Multilevel Compression of Linear Operators: Descendants of Fast Multipole Methods and Calderón-Zygmund Theory

Per-Gunnar Martinsson and Mark Tygert

November 13, 2013

Preface

The authors would like to thank V. Rokhlin both for discovering much of the content of later parts of these notes and for elaborating the material for us while we prepared the notes. We would also like to thank R. R. Coifman and L. Greengard for many discussions related to the content of these notes, and S. Zucker for proposing that we develop the course which stimulated us to produce these notes.

Contents

Chapter 1

Introduction

This document provides largely self-contained lecture notes for the first half of an applied math course, introducing the numerical treatment of differential equations such as the Laplace, Helmholtz, wave, and Maxwell equations via fast multipole methods and related techniques of matrix compression. In particular, the notes discuss a method that the IEEE's and American Institute of Physics' *Computing in Science and Engineering* journal deemed to be one of the top ten algorithms of the twentieth century (see Dongarra and Sullivan [15]). The minimal prerequisites are linear algebra, Calculus, complex-analytic contour-integration, and some familiarity with convolution, the Fourier transform, and sampling. Naturally, some familiarity with the material cited in the recommended reference list (see Chapter 9) could enable deeper understanding of the content of these lecture notes. See Nishimura [48] for the history of the development of these methods within the computational community, and Daubechies [14] for (among many other topics) the history of the more-purely mathematical strains of these methods (Littlewood-Paley, Calderón-Zygmund, and microlocal analysis).

1.1 Rationale for modeling

These notes describe a class of algorithms for modeling systems governed by the fundamental equations of physical fields (the Maxwell equations, equations of fluid dynamics, Schrödinger and Dirac equations, equations of elasticity theory, etc.), with particular emphasis on the Helmholtz equation and its various special cases (the Laplace and Poisson equations, the Yukawa/screened-Coulomb equation, etc.). The algorithms have many other important applications, to topics ranging from computing contour integrals to computing singular value decompositions to analyzing and synthesizing linear combinations of special functions.

The benefits of modeling physical phenomena computationally are manifold. Computer-assisted modeling can reduce or even eliminate the need for prototyping when engineering new products, and enables quantitative exploration of physical theories. The value of the methods described in the present notes is well known in a wide variety of scientific and engineering disciplines; in electrical engineering alone the methods are used to design circuits free of undesired crosstalk, antennas with desired reception and broadcast patterns, stealthy aircraft nearly invisible to radar, etc. (see, for example, Cheng et al. [8], Chew et al. [11], and `http://www.integrandsoftware.com`). Also, new applications are appearing constantly.

1.2 Outline of the notes

We take the following approach to solving linear partial differential equations of interest:

1. Convert the linear partial differential equation (PDE) to a mathematically equivalent linear Fredholm integral equation of the second kind (see Chapter 2).

2. Discretize the integral equation via the Nyström method (see Chapter 3).

3. Use an iterative solver (GMRES) based on Krylov subspaces (see Chapter 4) to solve the discretized integral equation, performing the required applications to vectors of the matrix associated with the discretized integral equation via the following means:

4. Accelerate the application of the relevant matrix to vectors via a fast multipole method (FMM) (see Chapter 8), observing that blocks of the matrix that are well-separated from the diagonal can be approximated to high precision via low-rank matrices, as follows:

5. Construct the low-rank approximations via the interpolative decomposition (ID) (see Chapter 7), using randomized algorithms to check the accuracy of the approximations (see Chapter 5), while using the singular value decomposition (SVD) as a theoretical tool (see Chapter 6).

This paradigm is very effective for solving the Laplace and Poisson equations, for solving the Helmholtz and time-harmonic Maxwell equations for objects that are at most a few wavelengths in size, as well as for solving many other equations arising in the sciences and engineering, including linearized Navier-Stokes equations. Variations on this paradigm can handle the Maxwell equations (both time-harmonic and time-dependent) for arbitrarily large objects (see Cheng et al. [8] and Chew et al. [11]). For an overview, see Nishimura [48].

Remark 1 The above five-point list does not reference the remaining chapters in the logical order; however, the presentation in the present notes does follow the logical order — later chapters presume that the reader is already familiar with material from earlier chapters, but not vice versa. The reader may wish to review periodically the brief list above while studying the remaining chapters.

1.3 Rationale for using integral equations

Three basic criteria determine the usefulness of a numerical method: computational cost (both running-time and memory usage), robustness (guaranteeing accurate, trustworthy, meaningful results regardless of the particular structure of the input data), and ease-of-use (particularly when applied in the complicated circumstances often encountered in physical reality and engineering practice). Substantial improvements in any one of the three criteria can enable science fictions to become matter-of-fact technologies. Furthermore, ease-of-use impacts the amount of human time entailed in using an algorithm. Converting differential equations to integral equations (as in point 1 of the above five-point list) and then applying the other methods described in the present notes yields cost-effective, robust, easy-to-use algorithms:

1.3.1 Computational cost

When modeling homogeneous media, integral-equation formulations can focus on functions whose domains are restricted to the boundaries of the media. Therefore, when we use integral-equation formulations, we often need only track functions on the boundaries of the media, rather than throughout the volume of the media. This reduced dimensionality can allow integral-equation formulations to cost far less than the equivalent differential-equation formulations.

When modeling scattering from inhomogeneous media, integral-equation formulations can focus on functions whose domains are restricted to where the scattering potential is nonzero. In contrast, differential-equation formulations often require tracking functions throughout a volume large enough to enclose either a perfectly matched layer or a huge rectangular or spherical container with nonreflecting walls (see, for example, Chew et al. [11]).

The algorithms described in the present notes have costs that are directly proportional to the minimal amount of input data necessary for determining the output data.

1.3.2 Robustness

The condition number of a numerical method governs the accuracy attained by the method (see, for example, Dahlquist and Björck [13] for a discussion of numerical conditioning). Any algorithm based on a direct (not preconditioned) discretization of a differential equation must have a high condition-number, sometimes prohibitively high. In contrast, algorithms based on proper integral-equation formulations often have low condition-numbers and hence produce more accurate solutions.

Furthermore, integral-equation formulations can transparently incorporate all boundary conditions exactly. In contrast, differential-equation formulations often require some sort of approximate scheme for handling boundary conditions. For example, differential-equation formulations of scattering problems require the use of either what is known as a perfectly matched layer, or a rectangular or spherical container with approximately nonreflecting walls (see, for example, Chew et al. [11]).

1.3.3 Ease-of-use

Integral-equation formulations often handle domains with complicated geometries — such as arbitrary computational meshes from computer-aided design (CAD) packages — completely transparently, whereas differential-equation formulations may require herculean efforts. It is generally much easier to attain reliable accuracy using integral-equation formulations.

Chapter 2

Integral equations

In this chapter, we reformulate various differential equations as integral equations, to facilitate their solution by means of the methods described later. Many — perhaps even most — of the partial differential equations encountered in physics and in mathematics reduce to equations of the type considered here, namely the Helmholtz equation and its variants.

2.1 Volume integral equation formulation of the Helmholtz equation

Given a complex number k, a positive integer d, and a function V on $\mathbb{R}^d$, we would like to solve the following pair of equations for ψ_{scat}:

$$(\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})\,\psi_{\text{tot}} = V\psi_{\text{tot}} \tag{2.1}$$

and

$$\psi_{\text{tot}} = \psi_{\text{in}} + \psi_{\text{scat}}, \tag{2.2}$$

where ψ_{tot}, ψ_{in}, and ψ_{scat} are functions on $\mathbb{R}^d$, $\boldsymbol{\nabla}^2$ is the Laplacian (the sum of the second-order partial derivatives with respect to the Cartesian coordinate axes), and $\mathbf{1}$ is the identity operator. In (2.1) and (2.2), we assume that ψ_{in} satisfies an analogue of (2.1) in which V is identically zero, namely,

$$(\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})\,\psi_{\text{in}} = 0. \tag{2.3}$$

Formula (2.1) is known as the Helmholtz equation (or time-harmonic wave equation) in the presence of the potential V; ψ_{scat} is known as the scattered field, and ψ_{in} is known as the incident or incoming field (which satisfies an analogue (2.3) of (2.1) with a potential that is zero everywhere).

Inserting (2.2) into (2.1), and then using (2.3), we obtain that

$$(\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})\,\psi_{\text{scat}} = V\psi_{\text{in}} + V\psi_{\text{scat}}. \tag{2.4}$$

Applying the operator $(\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})^{-1}$ to both sides of (2.4), we obtain the Lippmann-Schwinger (or Rayleigh) integral equation

$$\psi_{\text{scat}} = (\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})^{-1}\,V\psi_{\text{in}} + (\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})^{-1}\,V\psi_{\text{scat}}, \tag{2.5}$$

which represents ψ_{scat} in terms of a distorted version of itself, plus a distorted version of ψ_{in}. In the following section, we will express $(\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})^{-1}$ explicitly as an integral operator.

2.2 Green functions

In accordance with the general definition of a Green function, for any integer $d > 1$ and complex number k whose imaginary part is nonnegative, the Green function G_k for $\boldsymbol{\nabla}^2 + k^2\,\mathbf{1}$ on $\mathbb{R}^d$ is the function on $\mathbb{R}^d \times \mathbb{R}^d$ such that

$$(\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})\,G_k(\mathbf{x},\mathbf{y}) = \boldsymbol{\delta}(\mathbf{x}-\mathbf{y}) \tag{2.6}$$

for all $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^d$, where $\boldsymbol{\delta}$ is the Dirac delta, the differential operator $\boldsymbol{\nabla}^2 + k^2\,\mathbf{1}$ acts with respect to $\mathbf{x}$, and $G_k(\mathbf{x},\mathbf{y})$ satisfies what is known as the outgoing Sommerfeld radiation condition as a function of $\mathbf{x}$,

$$\lim_{r\to\infty}\; r^{(d-1)/2}\left(\frac{\partial}{\partial r}G_k(\mathbf{x},\mathbf{y}) - ik\,G_k(\mathbf{x},\mathbf{y})\right) = 0 \tag{2.7}$$

for any $\mathbf{y} \in \mathbb{R}^d$ and uniformly in all directions (of $\mathbf{x}$) as $r \to \infty$, where $r = |\mathbf{x}|$ and $i = \sqrt{-1}$. The Sommerfeld condition ensures that the Green function is unique. For the rationale behind using the term "outgoing," look ahead to Remark 4 in Section 2.3.

In $\mathbb{R}^2$,

$$G_k(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{1}{2\pi}\ln|\mathbf{x}-\mathbf{y}|, & k = 0 \\ -\frac{i}{4}\,H_0^{(1)}(k|\mathbf{x}-\mathbf{y}|), & k \neq 0 \end{cases} \tag{2.8}$$

for any $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^2$ with $\mathbf{x} \neq \mathbf{y}$, where $H_0^{(1)}$ is the Hankel function of the first kind of order 0.

In $\mathbb{R}^3$,

$$G_k(\mathbf{x},\mathbf{y}) = \begin{cases} -\frac{1}{4\pi|\mathbf{x}-\mathbf{y}|}, & k = 0 \\ -\frac{\exp(ik|\mathbf{x}-\mathbf{y}|)}{4\pi|\mathbf{x}-\mathbf{y}|}, & k \neq 0 \end{cases} \tag{2.9}$$

for any $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^3$ with $\mathbf{x} \neq \mathbf{y}$.

It is easy to verify (2.9). The expression for $G_k(\mathbf{x},\mathbf{y})$ clearly has the correct dimensional analysis (that is, the units of $G_k(\mathbf{x},\mathbf{y})$ are the inverse of the length scale, as required in (2.6)). That $\boldsymbol{\nabla}^2 + k^2\,\mathbf{1}$ annihilates $G_k(\mathbf{x},\mathbf{y})$ when $\mathbf{x} \neq \mathbf{y}$ is clear from the expression for the Laplacian (acting on functions of $\mathbf{x}$) in spherical coordinates whose origin is at $\mathbf{y}$:

$$\boldsymbol{\nabla}^2 = \frac{\partial^2}{\partial r^2} + \frac{2}{r}\frac{\partial}{\partial r} + \frac{1}{r^2\,(\sin\theta)^2}\frac{\partial^2}{\partial\varphi^2} + \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2} + \frac{\cos\theta}{r^2\,\sin\theta}\frac{\partial}{\partial\theta}, \tag{2.10}$$

where r and φ denote the radial and azimuthal coordinates, respectively, and θ denotes the polar (zenith) angle ($r = |\mathbf{x}-\mathbf{y}| \geq 0$, $0 \leq \varphi \leq 2\pi$, and $0 \leq \theta \leq \pi$). Similarly, it is easy to verify (2.8), using the differential equation that defines Hankel functions, in conjunction with the expression for the Laplacian (acting on functions of $\mathbf{x}$) in polar coordinates whose origin is at $\mathbf{y}$.

For any integer $d > 1$, and complex number k whose imaginary part is nonnegative, applying $\boldsymbol{\nabla}^2 + k^2\,\mathbf{1}$ to the expressions on both sides of the following identity and using (2.6) verifies that

$$(\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})^{-1} f(\mathbf{x}) = \int_{\mathbb{R}^d} G_k(\mathbf{x},\mathbf{y})\, f(\mathbf{y})\, d\mathbf{y} \tag{2.11}$$

for any $\mathbf{x} \in \mathbb{R}^d$, and any (sufficiently regular) complex-valued function f on $\mathbb{R}^d$.

2.3 How the Helmholtz equation arises from the time-dependent wave equation

Given a positive integer d, a positive real number c, and a function s on $\mathbb{R}^d$ such that $0 < s(\mathbf{x}) \leq c$ for any $\mathbf{x} \in \mathbb{R}^d$, we would like to solve the following equation for the function $\psi = \psi(\mathbf{x}, t)$ on $\mathbb{R}^d \times \mathbb{R}$:

$$\frac{\partial^2}{\partial t^2}\psi = s^2\,\boldsymbol{\nabla}^2\psi. \tag{2.12}$$

Formula (2.12) is known as the time-dependent (scalar) wave equation; the number c is known as the speed of wave propagation in the ambient space, and the function s is known as the local speed of wave propagation.

To solve (2.12), we choose an appropriate nonzero complex number k whose imaginary part is nonnegative, and consider the time-harmonic/Fourier-Laplace-transform ansatz

$$\psi(\mathbf{x}, t) = \psi_k(\mathbf{x})\, e^{-ikct} \tag{2.13}$$

for any $\mathbf{x} \in \mathbb{R}^d$, and $t \in \mathbb{R}$, where ψ_k is a function on $\mathbb{R}^d$.

Inserting (2.13) into (2.12), we obtain the Helmholtz equation

$$(\boldsymbol{\nabla}^2 + k^2\,\mathbf{1})\,\psi_k = V_k\,\psi_k, \tag{2.14}$$

where V_k is the function defined on $\mathbb{R}^d$ via the formula

$$V_k(\mathbf{x}) = k^2\left(1 - \left(\frac{c}{s(\mathbf{x})}\right)^2\right). \tag{2.15}$$

Remark 2 Combining (2.15) and the fact that $s(\mathbf{x}) \leq c$ for any $\mathbf{x} \in \mathbb{R}^d$ yields that $V_k(\mathbf{x}) \leq 0$ for any $\mathbf{x} \in \mathbb{R}^d$ when k is real, guaranteeing that (2.14) does not have any resonances (*i.e.*, bound-states) when k is real and V_k is integrable.

Remark 3 Typically $s(\mathbf{x}) = c$ in "free space," that is, outside of objects that scatter the wave field. Therefore, $V_k(\mathbf{x})$ defined in (2.15) vanishes outside of objects that scatter the wave field.

Remark 4 If k is real and nonzero, then the time dependence in the ansatz (2.13) must be e^{-ikct} rather than e^{ikct} in order to use the corresponding Green functions from Section 2.2. With the time dependence e^{-ikct}, using in (2.11) the Green functions from Section 2.2 ensures that ψ_{scat} from (2.5) is an outgoing field, satisfying the outgoing Sommerfeld radiation condition. If k has a strictly positive imaginary part, then e^{-ikct} blows up as t increases, which is appropriate for the study of Anderson localization.

