^2_{j,i} = S((\mu_{j,i} - \mu_{j-1,2i})/\sigma_{j-1,2i},\ \sigma_{j,i}/\sigma_{j-1,2i}). \tag{4.17}$$

The matrices $M'_{j,i}$ given by Eq. (4.12) and Eq. (4.13) are easily seen to be mathematically equivalent to those defined by Eq. (4.10); nonetheless, computation of $M'_{j,i}$ using Eqs. (4.12) and (4.13) avoids the large roundoff errors which would otherwise result.

4.3.2 Transformation to Wavelet Bases

We assume that for equispaced points $x_1, \ldots, x_n$ (defined in Eq. (4.4)) and some k the orthogonal matrices $U_1, \ldots, U_l$ defined in Section 4.1 have been computed ($l = \log_2(n/k)$). We now present a procedure for computation of UTU^T, where $U = U_l \cdots U_1$ and T is the discretized integral operator defined in Eq. (4.5).

Simple Example

We begin with a simplified example in which T is replaced by an $n \times n$-matrix V of rank k whose elements V_{ij} are defined by the equation

$$V_{ij} = \sum_{r=1}^{k} \sum_{s=1}^{k} \Lambda_{rs} {x_i}^{r-1} {x_j}^{s-1}, \qquad i, j = 1, \ldots, n.$$

Each row and each column of V contain elements which are the values of a polynomial of degree $k-1$. The matrix V can be written as $V = P^T \Lambda P$, where the elements of the $k \times n$-matrix P are defined by $P_{ij} = {x_j}^{i-1}$ and Λ is the $k \times k$-matrix with elements Λ_{ij}. Recalling that the last k rows of the basis matrix U consist of an orthogonalization of the moment vectors $\langle {x_1}^j, \ldots, {x_n}^j \rangle$ for $j = 0, \ldots, k-1$, we can rewrite V as $V = (P')^T \Lambda' P'$. Here the $k \times n$-matrix P' consists of the last k rows of U and Λ' is a new $k \times k$-matrix with elements Λ'_{ij}.

By the orthogonality of U, it is clear the $n \times n$-matrix $UVU^T = U(P')^T \Lambda' P' U^T$ consists entirely of zero elements except the $k \times k$-submatrix in the lower-right corner, which is the matrix Λ'. Given a function to compute elements of the $n \times n$-matrix V, the matrix Λ' can be computed in time independent of n by using a $k \times k$ extract of values from V. We form the matrix V' with elements V'_{ij} defined by the formula

$$V'_{ij} = V_{in/k, jn/k}, \qquad i, j = 1, \ldots, k. \tag{4.18}$$

Then $V' = (P'')^T \Lambda' P''$, where P'' is the $k \times k$ extract of P' with elements given by $P''_{ij} = P'_{i,jn/k}$. Thus we obtain

$$\Lambda' = ((P'')^T)^{-1} V' (P'')^{-1} \tag{4.19}$$

from P'' and V' readily in $O(k^3)$ operations, and we have obtained UVU^T.

General Case

The integral operator matrix T is, of course, not of low rank, but it can be divided into submatrices, each approximately of rank k (see Fig. 4.4). The submatrices near the main diagonal are of size $k \times k$, those next removed are $2k \times 2k$, and so forth up to the largest submatrices, of size $n/4 \times n/4$. The total number of submatrices is proportional to n/k. Given an error tolerance $\epsilon > 0$, k may be chosen (independently of n) so that each submatrix of T, say T^i, may be written as a sum, $T^i = V^i + E^i$, where the elements of V^i are given by a polynomial of degree $k-1$ and $\|E^i\| < \epsilon \|T^i\|$.

The simplified example, in which the matrix to be transformed is of rank k, is now applicable. Each submatrix of T is treated as a matrix of rank k and is transformed to wavelet coordinates (for its own scale) in order $O(k^3)$ operations. To make this precise, we write $T = T_0 + \cdots + T_{l-2}$ where T_i consists of the submatrices of size $2^i k \times 2^i k$. For each i, the submatrices of T_i may be interpolated by rank k submatrices, as indicated by the extract of Eq. (4.18), to obtain matrices V_i. Thus $T_i = V_i + E_i$, where $\|E_i\|$ is small. In the simplified

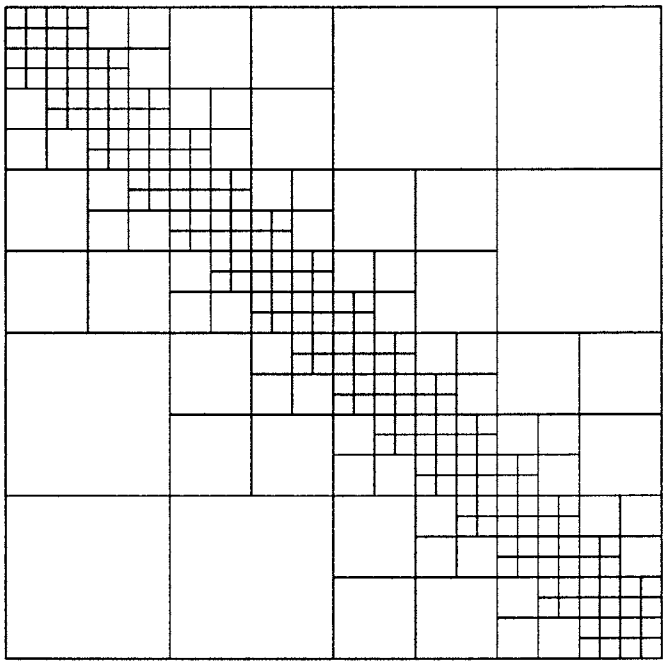

Figure 4.4: *The matrix represents a discretized integral operator with a kernel that is singular along the diagonal. The matrix is divided into submatrices of rank k (to high precision) and transformed to a sparse matrix with* $O(n \log n)$ *elements. Here* $n/k = 32$.

example above, we have shown that the transformed matrices

$$\begin{array}{lcc} W_0 & = & V_0 \\ W_1 & = & U_1 V_1 U_1{}^T \\ W_2 & = & U_2 U_1 V_2 U_1{}^T U_2{}^T \\ & & \vdots \\ W_{l-2} & = & U_{l-2} \cdots U_1 V_{l-2} U_1{}^T \cdots U_{l-2}{}^T \end{array} \tag{4.20}$$

can be computed by many applications of Eq. (4.19), all in order $O(nk^2)$ operations. This estimate follows from the fact that there are $O(n/k)$ submatrices, each of which is transformed in $O(k^3)$ operations. Now we define $n \times n$-matrices $R_0, \ldots, R_l$ recursively:

$$R_i = \begin{cases} W_0 & i = 0 \\ U_i R_{i-1} U_i{}^T + W_i & i \geq 1 \end{cases} \tag{4.21}$$

(here $W_{l-1} = W_l = 0$). Then R_l contains the final result, $R_l = U(T - E)U^T$, where $E = E_0 + \cdots + E_{l-2}$.

The matrix-matrix products in the definition of $R_0, \ldots, R_l$ can be computed directly, since the factors and the products contain no more than $O(n \log n)$ elements. A simple implementation with standard sparse matrix structures results in a total operation count of order $O(n \log^2 n)$, but an implementation using somewhat more elaborate data structures, in which repetitive handling of data is avoided, requires only order $O(n \log n)$ operations.

Computation using the result R_l is made more efficient by removing the elements of R_l which can be neglected, within the precision with which R_l approximates UTU^T. For a given precision ϵ, we discard a matrix E' by eliminating elements from R_l below a threshold τ. The threshold depends on the choice of norm; in our implementation, we use the row-sum norm

$$\|A\| = \max_i \sum_{j=1}^{n} |A_{ij}|,$$

for an $n \times n$-matrix A. The element threshold

$$\tau = \frac{\epsilon}{n}\|T\| \tag{4.22}$$

clearly results in a discarded matrix E' with $\|E'\| < \epsilon\|T\|$.

4.3.3 Detailed Descriptions of Algorithms

Procedure to compute $U_1, \ldots, U_l$

Comment [Input to this procedure consists of the number of points n, the number of zero moments k, and the points $x_1, \ldots, x_n$. Output is the matrices $U_{j,i}$ for $j = 1, \ldots, l$ and $i = 1, \ldots, n/(2^j k)$, which make up the matrices $U_1, \ldots, U_l$ (note $l = \log_2(n/k)$).]

Step 1.

Compute the shifted and scaled moments matrices $M'_{1,i}$ for $i = 1, \ldots, n/(2k)$ according to Eq. (4.12).

Step 2.

Compute $U_{1,i}$ from $M'_{1,i}$ by Eq. (4.11) using Gram-Schmidt orthogonalization for $i = 1, \ldots, n/(2k)$.

Step 3.

Comment [Compute $M'_{j,i}$ and $U_{j,i}$ for $j = 2, \ldots, l$ and $i = 1, \ldots, n/(2^j k)$.]
do $j = 2, \ldots, l$
  do $i = 1, \ldots, n/(2^j k)$

Compute $U_{j-1,2i-1}{}^{U}M'_{j-1,2i-1}$ and $U_{j-1,2i}{}^{U}M'_{j-1,2i}$.
Compute $S^1_{j,i}$ by Eq. (4.16) and $S^2_{j,i}$ by Eq. (4.17);
multiply to obtain $M'_{j,i}$ by Eq. (4.13).
Orthogonalize $M'_{j,i}$ to obtain $U_{j,i}$ by Eq. (4.11).
enddo
enddo

Procedure to compute UTU^T

Comment [Input to this procedure consists of n, k, the matrices $U_{j,i}$ computed above, a function to compute elements of T, and the chosen precision ϵ. Output is a matrix R_l such that $\|R_l - UTU^T\| < \epsilon\|T\|$.]

Step 4.

Compute the $k \times k$ extracts, indicated by Eq. (4.18), of the submatrices of T shown in Fig. 4.4.

Step 5.

Extract the matrices P'' (Eq. (4.19)) from U_1, U_2U_1, ..., $U_l \cdots U_1$ and compute $W_0, \ldots, W_{l-2}$ according to Eqs. (4.20).

Step 6.

Compute $R_0, \ldots, R_l$ by Eq. (4.21), discarding elements below a threshold τ determined by the precision ϵ (Eq. (4.22)).

Procedure to compute $UT^{-1}U^T$

Comment [Input to this procedure consists of n, the matrix R_l which approximates UTU^T, and the precision ϵ. Output is a matrix X_m that approximates $UT^{-1}U^T$.]

Step 7.

Compute the matrix $X_0 = R_l{}^T R_l / \|R_l{}^T R_l\|$ by direct matrix multiplication, discarding elements below a threshold τ determined by the precision ϵ (Eq. (4.22)).

Step 8.

Comment [Obtain the inverse by Schulz iteration.]
do $m = 0, 1, \ldots$ **while** $\|I - X_m R_l\| \geq \epsilon$
Compute $X_{m+1} = 2X_m - X_m R_l X_m$, discarding elements

below threshold.
enddo

4.3.4 Complexity Analysis

In the following table, we provide the operation count for each step of the computation of $UT^{-1}U^T$.