2.4 Boundary integral equations associated with the Laplace equation

2.4.1 The Dirichlet problem inside a closed curve in $\mathbb{R}^2$

The problem is as follows. Given a (sufficiently regular) real-valued function g on the boundary ∂D of an *open* (and sufficiently regular) simply-connected bounded domain D in $\mathbb{C}$, find a real-valued function f that is continuous on the closure of D, is harmonic (*i.e.*, satisfies the Laplace equation $\boldsymbol{\nabla}^2 f = 0$) on D, and matches g on ∂D (*i.e.*, satisfies $f|_{\partial D} = g$). Please note that (for notational convenience) D is open.

One way to derive an integral equation for the solution is as follows (see Mikhlin [44] for similar derivations). See Kellogg [31], Riesz and Sz.-Nagy [50], or (for more general and involved treatments) the works of Vladimir Maz'ya for existence and uniqueness proofs for the solution.

We define φ to be the holomorphic function on D whose real part is f. We will derive an integral equation on ∂D for a real-valued function $\rho : \partial D \to \mathbb{R}$ such that

$$\varphi(w) = \frac{1}{2\pi i} \int_{\partial D} \frac{\rho(\tilde{z})}{\tilde{z} - w} \, d\tilde{z} \tag{2.16}$$

for any $w \in D$.

By the Sokhotski-Plemelj formula (2.32) derived later, we obtain

$$\lim_{w \to z,\, w \in D} \varphi(w) = \frac{1}{2} \rho(z) + \frac{1}{2\pi i} \, \mathrm{PV} \int_{\partial D} \frac{\rho(\tilde{z})}{\tilde{z} - z} \, d\tilde{z} \tag{2.17}$$

for any $z \in \partial D$ ("PV" denotes the principal value; see, for example, Mikhlin [44] or the proof of Lemma 9).

Taking the real parts of both sides of (2.17), we obtain

$$g(z) = \frac{1}{2} \rho(z) + \frac{1}{2\pi} \, \mathrm{PV} \int_{\partial D} \rho(\tilde{z}) \, \mathrm{Im} \left(\frac{d\tilde{z}}{\tilde{z} - z} \right) \tag{2.18}$$

for any $z \in \partial D$.

For any fixed $z \in \partial D$, we express $\tilde{z} - z$ in polar coordinates as

$$\tilde{z} - z = r(\tilde{z}) \, e^{i\theta(\tilde{z})}, \tag{2.19}$$

where $\theta(\tilde{z})$ is real and $r(\tilde{z})$ is both real and nonnegative, in order to obtain that

$$\mathrm{Im} \left(\frac{d\tilde{z}}{\tilde{z} - z} \right) = \mathrm{Im} \, d \ln(\tilde{z} - z) = d \, \mathrm{Im} \ln(\tilde{z} - z) = d\theta(\tilde{z}) = \left. \frac{\partial [\theta(\tilde{z}(l))]}{\partial l} \right|_{l = l(\tilde{z})} dl(\tilde{z}), \tag{2.20}$$

where $l(\tilde{z})$ is the length of the arc along ∂D going counter-clockwise to $\tilde{z}$, starting from some arbitrary fixed reference point in ∂D, say z, and $\tilde{z}(l)$ is an inverse of $l(\tilde{z})$, *i.e.*, $\tilde{z}(l(\zeta)) = \zeta$ and $l(\tilde{z}(\lambda)) = \lambda$.

Due to the definition of $l(\tilde{z})$ just given, $\partial l(\tilde{z})$ is a unit tangent to ∂D at $\tilde{z}$ and is therefore orthogonal to the outward unit normal to ∂D at $\tilde{z}$, which we denote by $\partial\nu(\tilde{z})$. Therefore, by the Cauchy-Riemann equations for $\psi(\tilde{z}) = \ln(\tilde{z} - z) = \ln r + i\theta$, we have that

$$\left.\frac{\partial[\theta(\tilde{z}(l))]}{\partial l}\right|_{l=l(\tilde{z})} = \frac{\partial}{\partial\nu(\tilde{z})} \ln r, \tag{2.21}$$

where $\frac{\partial}{\partial\nu(\tilde{z})} \ln r$ is the derivative of $\ln r$ with respect to $\tilde{z}$ in the direction of the outward unit normal to ∂D at $\tilde{z}$.

Combining (2.18), (2.19), (2.20), and (2.21), we obtain the boundary integral equation

$$g(z) = \frac{1}{2}\,\rho(z) + \frac{1}{2\pi}\ \mathrm{PV}\int_{\partial D}\left(\frac{\partial}{\partial\nu(\tilde{z})} \ln|\tilde{z} - z|\right)\,\rho(\tilde{z})\,dl(\tilde{z}) \tag{2.22}$$

for any $z \in \partial D$, where $\frac{\partial}{\partial\nu(\tilde{z})} \ln|\tilde{z} - z|$ is the derivative with respect to $\tilde{z}$ of $\ln|\tilde{z} - z|$ in the direction of the outward unit normal to ∂D at $\tilde{z}$, and $l(\tilde{z})$ is the length of the arc along ∂D going counter-clockwise to $\tilde{z}$, starting from some arbitrary fixed reference point in ∂D, say z.

Similarly, combining (2.16) and the fact that f is the real part of φ yields that

$$f(w) = \frac{1}{2\pi}\int_{\partial D}\left(\frac{\partial}{\partial\nu(\tilde{z})} \ln|\tilde{z} - w|\right)\,\rho(\tilde{z})\,dl(\tilde{z}) \tag{2.23}$$

for any $w \in D$, where again $\frac{\partial}{\partial\nu(\tilde{z})} \ln|\tilde{z} - w|$ is the derivative with respect to $\tilde{z}$ of $\ln|\tilde{z} - w|$ in the direction of the outward unit normal to ∂D at $\tilde{z}$, and $l(\tilde{z})$ is the length of the arc along ∂D going counter-clockwise to $\tilde{z}$, starting from some arbitrary fixed reference point in ∂D.

Thus, we can solve (2.22) for ρ on ∂D, and use the result in (2.23) to calculate f on D.

Remark 5 In fact, the integrand in (2.22) is integrable, so the integral exists in the usual sense, not just the principal-value sense. (See, for example, Mikhlin [44] or Colton and Kress [12].)

Remark 6 The function $\ln|\tilde{z} - z|$, where $\tilde{z} \in \partial D$, is known as the single-layer potential for the Laplace equation in $\mathbb{R}^2$. The function $\frac{\partial}{\partial\nu(\tilde{z})} \ln|\tilde{z} - z|$ appearing in (2.22) is known as the double-layer potential for the Laplace equation in $\mathbb{R}^2$. In accordance with the definition of a derivative, the double-layer potential may be regarded as the difference of two infinitely close single-layer potentials.

Remark 7 Direct generalizations to the Maxwell equations of this method for deriving boundary integral equations would seem to require (augmented) Clifford analysis, differential forms, and whatnot (see, for example, McIntosh and Mitrea [41]). The resulting integral equations for space filled with individually homogeneous dielectrics are known as the Müller (or Müller-Weyl) equations, and are derived in Müller [46] using only the usual calculations with vectors (Gaussian "pillboxes," Stokes curl and divergence theorems, and so on). Perfect conductors require special methods when uniqueness of solutions to the integral equations matters — see Epstein and Greengard [19]; these special methods are also critical for the time-harmonic Maxwell equations at very low frequencies, for any media (dielectric, perfectly conducting, etc.).

Remark 8 The solutions to Neumann problems (which are the same as Dirichlet problems, except that you are given the values of the normal derivative of f along ∂D instead of the values of f itself) follow from the solutions to Dirichlet problems via integration by parts, altering the associated boundary integral equations to their adjoints. The solutions to problems involving multiply-connected domains usually appeal to simple variations on the counting principle familiar from complex analysis, topology, and geometrical index theories.

2.4.2 Jump conditions for holomorphic functions

One way to derive the jump conditions used in Subsection 2.4.1 is as follows (see Kilian [32] for similar derivations). We will need the results of a few simple calculations, as stated in the following lemma.

Lemma 9 *Suppose that D is a simply-connected bounded domain in $\mathbb{C}$ such that its boundary ∂D is the image of a circle under a differentiable function, and D itself is open.*

Then,

$$\frac{1}{2\pi i} \lim_{w \to z,\, w \in D} \int_{\partial D} \frac{1}{\tilde{z} - w} \, d\tilde{z} = 1 \tag{2.24}$$

and

$$\frac{1}{2\pi i} \, \mathrm{PV} \int_{\partial D} \frac{1}{\tilde{z} - z} \, d\tilde{z} = \frac{1}{2} \tag{2.25}$$

for any $z \in \partial D$, where "PV" denotes the "principal value" (for a definition of "principal value," see, for example, Mikhlin [44] or the proof of this lemma).

Proof. The Cauchy reproducing formula for the constant function taking the value 1 everywhere yields (2.24) immediately.

By the definition of the principal value of an integral, the left hand side of (2.25) is the limit as r tends to 0 of the quantity

$$\frac{1}{2\pi i} \int \frac{1}{\tilde{z} - z} \, d\tilde{z}, \tag{2.26}$$

with the integral taken along the part of ∂D that does not intersect a disc of radius r about z (*i.e.*, along the arc connecting a to b to c in Figure 2.1). Since $\frac{1}{\tilde{z}-z}$ is holomorphic with respect to $\tilde{z}$ except at $\tilde{z} = z$, the Cauchy theorem yields that (2.26) with the integral taken along the part of ∂D that does not intersect a disc of radius r about z (*i.e.*, along the arc connecting a to b to c in Figure 2.1) has the same value as (2.26) with the integral taken along the part of a circle of radius r about z that intersects D (*i.e.*, along the arc connecting a to d to c in Figure 2.1).

Expressed in polar coordinates centered about z, with

$$\tilde{z} - z = r \, e^{i\theta}, \tag{2.27}$$

(2.26) with the integral taken along the part of a circle of radius r about z that intersects D (*i.e.*, along the arc connecting a to d to c in Figure 2.1) is clearly equal to the quantity

$$\frac{1}{2\pi i} \int_{\theta_a}^{\theta_c} i \, d\theta, \tag{2.28}$$

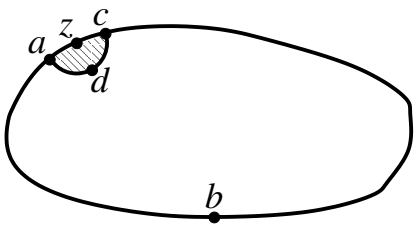

Figure 2.1: The arc connecting a to b to c to z and back to a is ∂D. The arc connecting a to d to c is part of a circle centered about z. The arc connecting a to b to c is the contour of integration in the definition of the principal value of the integral, when the arc connecting c to z to a is omitted from the contour of integration (which would otherwise include all of ∂D). The shaded region is the intersection of D and a disc centered about z.

where

$$a - z = r\, e^{i\theta_a}, \tag{2.29}$$

$$c - z = r\, e^{i\theta_c}, \tag{2.30}$$

and a and c are the points in the intersection of ∂D and the circle of radius r centered about z (see Figure 2.1). Obviously, we have that

$$\lim_{r \to 0} \frac{1}{2\pi i} \int_{\theta_a}^{\theta_c} i \, d\theta = \frac{1}{2}, \tag{2.31}$$

since the part of ∂D that intersects a disc of radius r about z (*i.e.*, the arc connecting a to z to c in Figure 2.1) straightens out as r tends to 0.

Combining the preceding two paragraphs of this proof yields (2.25). □

The following theorem, usually attributed to Sokhotski or Plemelj, provides the limiting value of the Cauchy integral $\varphi(w)$ of a function ρ, where ρ is defined on the boundary of a domain D in $\mathbb{C}$, as w tends to a fixed point z on ∂D through the interior of D. It is easy to relax the requirement stated in the theorem that ρ be continuously differentiable.

Theorem 10 *Suppose that D is a simply-connected bounded domain in $\mathbb{C}$ such that its boundary ∂D is the image of a circle under a differentiable function, and D itself is open. Suppose in addition that $\rho : \partial D \to \mathbb{R}$ is continuously differentiable.*

Then,

$$\lim_{w \to z,\, w \in D} \varphi(w) - \frac{1}{2\pi i} \, \mathrm{PV} \int_{\partial D} \frac{\rho(\tilde{z})}{\tilde{z} - z} \, d\tilde{z} = \frac{1}{2} \, \rho(z) \tag{2.32}$$

for any $z \in \partial D$, where "PV" denotes the "principal value" (see, for example, Mikhlin [44] or the proof of Lemma 9), and $\varphi : D \to \mathbb{C}$ is the function defined via the formula

$$\varphi(w) = \frac{1}{2\pi i} \int_{\partial D} \frac{\rho(\tilde{z})}{\tilde{z} - w} \, d\tilde{z} \tag{2.33}$$

for any $w \in D$.

Proof. By adding and subtracting the same quantity to and from the right-hand side of (2.33), we obtain that

$$\varphi(w) = \frac{1}{2\pi i} \int_{\partial D} \frac{\rho(\tilde{z}) - \rho(z)}{\tilde{z} - w}\, d\tilde{z} + \rho(z)\, \frac{1}{2\pi i} \int_{\partial D} \frac{1}{\tilde{z} - w}\, d\tilde{z} \tag{2.34}$$

for any $w \in D$ and $z \in \partial D$. Since ρ is continuously differentiable, the quantity

$$\frac{\rho(\tilde{z}) - \rho(z)}{\tilde{z} - z} \tag{2.35}$$

is bounded for any $\tilde{z}$ and z in ∂D, hence is integrable, and so

$$\mathrm{PV} \int_{\partial D} \frac{\rho(\tilde{z}) - \rho(z)}{\tilde{z} - z}\, d\tilde{z} = \int_{\partial D} \frac{\rho(\tilde{z}) - \rho(z)}{\tilde{z} - z}\, d\tilde{z} \tag{2.36}$$

for any $z \in \partial D$. Taking the limits of both sides of (2.34) as w tends to z, and noting that (2.35) is integrable, we get that

$$\lim_{w \to z,\, w \in D} \varphi(w) = \frac{1}{2\pi i} \int_{\partial D} \frac{\rho(\tilde{z}) - \rho(z)}{\tilde{z} - z}\, d\tilde{z} + \rho(z)\, \frac{1}{2\pi i} \lim_{w \to z,\, w \in D} \int_{\partial D} \frac{1}{\tilde{z} - w}\, d\tilde{z} \tag{2.37}$$

for any $z \in \partial D$.

Similarly, by adding and subtracting the same quantity to and from the right-hand side of (2.33), and taking the principal-value limits of both sides, we obtain that

$$\frac{1}{2\pi i}\, \mathrm{PV} \int_{\partial D} \frac{\rho(\tilde{z})}{\tilde{z} - z}\, d\tilde{z} = \frac{1}{2\pi i}\, \mathrm{PV} \int_{\partial D} \frac{\rho(\tilde{z}) - \rho(z)}{\tilde{z} - z}\, d\tilde{z} + \rho(z)\, \frac{1}{2\pi i}\, \mathrm{PV} \int_{\partial D} \frac{1}{\tilde{z} - z}\, d\tilde{z} \tag{2.38}$$

for any $z \in \partial D$.

Subtracting (2.38) from (2.37) and then applying (2.36), (2.24), and (2.25), we obtain (2.32). □

Chapter 3

Discretization of integral equations

In this chapter, we discretize an integral equation, via the Nyström method. To start, we discuss discretization in general, not necessarily using the Nyström method:

3.1 Overview

Physical systems are usually modeled in terms of continuous variables, whereas the digital computers which perform our calculations deal only with discrete data. Therefore, in order to use computers to model physical systems, we must somehow ensure that our numerical calculations yield good approximations to the desired values of continuous variables. The means for achieving this is known as *discretization*. We will discuss discretization only very briefly in the present notes, referring the reader to standard presentations of the Nyström, Galerkin, collocation, and qualocation methods (see, for example, Atkinson [3]).

We caution that methods for discretization remain under intensive development. There are many subtleties involved in discretization, as demonstrated for instance by spurious-resonance/fictitious-eigenfrequency problems (see Epstein and Greengard [19]). The boundaries of domain geometries encountered in practice often have corners, edges, and rough surfaces, complicating their discretization. Beware! The most commonly used procedures for discretization in electromagnetics, the curl- or divergence-conforming (flux-conservative) Rao-Wilton-Glisson "rooftop" Galerkin schemes, are only first-order accurate. Bremer [5], Bremer et al. [6], Helsing [27], Kloeckner et al. [33], Yarvin and Rokhlin [57], and others treat many issues involved with obtaining high-order discretization schemes.

3.2 The Dirichlet problem for the Laplace equation inside a closed curve in $\mathbb{R}^2$

In this section, we summarize the integral-equation formulation of the Dirichlet problem for the Laplace equation inside a closed curve in $\mathbb{R}^2$, derived in the previous chapter. We then discretize this formulation, in the last section of the present chapter. We will be identifying points in $\mathbb{R}^2$ with numbers in the complex plane $\mathbb{C}$, in the standard fashion.

The problem is as follows. Given a (sufficiently regular) real-valued function g on the boundary ∂D of an *open* (and sufficiently regular) simply-connected bounded domain D in $\mathbb{C}$, find a real-valued function f that is continuous on the closure of D, is harmonic (*i.e.*, satisfies the Laplace equation $\boldsymbol{\nabla}^2 f = 0$) on D, and matches g on ∂D (*i.e.*, satisfies $f|_{\partial D} = g$).

As described earlier (in Subsection 2.4.1), we obtain f via the formula

$$f(w) = \frac{1}{2\pi} \int_{\partial D} h(w, \tilde{z})\, \rho(\tilde{z})\, dl(\tilde{z}) \tag{3.1}$$

for any $w \in D$, where

$$h(w, \tilde{z}) = \frac{\partial}{\partial \nu(\tilde{z})} \ln |\tilde{z} - w|, \tag{3.2}$$

$\frac{\partial}{\partial \nu(\tilde{z})} \ln |\tilde{z} - w|$ is the derivative with respect to $\tilde{z}$ of $\ln |\tilde{z} - w|$ in the direction of the outward unit normal to ∂D at $\tilde{z}$, and $l(\tilde{z})$ is the length of the arc along ∂D going counter-clockwise to $\tilde{z}$, starting from some arbitrary fixed reference point in ∂D. We obtain the real-valued ρ in (3.1) by solving

$$\frac{1}{2}\, \rho(z) + \frac{1}{2\pi} \int_{\partial D} h(z, \tilde{z})\, \rho(\tilde{z})\, dl(\tilde{z}) = g(z) \tag{3.3}$$

for any $z \in \partial D$, where again $h(z, \tilde{z})$ is defined as in (3.2), and $l(\tilde{z})$ is the length of the arc along ∂D going counter-clockwise to $\tilde{z}$, starting from some arbitrary fixed reference point in ∂D, say z.

Remark 11 If ∂D is sufficiently regular, then $h(z, \tilde{z})$ defined as in (3.2) is smooth as a function of $\tilde{z} \in \partial D$, even when $z \in \partial D$. Similarly, if ∂D and g are sufficiently regular, then the solution ρ to (3.3) is smooth. (See, for example, Colton and Kress [12].)

3.3 Nyström methods

Trapezoidal quadrature of order n involves n points z_1, z_2, $\ldots$, z_{n-1}, z_n from ∂D that are equispaced in terms of arc length along ∂D, and provides a highly accurate approximation

$$\int_{\partial D} \varphi(\tilde{z})\, dl(\tilde{z}) \approx \frac{L}{n} \sum_{k=1}^{n} \varphi(z_k) \tag{3.4}$$

for any function φ that is smooth on the sufficiently regular ∂D, where L is the length of ∂D, and $l(\tilde{z})$ is the length of the arc along ∂D going counter-clockwise to $\tilde{z}$, starting from some arbitrary fixed reference point in ∂D; for precise characterizations of the accuracy of the approximation in (3.4), see, for example, Atkinson [3]. Recalling Remark 11, we may replace the integral in (3.3) with its approximation via trapezoidal quadrature to obtain

$$\frac{1}{2}\, \rho(z) + \frac{L}{2\pi n} \sum_{k=1}^{n} h(z, z_k)\, \rho(z_k) \approx g(z) \tag{3.5}$$

for any $z \in \partial D$, where h is defined as in (3.2), and L is the length of ∂D. In particular, we can enforce (3.5) for $z = z_1, z_2, \ldots, z_{n-1}, z_n$, obtaining $\rho(z_1), \rho(z_2), \ldots, \rho(z_{n-1}), \rho(z_n)$ as the solution to the system of linear equations

$$\frac{1}{2}\,\rho(z_j) + \frac{L}{2\pi n} \sum_{k=1}^{n} h(z_j, z_k)\,\rho(z_k) = g(z_j) \tag{3.6}$$

for $j = 1, 2, \ldots, n-1, n$, where again h is defined as in (3.2), and L is the length of ∂D. Similarly, replacing the integral in (3.1) with its approximation via trapezoidal quadrature, we obtain

$$f(w) \approx \frac{L}{2\pi n} \sum_{k=1}^{n} h(w, z_k)\,\rho(z_k) \tag{3.7}$$

for any $w \in D$, where again h is defined in (3.2), L is the length of ∂D, and $\rho(z_1), \rho(z_2), \ldots, \rho(z_{n-1}), \rho(z_n)$ are solutions to (3.6).

Remark 12 If the kernel in an integral equation is singular, then the trapezoidal rule requires corrections in order to provide a high-order quadrature; see Duan and Rokhlin [16] and Kapur and Rokhlin [29]. Discretizing an integral equation whose kernel is singular is entirely similar to the procedure outlined in the present section, aside from the need for corrections to the trapezoidal quadrature.

Remark 13 The simple scheme for discretization described in the present section is known as the Nyström method. Other possibilities for discretization include the Galerkin and collocation methods. The Galerkin method often involves finite or boundary elements.

Chapter 4

Iterative methods for solving systems of linear-algebraic equations

In this chapter, we describe iterative methods for solving systems of linear equations, such as (3.6). Iterative methods are efficient when the matrix associated with the system of linear-algebraic equations is well-conditioned and can be applied rapidly to arbitrary vectors; Chapter 8 describes methods for such rapid applications (specifically, for the matrices associated with (3.6) and similar equations).

4.1 Simplest/stationary

The simplest iterative methods for solving systems of linear equations are Neumann/Born series and Chebyshev approximations.

If the norm of the difference of a linear operator $\mathbf{A}$ from the identity operator $\mathbf{1}$, *i.e.*,

$$\|\mathbf{1} - \mathbf{A}\| = \max_{\mathbf{x} \neq 0} \frac{\|(\mathbf{1} - \mathbf{A})\,\mathbf{x}\|}{\|\mathbf{x}\|}, \tag{4.1}$$

is strictly less than 1, then the problem

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}, \tag{4.2}$$

where $\mathbf{b}$ is an element in the range of $\mathbf{A}$, has the solution

$$\mathbf{x} = \mathbf{b} + \mathbf{T}\,\mathbf{b} + \mathbf{T}(\mathbf{T}\,\mathbf{b}) + \mathbf{T}(\mathbf{T}(\mathbf{T}\,\mathbf{b})) + \ldots, \tag{4.3}$$

where $\mathbf{T} = \mathbf{1} - \mathbf{A}$ (after all, $\mathbf{1} + \mathbf{T} + \mathbf{T}^2 + \mathbf{T}^3 + \ldots$ is a geometric series whose sum is $(\mathbf{1} - \mathbf{T})^{-1} = \mathbf{A}^{-1}$; the series converges since $\|\mathbf{T}\| < 1$). The right-hand side of (4.3) is known as the Neumann/Born series.

When applying $\mathbf{A}$ is fairly expensive (which is normally so), we can obtain a sensible approximation to the solution $\mathbf{x}$ by calculating the truncation of the Neumann/Born series to (say) n terms, a calculation which entails just $n-1$ applications of $\mathbf{A}$. However, truncating the Neumann/Born series is not nearly as efficient as using the optimized Krylov subspace methods that Section 4.2 discusses. Methods known as Chebyshev acceleration improve upon simply truncating the Neumann/Born series, but such techniques generally are not competitive with those discussed in the following section.

4.2 Krylov-subspace-based/non-stationary

Suppose that we want to solve the system of linear-algebraic equations (4.2) for $\mathbf{x}$, as accurately as possible, by first applying $\mathbf{A}$ a total of n times to vectors of our choosing, and then forming linear combinations of the resulting vectors; this is about the best we can do when we expect the costs of applying $\mathbf{A}$ to dominate, and cannot afford to do anything with $\mathbf{A}$ except apply it n times to vectors. Since the only known vector associated with (4.2) is $\mathbf{b}$, we thus need to find an algorithm that computes the vector $\mathbf{x} = p(\mathbf{A})\,\mathbf{b}$ which minimizes $\|\mathbf{b} - \mathbf{A}\,\mathbf{x}\|$ over every polynomial p of degree at most $n-1$.

For $n = 0,\, 1,\, 2,\, \ldots$, the $(n+1)^{\text{st}}$ Krylov subspace for $\mathbf{A}$ and $\mathbf{b}$ is the range of $p(\mathbf{A})\,\mathbf{b}$ over every polynomial p of degree at most n. We now minimize over these Krylov subspaces.

Given any positive integer n such that the vectors $\mathbf{A}^0\,\mathbf{b},\ \mathbf{A}^1\,\mathbf{b},\ \ldots,\ \mathbf{A}^{n-1}\,\mathbf{b},\ \mathbf{A}^n\,\mathbf{b}$ are linearly independent, we apply the Gram-Schmidt process in order to obtain orthonormal vectors $\mathbf{u}_1,\, \mathbf{u}_2,\, \ldots,\, \mathbf{u}_n,\, \mathbf{u}_{n+1}$ forming a basis for the $(n+1)^{\text{st}}$ Krylov subspace, defining first

$$\mathbf{u}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}, \tag{4.4}$$

and then $\mathbf{u}_{k+1}$ and $H_{l,k}$ for each $k = 1,\, 2,\, \ldots,\, n-1,\, n$, via the following formulae:

$$H_{l,k} = \langle \mathbf{u}_l,\, \mathbf{A}\,\mathbf{u}_k \rangle \tag{4.5}$$

for $l = 1,\, 2,\, \ldots,\, k-1,\, k$,

$$\mathbf{w}_k = \mathbf{A}\,\mathbf{u}_k - \sum_{l=1}^{k} \mathbf{u}_l\,\langle \mathbf{u}_l,\, \mathbf{A}\,\mathbf{u}_k \rangle = \mathbf{A}\,\mathbf{u}_k - \sum_{l=1}^{k} \mathbf{u}_l\,H_{l,k}, \tag{4.6}$$

$$H_{k+1,k} = \|\mathbf{w}_k\|, \tag{4.7}$$

$$\mathbf{u}_{k+1} = \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|} = \frac{\mathbf{w}_k}{H_{k+1,k}}, \tag{4.8}$$

and

$$H_{l,k} = 0 \tag{4.9}$$

for $l = k+2,\, k+3,\, \ldots,\, n,\, n+1$, where $\|\boldsymbol{\alpha}\|$ denotes the norm of $\boldsymbol{\alpha}$, and $\langle \boldsymbol{\beta}, \boldsymbol{\gamma} \rangle$ denotes the inner product of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ associated with the norm such that $\langle c\,\boldsymbol{\beta}, \boldsymbol{\gamma} \rangle = \overline{c}\,\langle \boldsymbol{\beta}, \boldsymbol{\gamma} \rangle$ for any vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$, and complex number c. (We require $\|\boldsymbol{\alpha}\|^2 = \langle \boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle$ for any vector $\boldsymbol{\alpha}$.)

Combining (4.6), (4.7), and (4.8) yields that

$$\mathbf{A}\,\mathbf{u}_k = \mathbf{u}_{k+1}\,H_{k+1,k} + \sum_{l=1}^{k} \mathbf{u}_l\,H_{l,k} \tag{4.10}$$

for $k = 1,\, 2,\, \ldots,\, n-1,\, n$. Defining $\mathbf{U}$ to be the matrix with $n+1$ columns whose k^{th} column is $\mathbf{u}_k$ for $k = 1,\, 2,\, \ldots,\, n,\, n+1$, and $\mathbf{V}$ to be the matrix with n columns whose k^{th} column is $\mathbf{u}_k$ for $k = 1,\, 2,\, \ldots,\, n-1,\, n$, we obtain from (4.10) that

$$\mathbf{A}\,\mathbf{V} = \mathbf{U}\,\mathbf{H}, \tag{4.11}$$

where $\mathbf{H}$ is the $(n+1) \times n$ matrix whose entries are defined in (4.5), (4.7), and (4.9). ($\mathbf{H}$ is upper Hessenberg, that is, (4.9) holds for $\mathbf{H}$.)

Suppose now that $\mathbf{x}$ is a linear combination of $\mathbf{A}^0\,\mathbf{b}$, $\mathbf{A}^1\,\mathbf{b}$, $\ldots$, $\mathbf{A}^{n-2}\,\mathbf{b}$, $\mathbf{A}^{n-1}\,\mathbf{b}$, *i.e.*, that there is an $n \times 1$ column vector $\mathbf{y}$ such that

$$\mathbf{x} = \mathbf{V}\,\mathbf{y}, \tag{4.12}$$

so that (since $\mathbf{u}_1$, $\mathbf{u}_2$, $\ldots$, $\mathbf{u}_{n-1}$, $\mathbf{u}_n$ are orthonormal and are the columns of $\mathbf{V}$) the k^{th} entry of $\mathbf{y}$ is $y_k = \langle \mathbf{u}_k, \mathbf{x} \rangle$ for $k = 1, 2, \ldots, n-1, n$. Then, combining (4.4), (4.11), and (4.12) yields that

$$\mathbf{b} - \mathbf{A}\,\mathbf{x} = \mathbf{U}\,(\mathbf{e} - \mathbf{H}\,\mathbf{y}), \tag{4.13}$$

where $\mathbf{e}$ is the $(n+1) \times 1$ column vector defined by the formula

$$\mathbf{e} = \begin{pmatrix} \|\mathbf{b}\| \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \tag{4.14}$$

and $\mathbf{H}$ is the $(n+1) \times n$ matrix whose entries are defined in (4.5), (4.7), and (4.9).

Combining (4.13) and the fact that the columns $\mathbf{u}_1$, $\mathbf{u}_2$, $\ldots$, $\mathbf{u}_n$, $\mathbf{u}_{n+1}$ of $\mathbf{U}$ are obtained from the Gram-Schmidt process as in (4.4)–(4.9) and are therefore orthonormal, we obtain that

$$\|\mathbf{b} - \mathbf{A}\,\mathbf{x}\| = \|\mathbf{e} - \mathbf{H}\,\mathbf{y}\|, \tag{4.15}$$

where again $\mathbf{x} = \mathbf{V}\,\mathbf{y} = \sum_{k=1}^{n} \mathbf{u}_k\,y_k$, $\mathbf{e}$ is the $(n+1) \times 1$ column vector defined in (4.14), and $\mathbf{H}$ is the $(n+1) \times n$ matrix whose entries are defined in (4.5), (4.7), and (4.9). (Needless to say, $\|\mathbf{e} - \mathbf{H}\,\mathbf{y}\|$ is the Euclidean norm of $\mathbf{e} - \mathbf{H}\,\mathbf{y}$.) Notice that the number of entries in the vector $\mathbf{e} - \mathbf{H}\,\mathbf{y}$ in the right-hand side of (4.15) is only $n+1$, whereas the vector $\mathbf{b} - \mathbf{A}\,\mathbf{x}$ in the left-hand side of (4.15) could be extremely long.