Step	Complexity	Explanation
1.	$O(nk)$	There are $n/(2k)$ $2k \times 2k$-matrices; each element of the matrices is computed in constant time.
2.	$O(nk^2)$	For each of the $n/(2k)$ matrices, perform a Gram-Schmidt orthogonalization requiring order $O(k^3)$ operations.
3.	$O(nk^2)$	For each of $n/(4k)+n/(8k)+\cdots+1 = n/(2k)-1$ matrices, compute four products of a $k \times 2k$-matrix with a $2k \times 2k$-matrix, construct two $2k \times 2k$-matrices, and orthogonalize one $2k \times 2k$-matrix.
4.	$O(nk)$	There are $6(1+3+7+\cdots+(n/(2k)-1))+3(n/k)-2$, or order $O(n/k)$, submatrices of T and for each matrix we compute k^2 elements.
5.	$O(nk^2)$	There are $n/(2k)+n/(4k)+\cdots+1 = n/k-1$ matrices P'', each the product of two $k \times k$-matrices. These are each inverted and multiplied with the $O(n/k)$ matrices of the previous step.
6.	$O(n \log n)$	The diagonally-banded matrix W_0, which contains $O(n)$ elements, grows to $O(n \log n)$ elements by the computation of UW_0U^T, as can be seen by simply examining pictures of W_0 and U. The nonzero elements of the transformed $W_1, \ldots, W_{l-2}$ are a subset of those of W_0.
7.	$O(n \log^2 n)$	Multiplication of two matrices, each with order $O(n \log n)$ elements, to obtain a product with order $O(n \log n)$ elements.

Step	Complexity	Explanation
8.	$O(n \log^2 n)$	Two multiplications like that of Step 7 are made per iteration; the number of iterations is independent of n and given by bound (2.23).
Total	$O(n \log^2 n)$	

4.4 Numerical Examples

In this section we present operators from several integral equations, the discretization and transformation of the operators to our wavelet bases, and the inversion of the operators via Schulz method.

4.4.1 Uncorrected Quadratures

We first examine simple quadratures with equal weights, except weight zero at the singularity, as represented by matrix $T = T(n)$ defined by Eq. (4.5). We transform the matrix $I - T$ to wavelet coordinates as described in Section 4.3.2, then compute $(I - T)^{-1}$.

These discretizations are not particularly useful for the solution of the integral equations, due to their slow convergence to the integral operators. They nonetheless make good illustrative examples, for they retain the smoothness of the operator kernels and produce correspondingly sparse matrices. In the next subsection, we examine the results of using high-order quadratures.

For various sizes n of discretization, we tabulate the average number of elements per row in the transformed matrix $U(I - T)U^T$ and the computation time to obtain the matrix. In addition, we display the average number of elements per row of its inverse, and the time to compute the inverse. Finally, we show the error introduced by these computations. The error is determined by the application of the forward and inverse transformations to a random vector: Choose a vector v of length n with uniformly distributed pseudo-random elements; compute $(I-T)v$ directly, by a standard procedure requiring order $O(n^2)$ operations; transform to wavelet coordinates, obtaining $U(I-T)v$; apply the computed value of $U(I - T)^{-1}U^T$ to the vector $U(I - T)v$; transform to original coordinates by application of U^T; compare the result to v. The measure of error is the relative $\mathcal{L}^2$ error $e_{\mathcal{L}^2}$, defined by Eq. (3.27).

The programs to transform and invert, as well as those to determine the error, were implemented in FORTRAN. All computations were performed in double-precision arithmetic on a Sun Sparcstation 1.

The first set of examples is for the kernel $K(x,t) = \log|x - t|$, for a wavelet basis of order $k = 4$ and various choices of precision ϵ. The matrix sparsities, execution times, and errors appear in Table 4.1. Although the sparse matrices

Table 4.1: *The integral operator $\mathcal{K}$ defined by the formula $(\mathcal{K}f)(x) = f(x) - \int_0^1 \log|x-t|\, f(t)\, dt$ is discretized, transformed to the wavelet coordinates with $k = 4$, and inverted. For various precisions ϵ and various sizes of discretization, we tabulate the average number of elements/row N_1 of the matrix in wavelet coordinates and the time in seconds t_1 to compute it, corresponding statistics N_2 and t_2 for the inverse, and the error (see text).*

		Transform.		Inversion		$\mathcal{L}^2$
ϵ	n	N_1	t_1	N_2	t_2	Error
10^{-2}	64	7.2	2	8.3	2	0.503E−02
	128	5.9	3	6.5	4	0.257E−02
	256	3.8	7	4.4	4	0.250E−02
	512	2.8	13	3.1	6	0.236E−02
	1024	1.9	26	2.1	6	0.227E−02
	2048	1.4	49	1.4	6	0.221E−02
	4096	1.2	97	1.2	8	0.221E−02
	8192	1.1	195	1.1	12	0.217E−02
10^{-3}	64	17.6	2	19.5	14	0.350E−03
	128	18.1	5	20.0	36	0.270E−03
	256	18.0	11	20.0	83	0.331E−03
	512	14.5	21	15.7	123	0.257E−03
	1024	13.3	41	15.5	262	0.340E−03
	2048	8.5	73	9.8	287	0.233E−03
	4096	5.8	131	6.5	304	0.222E−03
	8192	3.7	242	4.4	312	0.221E−03
10^{-4}	64	28.4	3	30.3	36	0.104E−03
	128	32.1	6	34.3	111	0.140E−03
	256	34.5	15	37.5	302	0.161E−03
	512	33.1	31	35.8	618	0.177E−03
	1024	30.2	63	33.6	1280	0.189E−03
	2048	25.0	121	27.6	2040	0.192E−03

are not banded, we loosely refer to the average number of matrix elements per row as the matrix *bandwidth*. We make the following observations:

1. The bandwidths N_1, N_2 of the operator and its inverse *decrease* with increasing matrix size. In other words, in the range of matrix sizes tabulated, the number of matrix elements grows *sublinearly* in the matrix dimension n.