Thus, in order to compute the vector $\mathbf{x} = p(\mathbf{A})\,\mathbf{b}$ which minimizes $\|\mathbf{b} - \mathbf{A}\,\mathbf{x}\|$ over all polynomials p of degree at most $n-1$, or, equivalently, over all linear combinations of the vectors $\mathbf{A}^0\,\mathbf{b}$, $\mathbf{A}^1\,\mathbf{b}$, $\ldots$, $\mathbf{A}^{n-2}\,\mathbf{b}$, $\mathbf{A}^{n-1}\,\mathbf{b}$, we can use what is known as the Generalized Minimum RESidual (GMRES) algorithm:

(1) Conduct the Gram-Schmidt process described in (4.4)–(4.9) to obtain the vectors $\mathbf{u}_1$, $\mathbf{u}_2$, $\ldots$, $\mathbf{u}_n$, $\mathbf{u}_{n+1}$ and the $(n+1) \times n$ matrix $\mathbf{H}$.

(2) Compute the $n \times 1$ vector $\mathbf{y}$ which minimizes the Euclidean norm $\|\mathbf{e} - \mathbf{H}\,\mathbf{y}\|$, with $\mathbf{e}$ as in (4.14). (For details, see, for example, Chapter 5 of Golub and Van Loan [22].)

(3) Calculate $\mathbf{x} = \mathbf{V}\,\mathbf{y} = \sum_{k=1}^{n} \mathbf{u}_k\,y_k$.

See Saad [53] for a similar, more comprehensive discussion of this GMRES and related algorithms. In practice we usually adjust n adaptively, setting $n = 1$, then $n = 2$, then $n = 3$, and so on, until the solution $\mathbf{x}$ becomes sufficiently accurate.

Remark 14 GMRES is not exorbitantly costly. However, GMRES simplifies when $\mathbf{A}$ is self-adjoint. If $\mathbf{A}$ is self-adjoint, then combining (4.5), (4.10), and the fact that $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_{n-1}$, $\mathbf{u}_n$ are orthonormal yields that $H_{l,k} = \langle \mathbf{A}\,\mathbf{u}_l, \mathbf{u}_k \rangle = 0$ for $l = 1, 2, \ldots, k-1, k-2$. What is known as the conjugate residual or MINimum RESidual (MINRES) algorithm takes advantage of this observation. When $\mathbf{A}$ is strictly positive definite, and at the expense of replacing the bona-fide least-squares solutions from the second and third steps of GMRES with easier-to-calculate approximations, what is known as the conjugate gradient algorithm simplifies the computational procedures used in the conjugate residual algorithm, and reduces the (forward) error in the solution vector $\mathbf{x}$ it produces.

Remark 15 Obviously, when the vectors $\mathbf{A}^1\,\mathbf{b}$, $\mathbf{A}^2\,\mathbf{b}$, ..., $\mathbf{A}^{n-1}\,\mathbf{b}$, $\mathbf{A}^n\,\mathbf{b}$ span the entire range of $\mathbf{A}$, the solution $\mathbf{x}$ that GMRES computes minimizes $\|\mathbf{A}\,\tilde{\mathbf{x}} - \mathbf{b}\|$ over all possible vectors $\tilde{\mathbf{x}}$. Unfortunately, however, there is little theoretical understanding of the rate of convergence for the GMRES iterations. Even so, GMRES often converges rapidly in practice (see, for example, Rokhlin [51] or Bruno et al. [7]; Rokhlin [51] refers to GMRES as the generalized conjugate residual algorithm — GCR or GCRA — which is a mathematically equivalent formulation). Theoretical analyses of the conjugate gradient and residual schemes described in Remark 14 are far better developed (see, for example, Tyrtyshnikov [55]), and these latter schemes work very well when $\mathbf{A}$ is self-adjoint. (However, matrices associated with discretizations of boundary integral equations are seldom self-adjoint.)

Remark 16 To avoid problems with roundoff errors, replace (4.5) and (4.6) with suitable reorthogonalization procedures, such as the following:

$$F_{l,k} = \langle \mathbf{u}_l, \mathbf{A}\,\mathbf{u}_k \rangle \tag{4.16}$$

for $l = 1, 2, \ldots, k-1, k$,

$$\mathbf{r}_k = \mathbf{A}\,\mathbf{u}_k - \sum_{l=1}^{k} \mathbf{u}_l\,F_{l,k}, \tag{4.17}$$

$$G_{l,k} = \langle \mathbf{u}_l, \mathbf{r}_k \rangle \tag{4.18}$$

for $l = 1, 2, \ldots, k-1, k$,

$$\mathbf{w}_k = \mathbf{r}_k - \sum_{l=1}^{k} \mathbf{u}_l\,G_{l,k}, \tag{4.19}$$

and

$$H_{l,k} = F_{l,k} + G_{l,k} \tag{4.20}$$

for $l = 1, 2, \ldots, k-1, k$. In exact arithmetic, $G_{l,k} = 0$ for $l = 1, 2, \ldots, k-1, k$.

Remark 17 Sometimes a good guess is available for the solution $\mathbf{x}$ to the system of linear equations (4.2). A simple modification of the GMRES procedure described above can take advantage of the guess; see, for example, Saad [53].

Chapter 5

Estimating the spectral norm of a matrix

In this chapter, we describe a method for estimating the spectral norm of a matrix (recall that the spectral norm $\|\mathbf{A}\|$ of a matrix $\mathbf{A}$ is $\|\mathbf{A}\| = \max_{\mathbf{x}\neq\mathbf{0}} \|\mathbf{A}\mathbf{x}\|/\|\mathbf{x}\|$, with the maximum taken over all nonzero vectors, where $\|\mathbf{A}\mathbf{x}\|$ is the Euclidean norm of $\mathbf{A}\mathbf{x}$, and $\|\mathbf{x}\|$ is the Euclidean norm of $\mathbf{x}$). This allows us to check the accuracy of implementations of the algorithms of Chapters 6 and 7. Chapter 8 uses these algorithms extensively in order to "compress" the linear operators used in GMRES of Chapter 4 to solve discretized integral equations such as (3.6). Such compression involves building an approximation $\mathbf{C}$ to a matrix $\mathbf{B}$ ($\mathbf{B}$ is frequently a block of another, larger matrix), such that $\mathbf{C}$ can be applied efficiently to arbitrary vectors. The accuracy of the compressed approximation is good when the spectral norm $\|\mathbf{A}\|$ of the difference $\mathbf{A} = \mathbf{B} - \mathbf{C}$ is small. To assess the quality of an approximation $\mathbf{C}$ to $\mathbf{B}$, we need to be able to estimate the spectral norm $\|\mathbf{A}\|$ of $\mathbf{A} = \mathbf{B} - \mathbf{C}$. Most often we would like $\|\mathbf{A}\|/\|\mathbf{B}\|$ to be near the machine precision. In such cases, we need a way to ascertain that $\|\mathbf{A}\|/\|\mathbf{B}\|$ is really small (say, within a few digits of the machine precision), but do not necessarily need to estimate $\|\mathbf{A}\|/\|\mathbf{B}\|$ to high relative accuracy. For this, the simple power method is effective; the more complicated Lanczos method described, for example, by Kuczyński and Woźniakowski [35] and Golub and Van Loan [22] can also be suitable.

The following theorem states that the power method provides an efficient means of estimating the spectral norm $\|\mathbf{A}\|$ of a matrix $\mathbf{A}$. The estimate $p_j(\mathbf{A})$ produced by the power method never exceeds $\|\mathbf{A}\|$ and is very rarely much less than $\|\mathbf{A}\|$. Computing the estimate $p_j(\mathbf{A})$ requires only applications of $\mathbf{A}$ and $\mathbf{A}^*$ to vectors. The theorem is a slight reformulation of Theorem 4.1(a) of Kuczyński and Woźniakowski [35] (which provides a proof).

Theorem 18 *Suppose that j, m, and n are positive integers, $\mathbf{A}$ is an $m \times n$ matrix, $\boldsymbol{\omega}$ is an $n \times 1$ vector whose entries are i.i.d. centered Gaussian random variables, and*

$$p_j(\mathbf{A}) = \frac{\|(\mathbf{A}^*\,\mathbf{A})^j\,\boldsymbol{\omega}\|}{\|\mathbf{A}\,(\mathbf{A}^*\,\mathbf{A})^{j-1}\,\boldsymbol{\omega}\|}. \tag{5.1}$$

Then,

$$p_j(\mathbf{A}) \leq \|\mathbf{A}\|. \tag{5.2}$$

Furthermore,

$$p_j(\mathbf{A}) \geq (1-\varepsilon)\,\|\mathbf{A}\| \tag{5.3}$$

with probability at least

$$1 - \min\left(0.824, \frac{0.354}{\sqrt{\varepsilon\,(2j-1)}}\right)\sqrt{n}\,(1-\varepsilon)^{2j-1} \tag{5.4}$$

for any positive real number $\varepsilon < 1$.

Remark 19 With $\varepsilon = 0.5$, (5.4) becomes at least

$$1 - \sqrt{\frac{n}{2j-1}}\;4^{-j}. \tag{5.5}$$

With $\varepsilon = 0.9$, (5.4) becomes at least

$$1 - 4\sqrt{\frac{n}{2j-1}}\;100^{-j}. \tag{5.6}$$

Remark 20 The bound (5.4) on the probability of success does not depend on the structure of the spectrum of the matrix $\mathbf{A}$ whose spectral norm is being estimated. Gaps between the singular values may improve the performance, but are not necessary to produce the guarantees stated in Theorem 18.

Generally, we compute $p_j(\mathbf{A})$ defined in (5.1) by constructing the following sequence:

$$\begin{aligned}
\mathbf{v}^{(0)} &= \boldsymbol{\omega}, && (5.7)\\
\mathbf{v}^{(1)} &= \mathbf{A}\,\mathbf{v}^{(0)}/\|\mathbf{v}^{(0)}\|, && (5.8)\\
\mathbf{v}^{(2)} &= \mathbf{A}^*\,\mathbf{v}^{(1)}/\|\mathbf{v}^{(1)}\|, && (5.9)\\
\mathbf{v}^{(3)} &= \mathbf{A}\,\mathbf{v}^{(2)}/\|\mathbf{v}^{(2)}\|, && (5.10)\\
\mathbf{v}^{(4)} &= \mathbf{A}^*\,\mathbf{v}^{(3)}/\|\mathbf{v}^{(3)}\|, && (5.11)\\
&\vdots &&\\
\mathbf{v}^{(2j-3)} &= \mathbf{A}\,\mathbf{v}^{(2j-4)}/\|\mathbf{v}^{(2j-4)}\|, && (5.12)\\
\mathbf{v}^{(2j-2)} &= \mathbf{A}^*\,\mathbf{v}^{(2j-3)}/\|\mathbf{v}^{(2j-3)}\|, && (5.13)\\
\mathbf{v}^{(2j-1)} &= \mathbf{A}\,\mathbf{v}^{(2j-2)}/\|\mathbf{v}^{(2j-2)}\|, && (5.14)\\
\mathbf{v}^{(2j)} &= \mathbf{A}^*\,\mathbf{v}^{(2j-1)}/\|\mathbf{v}^{(2j-1)}\|, && (5.15)\\
p_j(\mathbf{A}) &= \|\mathbf{v}^{(2j)}\|, && (5.16)
\end{aligned}$$

where $\mathbf{A}$ is the $m \times n$ matrix whose spectral norm $\|\mathbf{A}\|$ is being estimated by $p_j(\mathbf{A})$, and $\boldsymbol{\omega}$ is an $n \times 1$ vector whose entries are i.i.d. centered Gaussian random variables.

Chapter 6

The singular value decomposition (SVD)

In this chapter, we describe the low-rank approximation of matrices via the singular value decomposition (SVD), following the presentation in Golub and Van Loan [22]. This provides a theoretical basis for understanding the interpolative decomposition (ID) of Chapter 7, which we use extensively in Chapter 8. For proof of the existence of the SVDs used below, as well as descriptions of reasonably efficient algorithms for computing SVDs, see, for example, Golub and Van Loan [22].

6.1 Notation

Suppose that m and n are positive integers, and $\mathbf{A}$ is an $m \times n$ matrix. We define

$$l = \min(m, n). \tag{6.1}$$

The full SVD of $\mathbf{A}$ consists of a unitary $m \times m$ matrix $\mathbf{U}^{(\text{full})}$, a unitary $n \times n$ matrix $\mathbf{V}^{(\text{full})}$, and an $m \times n$ matrix $\mathbf{\Sigma}^{(\text{full})}$ whose only nonzero entries are nonnegative and appear in nonincreasing order on the main diagonal, such that

$$\mathbf{A}_{m \times n} = \mathbf{U}^{(\text{full})}_{m \times m} \cdot \mathbf{\Sigma}^{(\text{full})}_{m \times n} \cdot (\mathbf{V}^{(\text{full})}_{n \times n})^*. \tag{6.2}$$

The columns of $\mathbf{U}^{(\text{full})}$ are known as the left singular vectors of $\mathbf{A}$; the columns of $\mathbf{V}^{(\text{full})}$ are known as the right singular vectors of $\mathbf{A}$. The entries on the main diagonal of $\mathbf{\Sigma}^{(\text{full})}$ are known as the singular values of $\mathbf{A}$.

The thin SVD of $\mathbf{A}$ consists of an $m \times l$ matrix $\mathbf{U}^{(\text{thin})}$ whose columns are orthonormal, an $n \times l$ matrix $\mathbf{V}^{(\text{thin})}$ whose columns are orthonormal, and a diagonal $l \times l$ matrix $\mathbf{\Sigma}^{(\text{thin})}$ whose only nonzero entries are nonnegative and appear in nonincreasing order on the diagonal, such that

$$\mathbf{A}_{m \times n} = \mathbf{U}^{(\text{thin})}_{m \times l} \cdot \mathbf{\Sigma}^{(\text{thin})}_{l \times l} \cdot (\mathbf{V}^{(\text{thin})}_{n \times l})^*, \tag{6.3}$$

where l is defined in (6.1). The thin SVD is also known as the "economy-size" or reduced SVD.

Denoting the rank of $\mathbf{A}$ by k, the compact SVD consists of an $m \times k$ matrix $\mathbf{U}^{(\text{comp})}$ whose columns are orthonormal, an $n \times k$ matrix $\mathbf{V}^{(\text{comp})}$ whose columns are orthonormal, and a diagonal $k \times k$ matrix $\mathbf{\Sigma}^{(\text{comp})}$ whose only nonzero entries are nonnegative and appear in nonincreasing order on the diagonal, such that

$$\mathbf{A}_{m \times n} = \mathbf{U}^{(\text{comp})}_{m \times k} \cdot \mathbf{\Sigma}^{(\text{comp})}_{k \times k} \cdot (\mathbf{V}^{(\text{comp})}_{n \times k})^*, \tag{6.4}$$

where again k is the rank of $\mathbf{A}$.

6.2 Low-rank approximation

The SVD provides a means for characterizing the best low-rank approximations to a matrix, as well as for constructing such approximations. The following theorem states that the spectral norm of the difference between a matrix $\mathbf{A}$ and its best rank-k approximation is simply the $(k+1)^{\text{st}}$ greatest singular value $\sigma_{k+1}(\mathbf{A})$ of $\mathbf{A}$.

Theorem 21 *Suppose that k, m, and n are positive integers, with $k < m$ and $k < n$, and $\mathbf{A}$ is an $m \times n$ matrix.*

Then,

$$\min \|\mathbf{A} - \mathbf{B}\| = \sigma_{k+1}(\mathbf{A}), \tag{6.5}$$

where the minimum is taken over all $m \times n$ matrices $\mathbf{B}$ whose rank is at most k, $\sigma_{k+1}(\mathbf{A})$ is the $(k+1)^{st}$ greatest singular value of $\mathbf{A}$ (counting multiplicity), and $\|\mathbf{A} - \mathbf{B}\|$ is the spectral norm of $\mathbf{A} - \mathbf{B}$,

$$\|\mathbf{A} - \mathbf{B}\| = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|(\mathbf{A} - \mathbf{B})\,\mathbf{x}\|}{\|\mathbf{x}\|}, \tag{6.6}$$

with the maximum taken over all nonzero vectors of length n, and with $\|\cdot\|$ in the right-hand side of (6.6) denoting the Euclidean norm (the spectral norm of a matrix is equal to the greatest singular value of the matrix).

Proof. We start by forming the thin SVD (6.3) of $\mathbf{A}$. We define $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$, $\ldots$, $\mathbf{u}^{(l-1)}$, $\mathbf{u}^{(l)}$ to be the columns of $\mathbf{U}^{(\text{thin})}$, we define $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, $\ldots$, $\mathbf{v}^{(l-1)}$, $\mathbf{v}^{(l)}$ to be the columns of $\mathbf{V}^{(\text{thin})}$, and we define σ_1, σ_2, $\ldots$, σ_{l-1}, σ_l to be the diagonal entries of $\mathbf{\Sigma}^{(\text{thin})}$, where $l = \min(m, n)$.

If we set $\mathbf{B} = \sum_{j=1}^{k} \sigma_j\, \mathbf{u}^{(j)}\, (\mathbf{v}^{(j)})^*$, then combining (6.3) (that is, $\mathbf{A} = \sum_{j=1}^{l} \sigma_j\, \mathbf{u}^{(j)}\, (\mathbf{v}^{(j)})^*$) and the facts that $\mathbf{u}^{(k+1)}$, $\mathbf{u}^{(k+2)}$, $\ldots$, $\mathbf{u}^{(l-1)}$, $\mathbf{u}^{(l)}$ are orthonormal, as are $\mathbf{v}^{(k+1)}$, $\mathbf{v}^{(k+2)}$, $\ldots$, $\mathbf{v}^{(l-1)}$, $\mathbf{v}^{(l)}$, yields that $\|\mathbf{A} - \mathbf{B}\| = \sigma_{k+1}$. We now consider any arbitrary $m \times n$ matrix $\mathbf{B}$ whose rank is at most k, and complete the proof by showing that $\|\mathbf{A} - \mathbf{B}\| \geq \sigma_{k+1}$.

It follows from the fact that the rank of $\mathbf{B}$ is at most k that the dimension of the null space of $\mathbf{B}$ is at least $n - k$. Therefore, there must exist a nonzero vector $\mathbf{w}$ that belongs both to the null space of $\mathbf{B}$ and to the $(k+1)$-dimensional space spanned by the orthonormal vectors $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, $\ldots$, $\mathbf{v}^{(k)}$, $\mathbf{v}^{(k+1)}$, for otherwise the second dimension n of $\mathbf{B}$ would be at least $(n - k) + (k + 1)$.

It follows from the definition of the spectral norm that

$$\|\mathbf{A} - \mathbf{B}\| \geq \frac{\|(\mathbf{A} - \mathbf{B})\,\mathbf{w}\|}{\|\mathbf{w}\|}. \tag{6.7}$$

It follows from the fact that $\mathbf{w}$ belongs to the null space of B that

$$\|(\mathbf{A} - \mathbf{B})\,\mathbf{w}\| = \|\mathbf{A}\mathbf{w}\|. \tag{6.8}$$

It follows from the fact that $\mathbf{w}$ belongs to the space spanned by the orthonormal vectors $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, $\ldots$, $\mathbf{v}^{(k)}$, $\mathbf{v}^{(k+1)}$ that

$$\mathbf{w} = \sum_{j=1}^{k+1} w_j\,\mathbf{v}^{(j)}, \tag{6.9}$$

where w_j is the inner product of $\mathbf{v}^{(j)}$ and $\mathbf{w}$: $w_j = (\mathbf{v}^{(j)})^*\,\mathbf{w}$. Combining (6.3) (that is, $\mathbf{A} = \sum_{j=1}^{l} \sigma_j\,\mathbf{u}^{(j)}\,(\mathbf{v}^{(j)})^*$), (6.9), and the fact that $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, $\ldots$, $\mathbf{v}^{(l-1)}$, $\mathbf{v}^{(l)}$ are orthonormal yields that

$$\mathbf{A}\mathbf{w} = \sum_{j=1}^{k+1} w_j\,\sigma_j\,\mathbf{u}^{(j)} \tag{6.10}$$

The fact that $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$, $\ldots$, $\mathbf{u}^{(k)}$, $\mathbf{u}^{(k+1)}$ are orthonormal yields that

$$\left\| \sum_{j=1}^{k+1} w_j\,\sigma_j\,\mathbf{u}^{(j)} \right\|^2 = \sum_{j=1}^{k+1} |w_j\,\sigma_j|^2. \tag{6.11}$$

It follows from the fact that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq \sigma_{k+1} \geq 0$ that

$$\sum_{j=1}^{k+1} |w_j\,\sigma_j|^2 \geq (\sigma_{k+1})^2 \sum_{j=1}^{k+1} |w_j|^2. \tag{6.12}$$

Combining (6.9), the fact that w_j is the inner product of $\mathbf{v}^{(j)}$ and $\mathbf{w}$, and the fact that $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, $\ldots$, $\mathbf{v}^{(k)}$, $\mathbf{v}^{(k+1)}$ are orthonormal yields that

$$\sum_{j=1}^{k+1} |w_j|^2 = \|\mathbf{w}\|^2. \tag{6.13}$$

Combining (6.7), (6.8), and (6.10)–(6.13) yields that

$$\|\mathbf{A} - \mathbf{B}\| \geq \sigma_{k+1}, \tag{6.14}$$

completing the proof. □

Remark 22 The first two paragraphs in the above proof of Theorem 21 explicitly construct a matrix $\mathbf{B}$ such that the rank of $\mathbf{B}$ is at most k, and $\|\mathbf{A} - \mathbf{B}\| = \sigma_{k+1}(\mathbf{A})$. This choice for $\mathbf{B}$ also minimizes the Frobenius/Hilbert-Schmidt norm of $\mathbf{A} - \mathbf{B}$ (the Frobenius/Hilbert-Schmidt norm of $\mathbf{A} - \mathbf{B}$ is the square root of the sum of the squares of the absolute values of the entries of $\mathbf{A} - \mathbf{B}$).

Chapter 7

The interpolative decomposition (ID)

In this chapter, we describe the interpolative decomposition (ID) for the low-rank approximation of matrices. While the singular value decomposition (SVD) described, for example, in the previous chapter is useful theoretically and provides a basis for understanding the ID, the ID has many practical advantages over the SVD for high-precision matrix compression. We will use the ID extensively in Chapter 8.

Theorem 25 below is the main existential result, asserting that, for any set of n continuous functions on a compact space S, there exist numerically stable n-point interpolation formulae. (The reader who is not familiar with the notion of a compact space is welcome to assume throughout this chapter that S contains only finitely many points, instead of just being compact, and that any function on S is continuous.) Remarks 29 and 30 describe algorithms associated with Theorems 25 and 28, and formally define the ID, too. First we will need the following lemma.

Lemma 23 *Suppose that l is a positive integer, S is an arbitrary set containing at least l points, and f_1, f_2, ..., f_{l-1}, f_l are complex-valued functions on S that are linearly independent.*

Then, there exist l points x_1, x_2, ..., x_{l-1}, x_l in S such that the vectors $\mathbf{u}(x_1)$, $\mathbf{u}(x_2)$, ..., $\mathbf{u}(x_{l-1})$, $\mathbf{u}(x_l)$ are linearly independent, where, for any $x \in S$, $\mathbf{u} = \mathbf{u}(x)$ is the $l \times 1$ column vector with the entry

$$u_k = f_k(x). \tag{7.1}$$

Proof. We apply Gaussian elimination with row pivoting (see, for example, Dahlquist and Björck [13]) to the set of all $1 \times l$ row vectors $\mathbf{u}(x)^{\mathrm{T}}$ for all $x \in S$, while ensuring that all pivot vectors are non-zero via appropriate row-pivoting. The desired points x_1, x_2, ..., x_{l-1}, x_l in S are those corresponding to the pivot rows (recall that the rows are indexed by the elements of S). □

Remark 24 All results of this chapter also apply in the real-valued case, provided that the word "complex" is replaced with "real" everywhere.

Theorem 25 *Suppose that S is a compact space, l and m are positive integers with $l \leq m$, and f_1, f_2, ..., f_{m-1}, f_m are continuous complex-valued functions on S, such that at most l of f_1, f_2, ..., f_{m-1}, f_m are linearly independent.*

Then, there exist l points $x_1, x_2, \ldots, x_{l-1}, x_l$ in S and l functions $g_1, g_2, \ldots, g_{l-1}, g_l$ on S such that

$$|g_i(x)| \leq 1 \tag{7.2}$$

for any $x \in S$, and $i = 1, 2, \ldots, l-1, l$, and

$$f(x) = \sum_{i=1}^{l} f(x_i)\, g_i(x) \tag{7.3}$$

for any $x \in S$ and any function f defined on S via the formula

$$f(x) = \sum_{j=1}^{m} c_j\, f_j(x) \tag{7.4}$$

for some complex numbers $c_1, c_2, \ldots, c_{m-1}, c_m$.

Proof. Without loss of generality, we assume that $f_1, f_2, \ldots, f_{m-1}, f_m$ are linearly independent, that is, that $l = m$.

We first choose the points $x_1, x_2, \ldots, x_{l-1}, x_l$ in S by maximizing the function

$$D(x_1, x_2, \ldots, x_{l-1}, x_l) = |\det \mathbf{A}(x_1, x_2, \ldots, x_{l-1}, x_l)|, \tag{7.5}$$

where $\mathbf{A} = \mathbf{A}(x_1, x_2, \ldots, x_{l-1}, x_l)$ is the $l \times l$ matrix with the entry

$$A_{j,k} = f_j(x_k) \tag{7.6}$$

for $j, k = 1, 2, \ldots, l-1, l$, and "det" takes the determinant. There exist l points $x_1, x_2, \ldots, x_{l-1}, x_l$ in S such that $D(x_1, x_2, \ldots, x_{l-1}, x_l)$ achieves its maximal value, since S is compact, and D is continuous (after all, $f_1, f_2, \ldots, f_{l-1}, f_l$ are continuous). Defining

$$B = \max_{x_1, x_2, \ldots, x_{l-1}, x_l \in S} D(x_1, x_2, \ldots, x_{l-1}, x_l), \tag{7.7}$$

we thus know that there are points $x_1, x_2, \ldots, x_{l-1}, x_l$ in S satisfying

$$D(x_1, x_2, \ldots, x_{l-1}, x_l) = B. \tag{7.8}$$

Moreover, it follows from Lemma 23 that B is strictly positive.

We next define the functions $g_1, g_2, \ldots, g_{l-1}, g_l$ on S via the formula

$$g_i(x) = \frac{\det \mathbf{A}(x_1, x_2, \ldots, x_{i-2}, x_{i-1}, \ \ x, \ \ x_{i+1}, x_{i+2}, \ldots, x_{l-1}, x_l)}{\det \mathbf{A}(x_1, x_2, \ldots, x_{l-1}, x_l)} \tag{7.9}$$

(here, the numerator is the same as the denominator, but with x in place of x_i). We then obtain (7.3) by combining (7.9) and the Cramer rule applied to the linear system

$$\mathbf{A}\,\mathbf{v} = \mathbf{u}, \tag{7.10}$$

where $\mathbf{A} = \mathbf{A}(x_1, x_2, \ldots, x_{l-1}, x_l)$ is defined in (7.6), $\mathbf{v} = \mathbf{v}(x)$ is the $l \times 1$ column vector with the entry

$$v_i = g_i(x) \tag{7.11}$$

for $i = 1, 2, \ldots, l-1, l$, and $\mathbf{u} = \mathbf{u}(x)$ is the $l \times 1$ column vector with the entry

$$u_i = f_i(x) \tag{7.12}$$

for $i = 1, 2, \ldots, l-1, l$.

It follows from (7.7) that

$$D(x_1, x_2, \ldots, x_{i-2}, x_{i-1}, \ \ x, \ \ x_{i+1}, x_{i+2}, \ldots, x_{l-1}, x_l) \le B \tag{7.13}$$

for any $x \in S$. Combining (7.9), (7.5), (7.8), and (7.13) yields (7.2). □

Remark 26 Due to (7.2), the interpolation formula (7.3) is numerically stable.

Remark 27 The suppositions of Theorem 25 that S is a compact space and that $f_1, f_2, \ldots, f_{m-1}, f_m$ are continuous are not necessary in order to obtain a guarantee similar to (7.2). See Martinsson, Rokhlin, and Tygert [40] for a version of Theorem 25 that supposes only that S is an arbitrary set and that $f_1, f_2, \ldots, f_{m-1}, f_m$ are bounded.

Applied in the linear-algebraic setting, Theorem 25 above yields the following theorem.

Theorem 28 *Suppose that l, m, and n are positive integers with $l \le m$ and $l \le n$, and $\mathbf{A}$ is an $m \times n$ matrix, such that the rank of $\mathbf{A}$ is at most l.*

Then, there exist an $m \times l$ matrix $\mathbf{B}$ whose columns constitute a subset of the columns of $\mathbf{A}$, and an $l \times n$ matrix $\mathbf{P}$, such that

$$\mathbf{A}_{m \times n} = \mathbf{B}_{m \times l} \, \mathbf{P}_{l \times n} \tag{7.14}$$

and

$$|P_{i,k}| \le 1 \tag{7.15}$$

for $i = 1, 2, \ldots, l-1, l$, and $k = 1, 2, \ldots, n-1, n$.

Proof. By taking S to be a set consisting of n points, say $y_1, y_2, \ldots, y_{n-1}, y_n$, in Theorem 25, we obtain from (7.3) that

$$f_j(y_k) = \sum_{i=1}^{l} f_j(x_i) \, g_i(y_k) \tag{7.16}$$

for $j = 1, 2, \ldots, m-1, m$ and $k = 1, 2, \ldots, n-1, n$. Moreover, (7.16) yields (7.14), and (7.2) yields (7.15), provided that

$$A_{j,k} = f_j(y_k) \tag{7.17}$$

for $j = 1, 2, \ldots, m-1, m$ and $k = 1, 2, \ldots, n-1, n$,

$$B_{j,i} = f_j(x_i) \tag{7.18}$$

for $j = 1, 2, \ldots, m-1, m$ and $i = 1, 2, \ldots, l-1, l$, and

$$P_{i,k} = g_i(y_k) \tag{7.19}$$

for $i = 1, 2, \ldots, l-1, l$ and $k = 1, 2, \ldots, n-1, n$. □

Remark 29 An interpolative decomposition (ID) of a matrix $\mathbf{A}$ consists of matrices $\mathbf{B}$ and $\mathbf{P}$ such that (7.14) holds, the columns of $\mathbf{B}$ are also columns of $\mathbf{A}$, and $\mathbf{P}$ is not too large; $\mathbf{P}$ is the "interpolation matrix." To compute an ID of $\mathbf{A}$, we may start with its pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition

$$\mathbf{A}_{m \times n} = \mathbf{Q}_{m \times k}\, \mathbf{R}_{k \times n}\, \mathbf{\Pi}_{n \times n}, \tag{7.20}$$

where $\mathbf{Q}$ is a complex $m \times k$ matrix whose columns are orthonormal, $\mathbf{R}$ is a complex upper-triangular (meaning upper-trapezoidal) $k \times n$ matrix, and $\mathbf{\Pi}$ is a real $n \times n$ permutation matrix. For details on the computation of the pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition in (7.20), see, for example, Chapter 5 of Golub and Van Loan [22]. Defining $\mathbf{S}$ to be the leftmost $k \times k$ block of $\mathbf{R}$, and $\mathbf{T}$ to be the rightmost $k \times (n - k)$ block of $\mathbf{R}$, so that

$$\mathbf{R}_{k \times n} = \left(\begin{array}{c|c} \mathbf{S}_{k \times k} & \mathbf{T}_{k \times (n-k)} \end{array} \right), \tag{7.21}$$

we obtain from (7.20) and (7.21) that

$$\mathbf{A}_{m \times n} = \mathbf{B}_{m \times k}\, \mathbf{P}_{k \times n}, \tag{7.22}$$

where

$$\mathbf{B}_{m \times k} = \mathbf{Q}_{m \times k}\, \mathbf{S}_{k \times k} \tag{7.23}$$

and

$$\mathbf{P}_{k \times n} = \left(\begin{array}{c|c} \mathbf{1}_{k \times k} & (\mathbf{S}^{-1})_{k \times k}\, \mathbf{T}_{k \times (n-k)} \end{array} \right) \mathbf{\Pi}_{n \times n}. \tag{7.24}$$

Combining (7.22) and (7.24) yields that $\mathbf{B}_{m \times k}$ is the leftmost $m \times k$ block of $\mathbf{A}_{m \times n}\, (\mathbf{\Pi}^{-1})_{n \times n}$. Therefore, the columns of $\mathbf{B}$ constitute a subset of the columns of $\mathbf{A}$.