2. The operator matrix in wavelet coordinates is computed in time that grows nearly linearly in n.

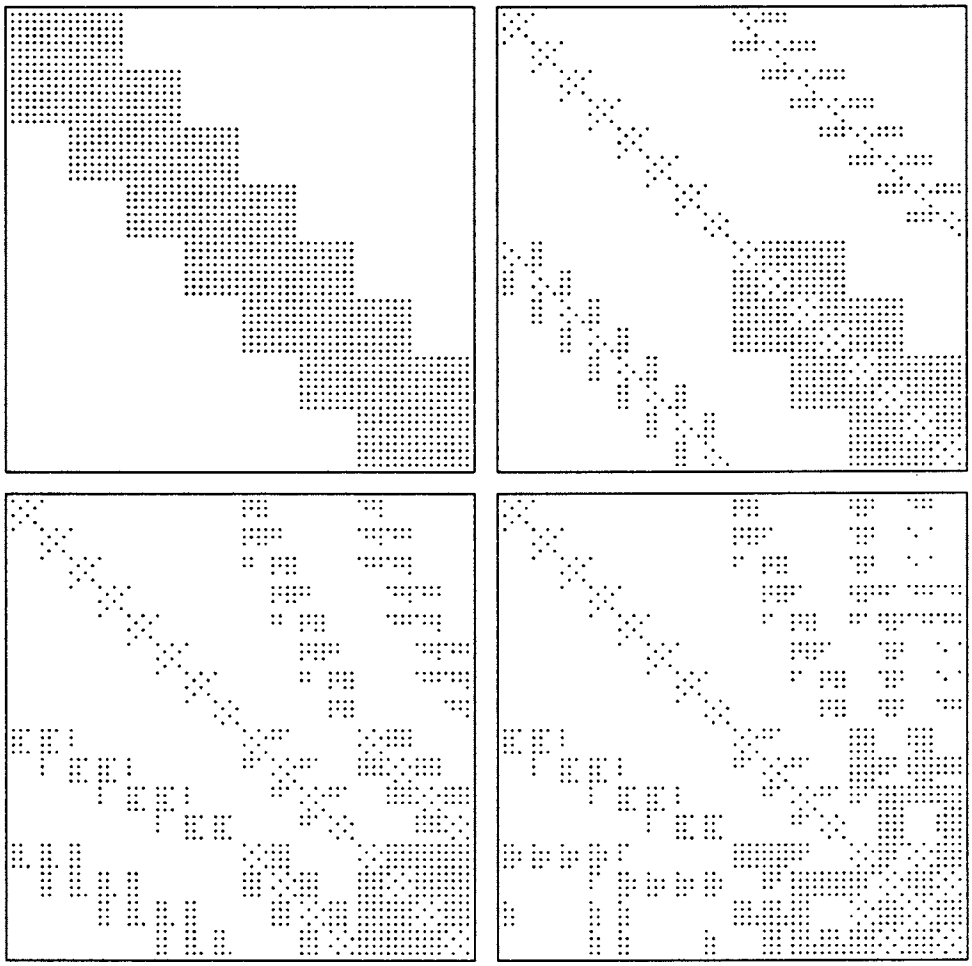

Figure 4.5: *The matrices constructed in the transformation of* $I - T$*, matrices* $R_0, \ldots, R_3$ *defined in Eq. (4.21), are shown for kernel* $K(x,t) = \log|x-t|$*,* $\epsilon = 10^{-3}$*, and* $n = 64$*. The matrix* R_4 *looks like* R_3 *and is not shown.*

3. The inverse matrix is computed in time which grows sublinearly in n. This is due to the fact that the cost of multiplying the sparse matrices is roughly order $O(nN^2)$, for size n and bandwidth N. One result is that the cost sometimes drops as n increases.

4. The accuracy is within the precision specified. In fact, due to the conservative element thresholding (Eq. 4.22), the actual error is considerably less than ϵ.

5. The cost increases with increasing precision ϵ, due to the increasing bandwidths generated. The the bandwidths increase approximately as $\log(1/\epsilon)$.

6. For $k = 4$, our fast transformation algorithm does not maintain the specified

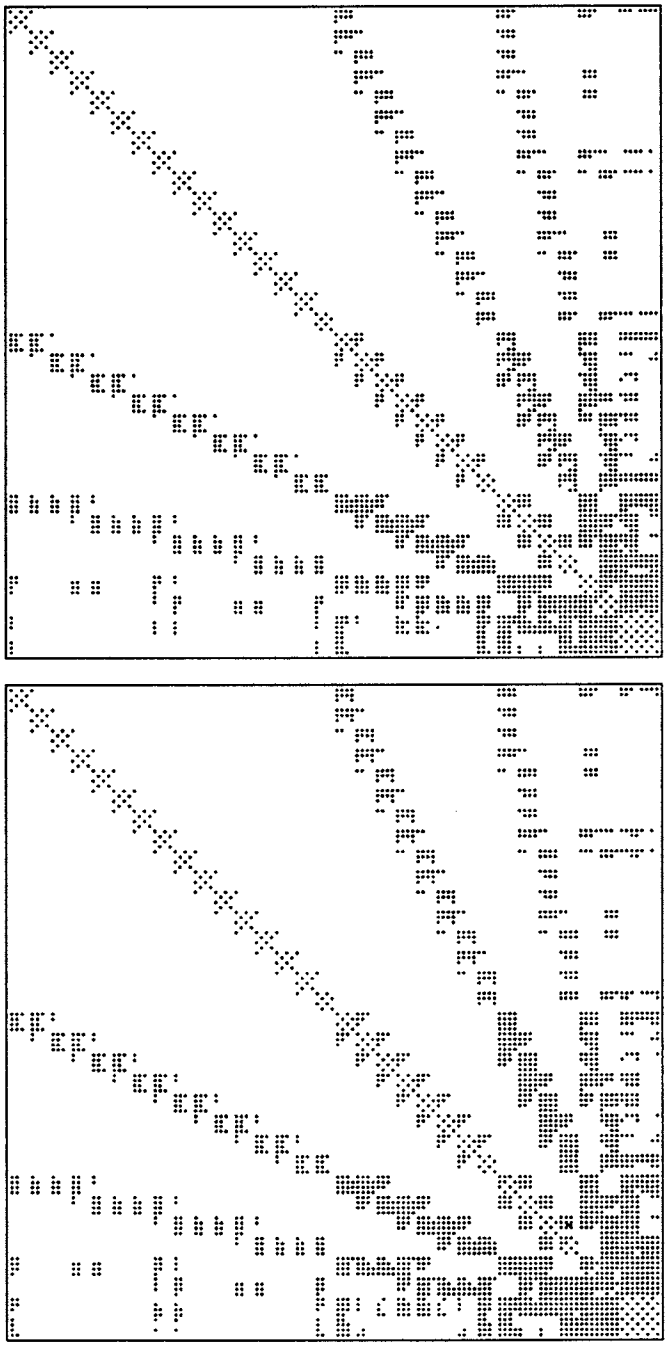

Figure 4.6: *Transformed matrix* $U(I-T)U^T$ *(top) and its inverse (bottom) are shown for kernel* $K(x,t) = \log|x-t|$, $\epsilon = 10^{-3}$, *and* $n = 128$.

precision of $\epsilon = 10^{-4}$. This anticipated result follows from the error estimate for polynomial interpolation of logarithm on intervals separated from the

origin. An unanticipated attendant result is that the bandwidth increases as the quality of approximation deteriorates (compare to $k = 8$, below). As a result, we did not complete examples for $n = 4096$, 8192.

7. The inversion of the 8192×8192 matrix preserving 3-digit accuracy is done in 5 minutes on the Sparcstation. This compares to 95 days (estimated) for inverting the dense matrix by Gauss-Jordan and to 24 minutes for one dense matrix-vector multiplication of that size.

The condition number of the problem, as approximated by the product of the row-sum norms of $U(I - T)U^T$ and its computed inverse, is 3 (independent of size). Five iterations were required by the Schulz method to achieve convergence.

In Fig. 4.5 we show stages in the transformation of the matrix $I - T$. In particular, for $\epsilon = 10^{-3}$ and $n = 64$, the matrices $R_0, \ldots, R_{l-1}$ defined in Eq. (4.21) are shown. In addition, for $n = 128$ the transformed matrix $U(I - T)U^T$ and its inverse are shown in Fig. 4.6.

In the next set of examples, for which results are displayed in Table 4.2, we used the wavelet basis of order $k = 8$. We observe:

1. The bandwidths of the operator matrix and its inverse are less for $k = 8$ than for $k = 4$. The inversion times are correspondingly smaller.

2. The time required to compute the operator matrix is almost four times as large as that for $k = 4$. This is due to the cost of transforming the near-diagonal band, which is twice as wide for $k = 8$ as for $k = 4$.

3. The obtained accuracy exceeds the specified precision consistently.

4. As for $k = 4$, the scaling with size n is linear for the transformation step and sublinear for the inversion step.

In the final set of examples in which uncorrected quadratures were used, we perform computations for $k = 4$ and $\epsilon = 10^{-3}$, with various operator kernels. Table 4.3 presents the results. The first three kernels contain singularities of the types $s(x) = \log(x)$ and $s(x) = x^\alpha$ for $\alpha = \pm\frac{1}{2}$, and are nonsymmetric and non-convolutional. It is readily seen that the bandwidth is strongly dependent on the type of singularity, with the singularity $x^{-1/2}$ producing the greatest bandwidth. We mention also that this particular integral equation is poorly-conditioned; the condition numbers of the discretizations for $n = 64, 128, 256, 512, 1024$ are $9, 17, 34, 98, 469$, respectively.

The fourth kernel provides an example with an oscillatory coefficient $p(x) = (1 + \frac{1}{2}\sin(100x))$. The bases developed in Section 4.2.3, which depend on p, are used to transform the discretized integral operator to sparse form. We see in Table 4.3 that the inverse is also very sparse.

Table 4.2: *The integral operator $\mathcal{K}$ defined by the formula $(\mathcal{K}f)(x) = f(x) - \int_0^1 \log|x-t|\, f(t)\, dt$ is discretized, transformed to the wavelet coordinates with $k = 8$, and inverted. (See Table 4.1 and text.)*

		Transform.		Inversion		$\mathcal{L}^2$
ϵ	n	N_1	t_1	N_2	t_2	Error
10^{-2}	64	5.8	4	6.2	1	0.191E−02
	128	5.0	10	5.5	2	0.368E−02
	256	3.3	22	3.6	3	0.184E−02
	512	2.7	46	2.9	4	0.113E−02
	1024	1.8	92	1.8	4	0.177E−02
	2048	1.4	182	1.4	5	0.170E−02
	4096	1.2	363	1.2	8	0.928E−03
	8192	1.1	729	1.1	11	0.166E−02
10^{-3}	64	13.4	5	14.5	8	0.373E−03
	128	14.2	13	15.5	21	0.332E−03
	256	13.5	28	14.5	46	0.259E−03
	512	12.7	57	13.6	90	0.225E−03
	1024	10.2	114	11.1	134	0.198E−03
	2048	7.7	221	8.3	176	0.179E−03
	4096	4.9	429	5.2	185	0.174E−03
	8192	3.5	818	3.7	208	0.173E−03
10^{-4}	64	21.8	6	23.0	23	0.280E−04
	128	26.3	15	28.0	81	0.253E−04
	256	28.7	35	31.0	235	0.246E−04
	512	28.4	75	30.9	538	0.184E−04
	1024	25.5	149	27.2	969	0.925E−05
	2048	22.0	297	23.8	1739	0.899E−05
	4096	17.7	561	19.1	2610	0.798E−05

4.4.2 Solution of Integral Equations

In the preceding subsection, we examined the characteristics of various integral operators and their inverses in wavelet coordinates. We used completely straightforward discretizations; the quadratures represented sums of the integrands at equispaced points (excluding singular points). Such simple quadratures converge too slowly to the integral operators to be of much use in solving integral equations, and we now turn to the high-order quadratures developed in Chapter 3.