In practice, we generally find that the entries of $\mathbf{P}$ are not too large. To guarantee that the absolute values of the entries of $\mathbf{P}$ are at most β, for some real number $\beta > 1$, we could use the algorithm of Gu and Eisenstat [25], which in the worst case requires about $\log_\beta(n)$ times more flops than classical pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition algorithms.

Thus, we may form the ID (7.22) of a matrix $\mathbf{A}$ by forming its pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition (7.20) and then constructing $\mathbf{B}$ and $\mathbf{P}$ via (7.23) and (7.24), commensurate with the partitioning of $\mathbf{R}$ in (7.21). Of course, the columns of $\mathbf{B}$ are just the columns of $\mathbf{A}$ corresponding to the pivots used in forming the pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition (7.20). Also, we do not need to form $\mathbf{S}^{-1}$ in (7.24) explicitly; we need only apply $\mathbf{S}^{-1}$ to each of the $n - k$ columns of $\mathbf{T}$, that is, to solve $n - k$ systems of linear-algebraic equations involving the triangular matrix $\mathbf{S}$.

Remark 30 Theorem 2 of Section 3 in Martinsson, Rokhlin, and Tygert [40] and Theorem 3 in Cheng, Gimbutas, Martinsson, and Rokhlin [9] state the following generalization of Theorem 28 (see also the original sources, such as Goreinov and Tyrtyshnikov [23]):

Suppose that m and n are positive integers, and $\mathbf{A}$ is a complex $m \times n$ matrix. Then, for any positive integer k with $k \leq m$ and $k \leq n$, there exist a complex $k \times n$ matrix $\mathbf{P}$, and a complex $m \times k$ matrix $\mathbf{B}$ whose columns constitute a subset of the columns of $\mathbf{A}$, such that

1. some subset of the columns of $\mathbf{P}$ makes up the $k \times k$ identity matrix,

2. no entry of $\mathbf{P}$ has an absolute value greater than 1,

3. $\|\mathbf{P}\| \leq \sqrt{k\,(n-k)+1}$ (where $\|\mathbf{P}\|$ is the spectral norm of $\mathbf{P}$),

4. the least (*i.e.*, the k^{th} greatest) singular value of $\mathbf{P}$ is at least 1,

5. $\mathbf{B}\,\mathbf{P} = \mathbf{A}$ when $k = m$ or $k = n$, and

6. $\|\mathbf{B}\,\mathbf{P} - \mathbf{A}\| \leq \sqrt{k\,(n-k)+1}\;\sigma_{k+1}$ when $k < m$ and $k < n$, where σ_{k+1} is the $(k+1)^{\text{st}}$ greatest singular value of $\mathbf{A}$, and $\|\mathbf{B}\,\mathbf{P} - \mathbf{A}\|$ is the spectral norm of $\mathbf{B}\,\mathbf{P} - \mathbf{A}$.

Of course, property 3 follows immediately from properties 1 and 2, and property 4 follows immediately from property 1. Properties 1–4 guarantee the numerical stability of the ID. Property 6 states that the rank-k approximation provided by the ID is accurate to within a factor of $\sqrt{k\,(n-k)+1}$ times the best possible (see Theorem 21 in Chapter 6).

While existing algorithms for computing $\mathbf{B}$ and $\mathbf{P}$ satisfying properties 1–6 above are computationally expensive (see, for example, Gu and Eisenstat [25]), given a real number $\beta > 1$, the algorithm of Gu and Eisenstat [25] produces $\mathbf{B}$ and $\mathbf{P}$ such that

1. some subset of the columns of $\mathbf{P}$ makes up the $k \times k$ identity matrix,

2. no entry of $\mathbf{P}$ has an absolute value greater than β,

3. $\|\mathbf{P}\| \leq \sqrt{\beta^2\,k\,(n-k)+1}$ (where $\|\mathbf{P}\|$ is the spectral norm of $\mathbf{P}$),

4. the least (*i.e.*, the k^{th} greatest) singular value of $\mathbf{P}$ is at least 1,

5. $\mathbf{B}\,\mathbf{P} = \mathbf{A}$ when $k = m$ or $k = n$, and

6. $\|\mathbf{B}\,\mathbf{P} - \mathbf{A}\| \leq \sqrt{\beta^2\,k\,(n-k)+1}\;\sigma_{k+1}$ when $k < m$ and $k < n$, where σ_{k+1} is the $(k+1)^{\text{st}}$ greatest singular value of $\mathbf{A}$, and $\|\mathbf{B}\,\mathbf{P} - \mathbf{A}\|$ is the spectral norm of $\mathbf{B}\,\mathbf{P} - \mathbf{A}$.

In the worst case, the algorithm of Gu and Eisenstat [25] requires about $\log_\beta(n)$ times more flops than the classical pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition algorithms.

Conveniently, a simple modification of the easily implemented algorithm described in the preceding remark generally produces results satisfying properties 1–6 above, with a reasonably small β. The modified algorithm is the same, except that (7.20) becomes

$$\|\mathbf{A}_{m\times n} - \mathbf{Q}_{m\times k}\,\mathbf{R}_{k\times n}\,\mathbf{\Pi}_{n\times n}\| \leq \sqrt{\beta^2\,k\,(n-k)+1}\;\sigma_{k+1}, \tag{7.25}$$

where σ_{k+1} is the $(k+1)^{\text{st}}$ greatest singular value of $\mathbf{A}$, and $\|\cdot\|$ is the spectral norm. Usually we have a desired precision, say ε, and instead of (7.20) we form a pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition $\mathbf{Q}\,\mathbf{R}\,\mathbf{\Pi}$ such that $\|\mathbf{A} - \mathbf{Q}\,\mathbf{R}\,\mathbf{\Pi}\| \leq \varepsilon$, by choosing the rank k of the approximation $\mathbf{Q}\,\mathbf{R}\,\mathbf{\Pi}$ to $\mathbf{A}$ appropriately. We then construct the matrices $\mathbf{B}$ and $\mathbf{P}$ according to (7.23) and (7.24), commensurate with the partitioning of $\mathbf{R}$ in (7.21), obtaining in place of (7.22) the approximation

$$\|\mathbf{A}_{m\times n} - \mathbf{B}_{m\times k}\,\mathbf{P}_{k\times n}\| \leq \varepsilon, \tag{7.26}$$

where ε is the spectral-norm accuracy of the pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition $\mathbf{Q}\,\mathbf{R}\,\mathbf{\Pi}$ approximating $\mathbf{A}$. Again we can take the columns of $\mathbf{B}$ to be the columns of $\mathbf{A}$ corresponding to the pivots used in forming the pivoted "$\mathbf{Q}\,\mathbf{R}$" decomposition. Moreover, we do not need to form $\mathbf{S}^{-1}$ in (7.24) explicitly; we need only apply $\mathbf{S}^{-1}$ to each of the $n-k$ columns of $\mathbf{T}$, that is, to solve $n-k$ systems of linear-algebraic equations involving the triangular matrix $\mathbf{S}$.

Chapter 8

Fast methods for applying nonoscillatory integral operators

In this chapter, we describe methods for rapidly applying certain special types of matrices to arbitrary vectors, including many of the matrices that can be used in GMRES of Chapter 4 to solve discretized integral equations such as (3.6). The algorithms we describe are known colloquially as "fast multipole methods" and their brethren. For simplicity and maximal efficiency, we will be using the interpolative decomposition (ID) described in the previous chapter, though similar techniques (such as the SVD) can work reasonably well.

8.1 Multipoles

Multipoles are expansions of functions (identical to Taylor expansions for functions of a single complex variable) which permit the rapid application of certain integral operators to arbitrary vectors. The fast computations of the present chapter require only the existence of multipole expansions (or of analogous low-rank structure described later). In the present section, we prove the existence, in Lemma 32.

Lemma 32 represents as a linear combination of $2m+1$ functions g_0, g_1, g_2, $\ldots$, g_{m-1}, g_m, and $\tilde{g}_1$, $\tilde{g}_2$, $\ldots$, $\tilde{g}_{m-1}$, $\tilde{g}_m$ (known as "multipoles") the action of the kernel $G_0(\mathbf{x}, \mathbf{y}) = \ln|\mathbf{x}-\mathbf{y}|$, as acting from an arbitrary finite number of points in $\mathbb{R}^2$ to any point separated by a factor S from a disc containing the source points. Figure 8.1 illustrates the geometry. We will be identifying points in $\mathbb{R}^2$ with numbers in the complex plane, and vice versa, in the standard fashion. For convenience, we first prove Lemma 32 for a single source point (*i.e.*, with $n = 1$ in Lemma 32), as the following lemma.

Lemma 31 *Suppose that R and S are real numbers, and ρ, z_0, and z_1 are complex numbers, such that $R > 0$, $S > 1$, and*

$$|z_1 - z_0| \leq R. \tag{8.1}$$

Then,

$$|\Phi(z) - \Phi_m(z)| \leq \frac{T}{(S-1)\,S^m} \tag{8.2}$$

for $m = 1,\ 2,\ 3,\ \ldots$ and any $z \in \mathbb{C}$ with

$$|z - z_0| \geq S \cdot R, \tag{8.3}$$

where

$$\Phi(z) = \rho \, \ln|z - z_1|, \tag{8.4}$$

$$\Phi_m(z) = c_0 \, g_0(z) + \sum_{j=1}^{m} c_j \, g_j(z) + \sum_{j=1}^{m} \tilde{c}_j \, \tilde{g}_j(z), \tag{8.5}$$

$$c_0 = f_0 \, \rho, \tag{8.6}$$

$$f_0 = 1, \tag{8.7}$$

$$c_j = f_j \, \rho, \tag{8.8}$$

$$f_j = -\frac{1}{2j} \left(\frac{z_1 - z_0}{R} \right)^j, \tag{8.9}$$

$$\tilde{c}_j = \tilde{f}_j \, \rho, \tag{8.10}$$

$$\tilde{f}_j = -\frac{1}{2j} \left(\frac{\overline{z_1 - z_0}}{R} \right)^j, \tag{8.11}$$

$$g_0(z) = \ln|z - z_0|, \tag{8.12}$$

$$g_j(z) = \left(\frac{R}{z - z_0} \right)^j, \tag{8.13}$$

$$\tilde{g}_j(z) = \left(\frac{R}{\overline{z - z_0}} \right)^j, \tag{8.14}$$

and

$$T = |\rho|. \tag{8.15}$$

Proof. We define w via the formula

$$w = \frac{z_1 - z_0}{z - z_0} \tag{8.16}$$

and observe that

$$\ln|z - z_1| = \ln|z - z_0| + \ln|1 - w| \tag{8.17}$$

and that, combining (8.1) and (8.3),

$$|w| \leq \frac{1}{S}. \tag{8.18}$$

Since $S > 1$, (8.18) yields that

$$\ln(1 - w) = \sum_{j=1}^{\infty} \frac{-w^j}{j}. \tag{8.19}$$

From (8.19) and the fact that the real part of $\ln(1-w)$ is $\ln|1-w|$, we get that

$$\left| \ln|1-w| - \left(\sum_{j=1}^{m} \frac{-w^j}{2j} + \sum_{j=1}^{m} \frac{-\overline{w}^j}{2j} \right) \right| \le \sum_{j=m+1}^{\infty} \frac{|w|^j}{j}. \tag{8.20}$$

The right-hand side of (8.20) has the bound

$$\sum_{j=m+1}^{\infty} \frac{|w|^j}{j} \le \sum_{j=m+1}^{\infty} |w|^j = \frac{|w|^{m+1}}{1-|w|}. \tag{8.21}$$

Combining (8.17), (8.20), (8.21), and (8.18) immediately yields (8.2). □

The following lemma represents as a linear combination of $2m+1$ functions g_0, g_1, g_2, ..., g_{m-1}, g_m, and $\tilde{g}_1$, $\tilde{g}_2$, ..., $\tilde{g}_{m-1}$, $\tilde{g}_m$ (known as "multipoles") the action of the kernel $G_0(\mathbf{x},\mathbf{y}) = \ln|\mathbf{x}-\mathbf{y}|$, as acting from an arbitrary finite number of points in $\mathbb{R}^2$ to any point separated by a factor S from a disc containing the source points; the lemma is a slight modification of a lemma from Greengard and Rokhlin [24]. Figure 8.1 illustrates the geometry. We will be identifying points in $\mathbb{R}^2$ with numbers in the complex plane, and vice versa, in the standard fashion. The expansion with $m = 0$ is often called the "center-of-mass" approximation, and is obviously a very efficient (though fairly inaccurate) representation. Higher-order expansions (with $m > 0$) can be both efficient and accurate. Notice that m does not have to be very large to ensure high accuracy.

Lemma 32 *Suppose that n is a positive integer, R and S are real numbers, and ρ_1, ρ_2, ..., ρ_{n-1}, ρ_n and z_0, z_1, z_2, ..., z_{n-1}, z_n are complex numbers, such that $R > 0$, $S > 1$, and*

$$|z_k - z_0| \le R \tag{8.22}$$

for $k = 1, 2, \ldots, n-1, n$.

Then,

$$|\Phi(z) - \Phi_m(z)| \le \frac{T}{(S-1)\,S^m} \tag{8.23}$$

for $m = 1, 2, 3, \ldots$ and any $z \in \mathbb{C}$ with

$$|z - z_0| \ge S \cdot R, \tag{8.24}$$

where

$$\Phi(z) = \sum_{k=1}^{n} \rho_k \, \ln|z - z_k|, \tag{8.25}$$

$$\Phi_m(z) = c_0\, g_0(z) + \sum_{j=1}^{m} c_j\, g_j(z) + \sum_{j=1}^{m} \tilde{c}_j\, \tilde{g}_j(z), \tag{8.26}$$

$$c_0 = \sum_{k=1}^{n} f_{0,k}\, \rho_k, \tag{8.27}$$

$$f_{0,k} = 1, \tag{8.28}$$

$$c_j = \sum_{k=1}^{n} f_{j,k}\, \rho_k, \tag{8.29}$$

$$f_{j,k} = -\frac{1}{2j} \left(\frac{z_k - z_0}{R} \right)^j, \tag{8.30}$$

$$\tilde{c}_j = \sum_{k=1}^{n} \tilde{f}_{j,k}\, \rho_k, \tag{8.31}$$

$$\tilde{f}_{j,k} = -\frac{1}{2j} \left(\frac{\overline{z_k - z_0}}{R} \right)^j, \tag{8.32}$$

$$g_0(z) = \ln |z - z_0|, \tag{8.33}$$

$$g_j(z) = \left(\frac{R}{z - z_0} \right)^j, \tag{8.34}$$

$$\tilde{g}_j(z) = \left(\frac{R}{\overline{z - z_0}} \right)^j, \tag{8.35}$$

and

$$T = \sum_{k=1}^{n} |\rho_k|. \tag{8.36}$$

Proof. This lemma follows immediately from Lemma 31. □

Remark 33 The representation (8.26) is in fact numerically stable, since (8.30), (8.32), (8.34), and (8.35) are at most 1 in magnitude, for any $z \in \mathbb{C}$, $j = 1, 2, \ldots, m-1, m$, and $k = 1, 2, \ldots, n-1, n$.