We first present examples which correspond to the various kernels already tested and shown in Table 4.3. In Table 4.4 we tabulate the results, and differ-

Table 4.3: *The integral operator $\mathcal{K}$ defined by the formula $(\mathcal{K}f)(x) = f(x) - \int_0^1 K(x,t)\,f(t)\,dt$, for nonsymmetric, nonconvolutional kernels $K(x,t)$ shown below, is discretized, transformed to the wavelet coordinates with $k = 4$ and $\epsilon = 10^{-3}$, and inverted. (See Table 4.1 and text.)*

		Transform.		Inversion		$\mathcal{L}^2$
$K(x,t)$	n	N_1	t_1	N_2	t_2	Error
$\cos(xt^2)\log\|x-t\|$	64	18.2	2	20.2	15	0.318E−03
	128	18.6	5	20.4	37	0.302E−03
	256	17.9	11	19.8	82	0.301E−03
	512	14.9	22	16.3	131	0.284E−03
	1024	12.9	42	14.7	242	0.315E−03
	2048	8.5	76	9.5	283	0.241E−03
	4096	5.5	137	6.1	291	0.231E−03
	8192	3.6	252	4.3	310	0.230E−03
$\cos(xt^2)\|x-t\|^{-1/2}$	64	27.2	3	28.9	32	0.256E−03
	128	31.6	7	34.1	122	0.357E−03
	256	35.6	16	40.6	454	0.434E−03
	512	37.3	35	46.3	1509	0.643E−03
	1024	34.5	72	45.4	4166	0.821E−03
$\cos(xt^2)\|x-t\|^{1/2}$	64	6.8	2	7.3	2	0.303E−03
	128	4.4	4	4.7	2	0.204E−03
	256	2.9	8	3.0	3	0.209E−03
	512	2.1	15	2.3	3	0.165E−03
	1024	1.5	30	1.5	3	0.208E−03
	2048	1.4	60	1.4	6	0.909E−03
	4096	1.1	119	1.2	7	0.614E−03
	8192	1.1	242	1.1	12	0.666E−03
$(1+\frac{1}{2}\sin(100x))\times$	64	30.5	3	33.8	44	0.344E−03
$\log\|x-t\|$	128	31.8	6	35.1	103	0.363E−03
	256	21.2	12	24.1	119	0.348E−03
	512	18.6	23	20.7	225	0.372E−03
	1024	15.8	45	18.4	404	0.392E−03
	2048	10.6	82	12.2	466	0.355E−03
	4096	6.4	145	7.4	497	0.336E−03
	8192	4.0	265	4.6	510	0.331E−03

ences from Table 4.3 reflect the effect of the quadratures.

For the remaining examples we choose integral equations that can be solved

Table 4.4: *The integral operator $\mathcal{K}$ defined by the formula $(\mathcal{K}f)(x) = f(x) - \int_0^1 K(x,t)\,f(t)\,dt$, for nonsymmetric, nonconvolutional kernels $K(x,t)$ shown below, is discretized with the corrected trapezoidal rules, transformed to the wavelet coordinates with $k = 4$ and $\epsilon = 10^{-3}$, and inverted. (Compare to Table 4.3.)*

		Transform.		Inversion		$\mathcal{L}^2$
$K(x,t)$	n	N_1	t_1	N_2	t_2	Error
$\cos(xt^2)\log\lvert x-t\rvert$	64	28.3	4	31.6	38	0.164E−03
	128	31.5	9	34.3	103	0.162E−03
	256	30.8	21	33.9	221	0.172E−03
	512	27.0	41	29.7	370	0.177E−03
	1024	21.0	80	23.7	454	0.357E−03
	2048	14.8	143	17.2	566	0.317E−03
	4096	9.5	250	10.4	555	0.282E−03
	8192	5.8	448	6.9	665	0.271E−03
$\cos(xt^2)\lvert x-t\rvert^{-1/2}$	64	32.4	4	39.8	87	0.133E−02
	128	38.3	10	45.7	251	0.412E−03
	256	42.7	23	49.3	638	0.464E−03
	512	45.1	51	51.3	1494	0.562E−03
	1024	46.2	110	52.1	3309	0.635E−03
$\cos(xt^2)\lvert x-t\rvert^{1/2}$	64	10.4	3	18.4	9	0.867E−03
	128	7.6	6	13.8	13	0.526E−03
	256	5.1	13	9.3	16	0.358E−03
	512	3.3	25	5.2	15	0.292E−03
	1024	2.3	48	3.1	15	0.201E−03
	2048	1.9	96	2.3	20	0.393E−03
	4096	1.5	188	1.7	25	0.405E−03
	8192	1.3	374	1.4	36	0.404E−03

analytically, so that the accuracy of the method can be checked. We consider a class of integral equations with logarithmic kernel,

$$f(x) - p(x)\int_0^1 \log|x-t|\,f(t)\,dt = g_m(x), \qquad x \in [0,1], \tag{4.23}$$

where the right hand side g_m is chosen so that the solution f is given by the formula $f(x) = \sin(mx)$. The integration can be performed explicitly, yielding the formula

Table 4.5: *The integral equations* $f(x) - \int_0^1 \log|x-t|\, f(t)\, dt = g_m(x)$, *for which an explicit solution is known, are solved by the methods of this chapter (compare to Table 4.1 and see text). For* $\epsilon = 10^{-2}, 10^{-3}, 10^{-4}$ *we set* $k = 4, 4, 8$, *respectively.*

		Transform.		Inversion		$\mathcal{L}^2$
ϵ	n, m	N_1	t_1	N_2	t_2	Error
10^{-2}	64	11.4	3	14.4	7	0.283E−02
	128	10.7	7	13.2	14	0.212E−02
	256	8.6	13	10.6	20	0.140E−02
	512	6.3	26	7.6	26	0.112E−02
	1024	3.6	48	4.5	28	0.821E−03
	2048	1.9	90	2.3	21	0.932E−03
	4096	1.3	174	1.5	15	0.674E−03
	8192	1.1	344	1.1	13	0.499E−03
10^{-3}	64	27.7	4	31.3	36	0.235E−03
	128	31.0	9	34.2	99	0.169E−03
	256	30.6	20	33.6	215	0.161E−03
	512	27.5	41	30.2	377	0.130E−03
	1024	21.7	79	24.4	470	0.597E−03
	2048	15.5	143	18.1	604	0.479E−03
	4096	9.7	248	10.6	579	0.415E−03
	8192	6.0	444	7.3	690	0.354E−03
10^{-4}	64	37.2	8	45.9	78	0.127E−03
	128	47.1	23	56.5	278	0.473E−04
	256	52.9	54	60.9	745	0.311E−04
	512	55.0	118	61.4	1701	0.100E−04
	1024	52.3	248	57.2	3287	0.734E−05

$$\begin{aligned}\int_0^1 \log|x-t|\, \sin(mt)\, dt \;=\; & \log(x) - \cos(m)\log(1-x) \\ & - \cos(mx)[\mathrm{Ci}(mx) - \mathrm{Ci}(m(1-x))] \\ & - \sin(mx)[\mathrm{Si}(mx) + \mathrm{Si}(m(1-x))],\end{aligned}$$

where Ci and Si are the cosine integral and sine integral (see, *e.g.*, [1], p. 231). Equation (4.23) clearly requires quadratures with increasing resolution as m increases; for our examples we let $n = m$, which corresponds to 2π points per oscillation of the right hand side g_m.

Initially we choose coefficient $p(x) = 1$. The results are given in Table 4.6. Here the error shown is the error of the computed solution relative to the true

solution of the integral equation. Many of the observations of the preceding examples can be repeated here; additionally, we make the following comments:

1. The bandwidths are greater than for the uncorrected quadratures, but this effect generally decreases with increasing size.

2. The integral equations are solved to within the specified precision in every case but one. The exception, for $\epsilon = 10^{-4}$ and $n = 64$, is likely due to the small number of quadrature points and high specified precision.

3. An integral equation requiring an 8192-point discretization is solved to 3-digit accuracy in less than 20 minutes on the Sparcstation.

For our second set of integral equations, we let the coefficient p be the oscillatory function given by the formula $p(x) = 1 + \frac{1}{2}\sin(100x)$. We carry out the transformation described in Section 4.2.3 to solve the integral equation 4.23. The results are shown in Table 4.6 and as with Table 4.5 the error refers to the error of the computed solution relative to the true solution of the integral equation. For the oscillatory coefficient we see performance similar to the constant-coefficient problem, but the cost is higher.

Table 4.6: *The integral equations* $f(x) - p(x)\int_0^1 \log|x-t|\, f(t)\,dt = g_m(x)$, *for which an explicit solution is known, are solved by the methods of this chapter (compare to Table 4.1 and see text). For* $\epsilon = 10^{-2}, 10^{-3}, 10^{-4}$ *we set* $k = 4, 4, 8$, *respectively.*

		Transform.		Inversion		$\mathcal{L}^2$
ϵ	n, m	N_1	t_1	N_2	t_2	Error
10^{-2}	64	19.7	4	23.9	18	0.360E−02
	128	17.7	8	21.0	36	0.182E−02
	256	12.6	15	14.6	47	0.174E−02
	512	8.4	29	9.8	57	0.112E−02
	1024	4.7	55	5.7	56	0.104E−02
	2048	2.4	103	2.7	45	0.902E−03
	4096	1.6	198	1.7	38	0.720E−03
	8192	1.3	392	1.3	35	0.543E−03
10^{-3}	64	36.2	4	41.3	63	0.228E−02
	128	40.8	10	47.0	186	0.209E−03
	256	40.5	23	47.3	427	0.177E−03
	512	34.7	46	40.9	712	0.125E−03
	1024	26.6	87	32.5	1042	0.134E−03
	2048	18.7	158	22.5	1065	0.597E−03
	4096	12.2	281	14.2	1127	0.529E−03
	8192	7.2	502	8.4	1104	0.461E−03
10^{-4}	64	47.6	9	58.2	123	0.230E−02
	128	60.7	25	77.3	479	0.180E−03
	256	64.1	59	81.2	1204	0.124E−03
	512	62.5	128	76.3	2492	0.125E−04
	1024	58.8	267	69.3	4672	0.862E−05

Chapter 5

Generalizations and Applications

A new class of bases for $\mathcal{L}^2[0,1]$ has been constructed in which a variety of integral operators are represented as sparse matrices. The inverses of these matrices are also sparse, a fact which enables the corresponding integral equations to be solved rapidly.

Vector-space analogues of the bases were also developed, and the latter bases appear preferable, due to their flexibility, for the solution of a variety of problems.

To pose integral equations as finite-dimensional numerical problems, the Nyström, or quadrature, method was used. Quadrature rules based on the trapezoidal rule, corrected to yield high-order convergence, were developed for several types of singularities. The combination of these quadratures and the vector-space wavelet bases yields a fast algorithm for the solution of second-kind integral equations. We have proven a time complexity of order $O(n \log^2 n)$, but observed order $O(n)$ performance in practice, where n is the number of points in the discretization. This cost should be contrasted with a cost of order $O(n^2)$ for direct application of a dense matrix, and order $O(n^3)$ for direct inversion.

A number of limitations exist in the procedures described above. These restrictions may be categorized as "software limitations" and "research questions". We discuss software limitations first.