8.2 Linear-algebraic formulation

The following lemma interprets Lemma 32 in linear-algebraic terms. As Remark 35 below discusses, these linear-algebraic properties enable fast computations. The lemma concerns the action $\mathbf{A}$ of the kernel $G_0(\mathbf{x}, \mathbf{y}) = \ln |\mathbf{x} - \mathbf{y}|$, as acting from n points in a disc of radius R in $\mathbb{R}^2$, to l points outside the concentric disc of radius $2R$. The lemma states that the accuracy $\sigma_{2m+2}(\mathbf{A}_{l \times n})$ of the best rank-$(2m+1)$ approximation to the matrix $\mathbf{A}$ improves exponentially fast with increasing m, independent of the dimensions l and n of $\mathbf{A}$. Again, we will be identifying points in $\mathbb{R}^2$ and numbers in the complex plane.

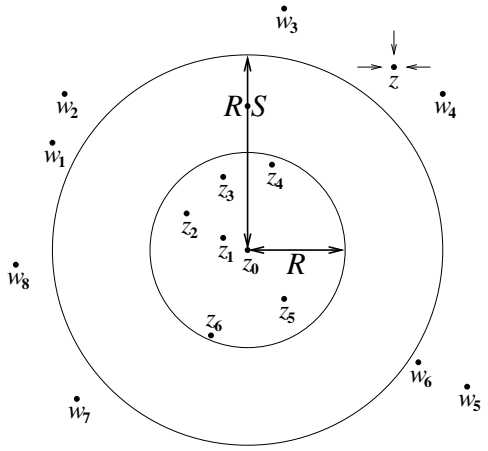

Figure 8.1: The points z_1, z_2, ..., z_{n-1}, z_n are all closer than R units to z_0. The points z and w_1, w_2, ..., w_{l-1}, w_l are all farther than $R \cdot S$ units from z_0.

Lemma 34 *Suppose that l and n are positive integers, R is a positive real number, and z_0, z_1, z_2, ..., z_{n-1}, z_n and w_1, w_2, ..., w_{l-1}, w_l are complex numbers, such that*

$$|z_k - z_0| \le R \tag{8.37}$$

for $k = 1, 2, \ldots, n-1, n$, and

$$|w_i - z_0| \ge 2R \tag{8.38}$$

for $i = 1, 2, \ldots, l-1, l$.

Then, the $(2m+2)^{nd}$ greatest singular value $\sigma_{2m+2}(\mathbf{A})$ of $\mathbf{A}$ satisfies

$$\sigma_{2m+2}(\mathbf{A}_{l \times n}) \le \frac{\sqrt{nl}}{2^m} \tag{8.39}$$

for any positive integer m with $2m+1 < l$ and $2m+1 < n$, where $\mathbf{A}$ is the $l \times n$ matrix whose entries are

$$A_{i,k} = \ln |w_i - z_k| \tag{8.40}$$

for $i = 1, 2, \ldots, l-1, l$ and $k = 1, 2, \ldots, n-1, n$.

Proof. We will construct a rank-$(2m+1)$ approximation $\mathbf{G}_{l \times (2m+1)} \cdot \mathbf{F}_{(2m+1) \times n}$ to $\mathbf{A}_{l \times n}$ such that

$$\|\mathbf{A}_{l \times n} - \mathbf{G}_{l \times (2m+1)} \cdot \mathbf{F}_{(2m+1) \times n}\| \le \frac{\sqrt{nl}}{2^m}, \tag{8.41}$$

where $\|\mathbf{A} - \mathbf{G}\,\mathbf{F}\|$ is the spectral norm of $\mathbf{A} - \mathbf{G}\,\mathbf{F}$; (8.39) follows immediately from (8.41), due to (6.5). We define $\mathbf{F}$ to be the $(2m+1) \times n$ matrix whose entries are

$$F_{j,k} = \begin{cases} 1, & j = 1 \\ -\frac{1}{2j-2} \left(\frac{z_k - z_0}{R}\right)^{j-1}, & 2 \le j \le m+1 \\ -\frac{1}{2j-2m-2} \left(\frac{\overline{z_k - z_0}}{R}\right)^{j-m-1}, & m+2 \le j \le 2m+1 \end{cases} \tag{8.42}$$

for $j = 1, 2, \ldots, 2m, 2m+1$ and $k = 1, 2, \ldots, n-1, n$. We define $\mathbf{G}$ to be the $l \times (2m+1)$ matrix whose entries are

$$G_{i,j} = \begin{cases} \ln |w_i - z_0|, & j = 1 \\ \left(\frac{R}{w_i - z_0}\right)^{j-1}, & 2 \le j \le m+1 \\ \left(\frac{R}{w_i - z_0}\right)^{j-m-1}, & m+2 \le j \le 2m+1 \end{cases} \tag{8.43}$$

for $i = 1, 2, \ldots, l-1, l$ and $j = 1, 2, \ldots, 2m, 2m+1$. With $S = 2$, (8.23) yields that

$$\left| \sum_{k=1}^{n} \left(A_{i,k} - \sum_{j=1}^{2m+1} G_{i,j} \, F_{j,k} \right) \rho_k \right| \le \frac{1}{2^m} \sum_{k=1}^{n} |\rho_k| \tag{8.44}$$

for $i = 1, 2, \ldots, l-1, l$ and any complex numbers $\rho_1, \rho_2, \ldots, \rho_{n-1}, \rho_n$. Combining (8.44) and the Cauchy-Schwarz inequality yields that

$$\sum_{i=1}^{l} \left| \sum_{k=1}^{n} \left(A_{i,k} - \sum_{j=1}^{2m+1} G_{i,j} \, F_{j,k} \right) \rho_k \right|^2 \le \frac{nl}{2^{2m}} \sum_{k=1}^{n} |\rho_k|^2 \tag{8.45}$$

for any complex numbers $\rho_1, \rho_2, \ldots, \rho_{n-1}, \rho_n$. Combining (8.45) and the definition of the spectral norm yields (8.41), proving (8.39). □

Remark 35 Because of (8.39), we may efficiently compute to high precision the result of applying the matrix $\mathbf{A}$ defined in (8.40) to any arbitrary vector, assuming that the sources $z_1, z_2, \ldots, z_{n-1}, z_n$ and targets $w_1, w_2, \ldots, w_{l-1}, w_l$ are well-separated, as depicted in Figure 8.1. Indeed, due to (8.39) and (6.5), we may replace $\mathbf{A}$ to high precision with a rank-$(2m+1)$ matrix, decomposing $\mathbf{A}_{l \times n} \approx \mathbf{G}_{l \times (2m+1)} \cdot \mathbf{F}_{(2m+1) \times n}$, and then applying $\mathbf{F}$ first to any vector and then $\mathbf{G}$ to the result, instead of applying $\mathbf{A}$ directly. Similarly, we may efficiently apply $\mathbf{A}^{\mathrm{T}}$ to an arbitrary vector to high precision, in effect swapping the sources and the targets. Notice that m does not have to be very large to ensure high accuracy.

8.3 Multilevel compression

Suppose that n is a positive integer power of 2, and $z_1 = 1, z_2 = 2, \ldots, z_{n-1} = n-1, z_n = n$ are n points in the interval $[0, n]$. Suppose also that $\mathbf{A}$ is the $n \times n$ matrix whose entries are

$$A_{j,k} = \ln |z_j - z_k| \tag{8.46}$$

for $j = 1, 2, \ldots n-1, n$ and $k = 1, 2, \ldots, n-1, n$ when $j \neq k$. In this section, we describe a scheme for applying any such $\mathbf{A}$ rapidly and with high precision to arbitrary vectors. In subsequent sections, we will accelerate this scheme even further. The scheme generalizes essentially unchanged to arbitrary points $z_1, z_2, \ldots, z_{n-1}, z_n$ in the complex plane. As we will see, the scheme also generalizes to many other matrices associated with points in one, two, or three dimensions.

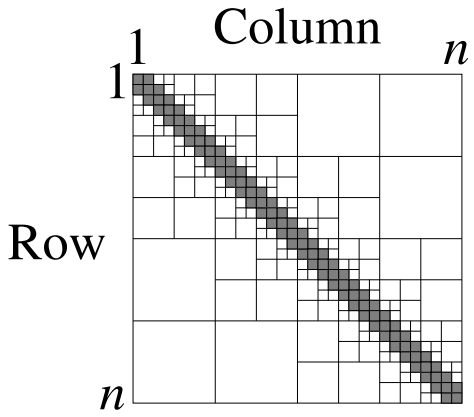

Figure 8.2: The undarkened blocks in this decomposition of an $n \times n$ matrix are separated from the diagonal. The darkened blocks are not separated from the diagonal, but there are only $3n - 2$ of them.

Consider any undarkened block $\mathbf{B}$ in the decomposition of $\mathbf{A}$ depicted in Figure 8.2. We denote by I the set of column indices associated with this block $\mathbf{B}$ of $\mathbf{A}$; we denote by H the set of row indices associated with the block. It is easy to see that $\min_{j \in H} |z_j - z_0| > 2 \max_{k \in I} |z_k - z_0|$, where z_0 is the point minimizing $\max_{k \in I} |z_k - z_0|$; that is, the points associated with the rows of $\mathbf{B}$ are well-separated from the points associated with the columns of $\mathbf{B}$, as required in Lemma 34. We thus obtain a rank-$(2m+1)$ approximation, say $\tilde{\mathbf{B}}$, to $\mathbf{B}$ via (8.39) and (6.5), such that $\|\mathbf{B} - \tilde{\mathbf{B}}\| \leq l/2^m$, where l is the dimension of $\mathbf{B}$. This low-rank approximation allows us to compute efficiently to high precision the result of applying $\mathbf{B}$ to any arbitrary vector (see Remark 35). We define C_m to be the number of floating-point operations (flops) required to apply $\mathbf{B}$, divided by the dimension of $\mathbf{B}$, so that the number of flops required for applying $\mathbf{B}$ thus is C_m times the dimension of $\mathbf{B}$. Of course, $C_m = \mathcal{O}(m)$.

Using these rank-$(2m + 1)$ representations of the undarkened blocks in Figure 8.2, we can apply all these blocks using less than $3C_m n \log_2(n) = \mathcal{O}(mn \log(n))$ flops (as seen by summing the costs in the rightmost column of Table 8.1). To apply the original matrix $\mathbf{A}$ to a vector, we must add in the contributions from the darkened entries in Figure 8.2, which requires $\mathcal{O}(n)$ flops (since there are $3n - 2$ darkened entries). In all, applying the original matrix $\mathbf{A}$ to high precision using the low-rank representations requires $\mathcal{O}(mn \log(n))$ flops. The Euclidean norm of the difference between the actual result and the approximation obtained via low-rank approximations is less than $3n \log_2(n)/2^m$, relative to the Euclidean norm of the vector to which we are applying $\mathbf{A}$. Thus, the accuracy improves exponentially fast as m increases; m need not be very large to produce accuracy close to the machine precision, even though applying the matrix $\mathbf{A}$ via this method requires only $\mathcal{O}(mn \log(n))$ flops.

We see, then, that the key property leading to fast algorithms is that there exist low-rank matrices approximating the undarkened blocks in Figure 8.2 to high accuracy. The ranks of these matrices ideally should be bounded by a small constant (preferably depending only weakly on the accuracy of the approximations).

Dimension	Number	Cost of applying to vectors (C_m· Number · Dimension)
$n/4$	$3 \cdot (4-2)$	$C_m \cdot 3 \cdot (4-2) \cdot n/4$
$n/8$	$3 \cdot (8-2)$	$C_m \cdot 3 \cdot (8-2) \cdot n/8$
$n/16$	$3 \cdot (16-2)$	$C_m \cdot 3 \cdot (16-2) \cdot n/16$
$\vdots$	$\vdots$	$\vdots$
4	$3 \cdot (n/4-2)$	$C_m \cdot 3 \cdot (n/4-2) \cdot 4$
2	$3 \cdot (n/2-2)$	$C_m \cdot 3 \cdot (n/2-2) \cdot 2$
1	$3 \cdot (n-2)$	$C_m \cdot 3 \cdot (n-2) \cdot 1$

Table 8.1: Undarkened blocks in Figure 8.2

8.4 Hierarchical construction

An especially efficient construction of the low-rank representations for the undarkened blocks in Figure 8.2 proceeds hierarchically. For some small positive integer l, and for each collection of l columns, say columns $jl+1$ through $jl+l$ for some nonnegative integer j, we construct an interpolative decomposition (ID) for the matrix obtained by deleting the diagonal $l \times l$ block and the $l \times l$ blocks neighboring the diagonal block above and below it. In Figure 8.3, two such collections are labeled "a" and "b."

For each collection of $2l$ columns, say columns $2jl+1$ through $2jl+2l$ for some nonnegative integer j, we could similarly construct an ID for the matrix obtained by deleting the diagonal $2l \times 2l$ block and the $2l \times 2l$ blocks neighboring the diagonal block above and below it; in Figure 8.3, two such collections are labeled "c" and "d." However, constructing the IDs directly is less efficient than recycling the IDs for the collections of l columns: Let us focus on the collection labeled "c" in Figure 8.3. As described in the previous paragraph, we have already constructed IDs for the collections of size l that overlap with "c" — there are two such collections, just like "a" and "b," but overlapping with "c." The IDs for these two narrower collections consist of selected columns, along with interpolation matrices. To high precision, the columns of the matrix for "c" are linear combinations of these selected columns, with the coefficients in the linear combinations given by the entries in the interpolation matrices (technically, this reconstruction is valid only in the parts of the columns in the region "c," not near the diagonal). We gather all these selected columns together into a matrix, delete the entries from the $2l \times 2l$ diagonal block and its upper and lower $2l \times 2l$ neighbors, and form an ID approximating the resulting matrix to high precision. This new ID selects a subset S of columns from the collection of previously selected columns; the previously selected columns are linear combinations of the subset, with the coefficients in the linear combinations given by the entries in the interpolation matrix (again, this reconstruction is valid only in the parts of the columns in the region "c," not near the diagonal). Since we can already interpolate from the previously selected columns to all columns in "c," we can now interpolate from S to all columns in "c," by first interpolating to the previously selected columns, and then interpolating from the previously selected columns to the rest.

Needless to say, we process in the same way all other collections of $2l$ columns, columns $2jl+1$ through $2jl+2l$ for some nonnegative integer j. We then proceed to collections

of $4l$ columns (including those for "e" and "f" in Figure 8.3), and then to collections of $8l$ columns, and so on. We can process the rows similarly.

8.5 Hierarchical application

Using the hierarchical construction of the preceding section, we can accelerate the algorithm of Section 8.3, eliminating the logarithmic factor from its operation count.

Consider any undarkened block $\mathbf{B}$ from the matrix $\mathbf{A}$ depicted in Figure 8.2; $\mathbf{B}$ represents $\mathbf{A}$ restricted to mapping from indices in an interval S to indices in an interval T. Suppose that $\rule{1em}{0.4pt}_S$ is the interpolation matrix from the ID, constructed in Section 8.4, for the columns in $\mathbf{A}$ corresponding to S. Suppose also that $[\!]_T$ is the transpose of the interpolation matrix from the ID, constructed in Section 8.4, for the transposes of the rows in $\mathbf{A}$ corresponding to T. If we then denote by $\mathbf{A}_{T \leftarrow S}$ the matrix formed from $\mathbf{A}$ by retaining only those entries that are in both one of the columns selected for the ID and one of the rows selected for the (other) ID, then $[\!]_T \cdot \mathbf{A}_{T \leftarrow S} \cdot \rule{1em}{0.4pt}_S$ is an accurate approximation to the block $\mathbf{B}$.