Software Limitations

In both the function-space and vector-space settings, we have assumed that the size of the problem n has the form $n = 2^l k$ for some l. This restriction is not fundamental; it merely simplifies the software. For the multi-wavelet bases, one could expand "deeper" in some parts of the interval of definition than others, to efficiently resolve the the right-hand-side. A similar effect can be obtained in the discrete case from non-uniform spacing of the discretization points.

A second software restriction is the assumption of only diagonal singularities. This case is an important one in practice, but in certain situations one may encounter singularities or near-singularities off the main diagonal. The scheme described in Section 4.3.2 for transformation of a matrix to wavelet bases can be readily revised to an adaptive scheme, which works as follows: an $m \times m$ submatrix A is transformed to wavelet coordinates, under the assumption that it can be approximated to high precision in each direction by a polynomial of degree less than k. This assumption is then checked by dividing A into four submatrices, each of dimension $m/2 \times m/2$, transforming each submatrix, and "glueing" the pieces together. If the results from the two computations match (to high precision), no further refinement of the original submatrix is needed. Otherwise, the procedure is repeated recursively on the $m/2 \times m/2$ submatrices. The cost of this adaptive procedure is roughly 5 times as great as the cost of a static procedure in which the structure of the singularities is known *a priori*.

Research Questions

The list of research issues is of course much longer. One of the most pressing issues is the generalization to two and three dimensions. Although the generalization of the multi-wavelet bases to several dimensions is described in Chapter 2, and an analogous construction creates vector-space bases for several dimensions, quadratures corresponding to those of Chapter 3 have not been developed for two and three-dimensional problems.

There exist integral equations, as observed in the numerical examples of the previous chapter, in which the inverse operator's wavelet representation is less sparse than that of the original integral operator. In such cases it would be advantageous to avoid inversion of the operator and instead obtain a sparse LU-factorization. This was attempted with the bases developed in this thesis, but for the operators we examined the LU factors have substantial fill-in. We believe, however, that a somewhat different class of bases can be constructed with the property that the LU factors preserve the sparsity of the operator matrix.

Another question is whether similar "custom-constructed" bases can be used to create sparse representations of integral operators with oscillatory kernels. Initial efforts in this direction for a limited class of such operators, in particular for Fourier transforms with non-equispaced points and frequencies, appear promising [7].

Applications

In this thesis the primary application of our new wavelet bases has been the solution of second-kind integral equations. The bases are very effective for the

fast solution of a wide class of such problems. In addition, many other classes of problems can be solved efficiently using these techniques. We list a few of these problem types.

1. Elliptic partial differential equations rewritten as integral equations by the Lippman-Schwinger method, in which the Green's functions are non-oscillatory.

2. Evolution of homogeneous parabolic PDEs with constant or periodic boundary conditions, by explicit time steps. This method consists of repeated squarings of the operator for a single time step, leading to an order $O(n \log t)$ algorithm for evolving an n-point discretization for t time steps.

3. Evolution of general parabolic PDEs by implicit time steps, in which the elliptic problem on each time step is solved in wavelet coordinates.

4. Evolution of hyperbolic PDEs by a method of operator squaring analogous to the scheme proposed for homogeneous parabolic PDEs above.

5. Problems of potential theory and pseudo-differential operators.

6. Signal compression, including signals of seismic, visual, and vocal origin. There is also reason to expect that analysis of such compressed data will be simpler than analysis of data resulting from less efficient compression schemes.

In this thesis we strayed from the original mathematical definition of wavelets to construct classes of bases tailored for numerical computation. The basis functions' (or vectors') principal properties of local support and vanishing moments lead to sparse representations of functions and operators that are smooth except at a finite number of singularities. There is little doubt that other bases can be constructed along similar lines to possess various properties. One current challenge is the construction of bases suitable for the efficient representation of a variety of oscillatory operators.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions.* National Bureau of Standards, Washington, DC, 1972.

[2] B. Alpert and V. Rokhlin. A fast algorithm for the evaluation of Legendre expansions. Technical report, Department of Computer Science, Yale University, 1989.

[3] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. Technical report, Department of Computer Science, Yale University, 1989.

[4] G. Dahlquist and Å. Björck. *Numerical Methods.* Prentice Hall, Englewood Cliffs, NJ, 1974.

[5] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, XLI:909–996, 1988.

[6] L. M. Delves and J. L. Mohamed. *Computational Methods for Integral Equations.* Cambridge University Press, 1985.

[7] A. Dutt and V. Rokhlin. Personal communication.

[8] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[9] L. Greengard and J. Strain. The fast Gauss transform. Technical report, Department of Computer Science, Yale University, 1989.

[10] A. Grossman and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15:723–736, 1984.

[11] S. Karlin and W. Studden. *Tchebycheff Systems with Applications in Analysis and Statistics.* Wiley and Sons, 1966.

[12] S. Mallat. Multiresolution approximation and wavelets. Technical report, GRASP Lab., Department of Computer and Information Science, University of Pennsylvania.

[13] Y. Meyer. Principe d'incertitude, bases Hilbertiennes et algèbres d'opérateurs. Technical report, Séminaire Bourbaki, 1985-1986, nr. 662.

[14] Y. Meyer. Ondelettes et functions splines. Technical report, Séminaire EDP, Ecole Polytechnique, Paris, France, 1986.

[15] S. O'Donnell and V. Rokhlin. A fast algorithm for the numerical evaluation of conformal mappings. *SIAM Journal for Scientific and Statistical Computing*, 10:475–487, 1989.

[16] V. Rokhlin. End-point corrected trapezoidal quadrature rules for singular functions. *Computers and Mathematics with Applications*, to appear, 1990.

[17] G. Schulz. Iterative berechnung der reziproken matrix. *Zeitschrift für Angewandte Mathematik und Mechanik*, 13:57–59, 1933.

[18] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer Verlag, 1980.