Suppose further that $\mathbf{B}$ is larger than the smallest blocks in Figure 8.2. We then partition $S = S_1 \cup S_2$ into the union of its two halves, S_1 and S_2, and do the same with $T = T_1 \cup T_2$. The hierarchical construction of the previous section provides a slightly short and fat matrix $\mathbf{E}_S$ and a slightly tall and skinny matrix $\mathbf{P}_T$ such that

$$\rule{1em}{0.4pt}_S \approx \mathbf{E}_S \cdot \left(\begin{array}{c|c} \rule{1em}{0.4pt}_{S_1} & \mathbf{0} \\ \hline \mathbf{0} & \rule{1em}{0.4pt}_{S_2} \end{array} \right), \tag{8.47}$$

$$[\!]_T \approx \left(\begin{array}{c|c} [\!]_{T_1} & \mathbf{0} \\ \hline \mathbf{0} & [\!]_{T_2} \end{array} \right) \cdot \mathbf{P}_T. \tag{8.48}$$

In order to apply the block $\mathbf{B}$ to a vector $\mathbf{v}_S$, we use the representation $\mathbf{B} \approx [\!]_T \cdot \mathbf{A}_{T \leftarrow S} \cdot \rule{1em}{0.4pt}_S$ and partition $\mathbf{v}_S$:

$$\mathbf{v}_S = \left(\begin{array}{c} \mathbf{v}_{S_1} \\ \hline \mathbf{v}_{S_2} \end{array} \right), \tag{8.49}$$

$$\mathbf{B} \cdot \mathbf{v}_S \approx [\!]_T \cdot \mathbf{A}_{T \leftarrow S} \cdot \rule{1em}{0.4pt}_S \cdot \mathbf{v}_S. \tag{8.50}$$

To use (8.50), we need to form $\rule{1em}{0.4pt}_S \cdot \mathbf{v}_S$. Combining (8.47) and (8.49) yields that

$$\rule{1em}{0.4pt}_S \cdot \mathbf{v}_S \approx \mathbf{E}_S \cdot \left(\begin{array}{c} \rule{1em}{0.4pt}_{S_1} \cdot \mathbf{v}_{S_1} \\ \hline \rule{1em}{0.4pt}_{S_2} \cdot \mathbf{v}_{S_2} \end{array} \right), \tag{8.51}$$

allowing us to construct $\rule{1em}{0.4pt}_S \cdot \mathbf{v}_S$ efficiently, given $\rule{1em}{0.4pt}_{S_1} \cdot \mathbf{v}_{S_1}$ and $\rule{1em}{0.4pt}_{S_2} \cdot \mathbf{v}_{S_2}$. We can obtain $\rule{1em}{0.4pt}_{S_1} \cdot \mathbf{v}_{S_1}$ and $\rule{1em}{0.4pt}_{S_2} \cdot \mathbf{v}_{S_2}$ similarly, via recursion (splitting in half S_1 and S_2, etc.).

Similarly, to apply $[\!]_T$ to a vector $\mathbf{w}_T$, we partition $\mathbf{P}_T \cdot \mathbf{w}_T$:

$$\mathbf{P}_T \cdot \mathbf{w}_T = \left(\begin{array}{c} \mathbf{w}_{T_1} \\ \hline \mathbf{w}_{T_2} \end{array} \right). \tag{8.52}$$

Combining (8.48) and (8.52) yields that

$$[\!]_T \cdot \mathbf{w}_T \approx \left(\begin{array}{c} [\!]_{T_1} \cdot \mathbf{w}_{T_1} \\ \hline [\!]_{T_2} \cdot \mathbf{w}_{T_2} \end{array} \right). \tag{8.53}$$

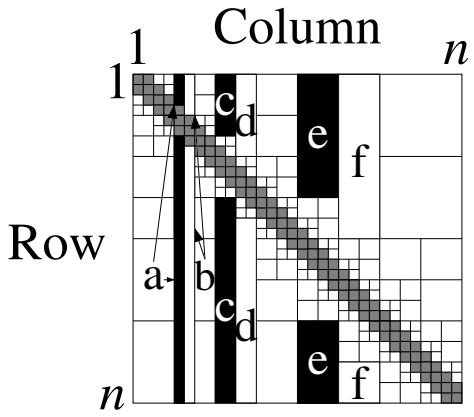

Figure 8.3: "a," "b," "c," "d," "e," and "f" are sets of entries consisting of contiguous collections of columns of the displayed $n \times n$ matrix, without some of the entries near the diagonal. Specifically, if k denotes the width in columns of a swath (say one of "a" or "b"), then the diagonal $k \times k$ block and its two nearest $k \times k$ neighbors are omitted.

Of course, we will also need to apply $[\![]\!]_{T_1}$ and $[\![]\!]_{T_2}$ to vectors arising from blocks other than $\mathbf{B}$ in the decomposition depicted in Figure 8.2. Rather than forming $[\![]\!]_{T_1} \cdot \mathbf{w}_{T_1}$ and $[\![]\!]_{T_2} \cdot \mathbf{w}_{T_2}$ at the stage in the procedure when $\mathbf{w}_{T_1}$ and $\mathbf{w}_{T_2}$ become available, we add $\mathbf{w}_{T_1}$ to the other vectors to which we must apply $[\![]\!]_{T_1}$, and add $\mathbf{w}_{T_2}$ to the other vectors to which we must apply $[\![]\!]_{T_2}$ (and then recursively split in half T_1 and T_2, and repeat).

8.6 Tree organization

To tabulate the costs of the hierarchical application, we need a great deal of notation, detailing the full procedure. The present and subsequent sections set up the required notation.

We will describe the algorithm as applicable in any dimension for any number of points. However, for motivation, we consider the case when the set P of source points and the set Q of target/test points both consist of points in $\mathbb{R}^1$, specifically the numbers 1, 2, ..., $n-1$, n, where n is some positive integer power of 2. The algorithm will be efficient when, as in Section 8.3, there is a reasonably small positive integer m such that each of the blocks of matrix entries indicated in Figure 8.2 that does not touch the diagonal can be accurately approximated by a matrix of rank at most m. The multilevel algorithm efficiently applies the whole matrix to a vector by applying each of the numerically low-rank blocks, and summing up the results.

We will denote the entries of the matrix being applied by $G(x, y)$, where $x \in P$ and $y \in Q$; G is known as the *kernel* of the interactions between points in P and points in Q. We will denote the entries of the vector to which the matrix is being applied by $\rho(x)$, where $x \in P$; ρ is known as the input charge distribution. The purpose of the algorithm is to compute efficiently the sums

$$\sum_{x \in P} \rho(x)\, G(x, y) \tag{8.54}$$

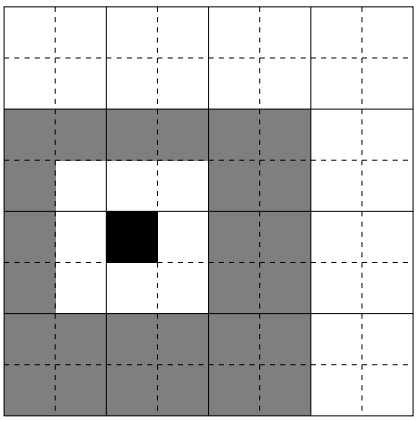

Figure 8.4: The shaded cubes that surround the isolated dark black cube are all its patrons — the source cubes that are of the same size as the dark black cube and that do not touch it, but that do lie in the source cubes of the same size as the parent of the dark black cube which touch the parent.

for all $y \in Q$.

In the remainder of the present section, as well as in Sections 8.7 and 8.8, we set notation and construct explicit low-rank representations for the undarkened blocks in Figure 8.2.

We denote the dimension of the space containing P and Q by d; that is, $P \subseteq \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^d$. We define the largest source cube to be a cube which contains all of the source points, the points in P. We then choose a positive integer l, and form a tree structure by recursively dividing any source cube containing more than l points into 2^d source cubes (known as *children*), each with half the sidelength of the parent. The operation count of the algorithm will depend on l; later we will choose $l \approx m$ to minimize the operation count.

Similarly, we define the largest target/test cube to be a source cube which contains all of the target/test points, the points in Q. We then form a tree structure by recursively dividing any target/test cube containing more than l points into 2^d target/test cubes (known as *children*), each with half the sidelength of the parent.

For any target/test cube C, we say that B is a *patron* of C to mean that B is a source cube of the same size as C that does not touch C, but that does lie in a cube of the same size as the parent of C which touches the parent. Figure 8.4 displays a particular target/test cube C in $\mathbb{R}^2$, as well as all the patrons of C.

8.7 Source tree

In this section, we detail the procedure from Section 8.4 for sources (columns in the matrix).

We first organize $\mathbb{R}^d$ in a tree for P, as in Section 8.6. We then mark all source cubes as unprocessed. We sweep through all of the source cubes multiple times, processing during every sweep each unprocessed cube whose children have all been processed (on the first sweep, we process each cube which does not have any children).

To process a source cube C which does not have any children, we define P_C to be the set

of all points that lie in both P and C, *i.e.*,

$$P_C = P \cap C. \tag{8.55}$$

To process a source cube C whose children have already been processed, we define P_C to be the union of all the sets U_B defined below, with B ranging over all children B of C, *i.e.*,

$$P_C = \bigcup_B U_B, \tag{8.56}$$

where the union is taken over all children of C.

We choose the smallest positive integer m such that we can construct an accurate ID for every source cube C as follows: we construct a subset U_C of at most m points from P_C, and at most m functions $g_{C,z}$, with z ranging through U_C, such that

$$f_y(x) \approx \sum_{z \in U_C} f_y(z)\ g_{C,z}(x) \tag{8.57}$$

for any $x \in P_C$, and any $y \in T_C$, where f_y is the function defined on P_C via the formula

$$f_y(x) = G(x, y), \tag{8.58}$$

and T_C is the set of all points in Q that lie outside of both C and the cubes of the same size as C that touch C. The range of $g_{C,z}$ should be reasonably small (lying entirely in the interval $[-2, 2]$, for example), to ensure numerical stability in accordance with the definition of an ID. The operation count of the multilevel algorithm for the application of the matrix to a vector will be at most a constant times m.

Remark 36 For each source cube C, the set U_C contains at most m points and the set P_C contains at most $2^d\, m$ points (at least when $l \leq 2^d\, m$).

8.8 Target/test tree

In this section, we detail the procedure from Section 8.4 for targets (rows in the matrix).

We now reverse the roles P and Q played in Section 8.7, gathering the points in Q according to the cubes to which they belong.

We first organize $\mathbb{R}^d$ in a tree for Q, as in Section 8.6. We then mark all target/test cubes as unprocessed. We sweep through all of the target/test cubes multiple times, processing during every sweep each unprocessed cube whose children have all been processed (on the first sweep, we process each cube which does not have any children).

To process a target/test cube C which does not have any children, we define Q_C to be the set of all points that lie in both Q and C, *i.e.*,

$$Q_C = Q \cap C. \tag{8.59}$$

To process a target/test cube C whose children have already been processed, we define Q_C to be the union of all the sets V_B defined below, with B ranging over all children B of C, *i.e.*,

$$Q_C = \bigcup_B V_B, \tag{8.60}$$

where the union is taken over all children of C.

We choose the smallest positive integer m such that we can construct an accurate ID for every target/test cube C as follows: we construct a subset V_C of at most m points from Q_C, and at most m functions $h_{C,z}$, with z ranging through V_C, such that

$$e_x(y) \approx \sum_{z \in V_C} e_x(z)\ h_{C,z}(y) \tag{8.61}$$

for any $y \in Q_C$, and any $x \in S_C$, where e_x is the function defined on Q_C via the formula

$$e_x(y) = G(x, y), \tag{8.62}$$

and S_C is the set of all points in P that lie outside of both C and the cubes of the same size as C that touch C. The range of $h_{C,z}$ should be reasonably small (lying entirely in the interval $[-2, 2]$, for example), to ensure numerical stability in accordance with the definition of an ID. The operation count of the multilevel algorithm for the application of the matrix to a vector will be at most a constant times m. For simplicity, we use the same integer m in both the present section and the previous section, Section 8.7.

Remark 37 For each target/test cube C, the set V_C contains at most m points and the set Q_C contains at most $2^d\, m$ points (at least when $l \leq 2^d\, m$).

8.9 Multilevel algorithm

In this section, we discuss details of the procedure from Section 8.5.

First, we discuss the rationale behind the algorithm. We use throughout the notation from Sections 8.6, 8.7, and 8.8, assuming that the decompositions required there are already available (Section 8.11 describes fast methods for constructing the required decompositions).

For any target/test cube C that does not have any children, we break up the sum (8.54) that we want to compute as follows:

$$\sum_{x \in P} \rho(x)\ e_x(y) = \left(\varphi_C(y) + \sum_{x \in P \setminus S_C} \rho(x)\ e_x(y) \right) \tag{8.63}$$

for any $y \in Q_C$, where

$$\varphi_C(y) = \sum_{x \in S_C} \rho(x)\ e_x(y), \tag{8.64}$$

e_x is defined (8.62), Q_C is defined in (8.59), and S_C is the set of all points in P that lie outside of both C and the cubes of the same size as C that touch C.

Thus, knowing the values of φ_C defined in (8.64), we can use (8.63) to reconstruct the sum

$$\sum_{x \in P} \rho(x)\ e_x(y) = \sum_{x \in P} \rho(x)\ G(x, y), \tag{8.65}$$

where e_x is defined (8.62). The last step, Step 5, of the full algorithm (detailed below) uses (8.63).

In order to calculate φ_C defined in (8.64) efficiently, we do the following. For any target/test cube C, we define the function ψ_C on Q_C (defined in (8.59) and (8.60)) via the formula

$$\psi_C(y) = \sum_B \sum_{x \in U_B} \rho_B(x) \; G(x,y), \tag{8.66}$$

where the leftmost sum in the right-hand side of (8.66) is taken over all patrons of C (see Figure 8.4), U_B is defined in Section 8.7, and the function ρ_B is defined on U_B via the recursion

$$\rho_B(z) = \sum_A \sum_{x \in U_A} \rho_A(x) \; g_{B,z}(x), \tag{8.67}$$

where the leftmost sum in the right-hand side of (8.67) is taken over all children of B, and where $g_{B,z}$ is from (8.57), and U_B and U_A are defined in Section 8.7; if B does not have any children, then we replace (8.67) with

$$\rho_B(z) = \sum_{x \in P_B} \rho(x) \; g_{B,z}(x), \tag{8.68}$$

where $P_B = P \cap B$, as in (8.55), and $g_{B,z}$ is from (8.57). Please note that ψ_C defined in (8.66) consists of reasonably short sums, evaluated at a reasonably small collection of points; this is key to the efficiency of the multilevel algorithm. We will account for most of the interactions between the source points in P and the target/test points in Q using these functions. Step 3 of the algorithm detailed below consists of using (8.66) to construct ψ_C. Step 2 calculates ρ_B in (8.67), unrolling the recursion, with Step 1 calculating ρ_B in (8.68) for source cubes that have no children.

Combining (8.57), (8.61), (8.64), (8.66), (8.67), and (8.68) yields that

$$\varphi_C(y) \approx \left(\psi_C(y) + \sum_{z \in V_C} \varphi_B(z) \; h_{C,z}(y) \right) \tag{8.69}$$

for any $y \in Q_C$, where B is the parent of C, φ_C and φ_B are defined in (8.64), ψ_C is defined in (8.66), $h_{C,z}$ is from (8.61), V_C is defined in Section 8.8, and Q_C is defined in (8.59) and (8.60). Step 3 involves calculating ψ_C defined in (8.66). Step 4 uses (8.69) to obtain φ_C defined in (8.64), given the analogous function φ_B for the parent B of C. Step 5 uses (8.63), together with (8.57) and (8.68).

Remark 38 It is possible to accelerate the computation of the sum in (8.69), as well as the associated computation in Step 4 below, using the fact that the set S_B defined in Section 8.8 is a proper subset of S_C (S_B is the set of all points in P that lie outside of both B and the cubes of the same size as B that touch B; S_C is the set of all points in P that lie outside of both C and the cubes of the same size as C that touch C). For simplicity, we will not be taking advantage of this acceleration (though the acceleration can be substantial in practice).

The following five steps therefore comprise the complete multilevel algorithm. Steps 1 and 2 aggregate source points, Step 3 maps from source points to target/test points, and Steps 4 and 5 disaggregate target/test points (in particular, Step 4 accumulates the results of Step 3 from level to level).

(1) For each source cube C that has no children, we compute the values of the function ρ_C that is defined on U_C (defined in Section 8.7) via the formula

$$\rho_C(z) = \sum_{x \in P_C} \rho(x)\ g_{C,z}(x), \tag{8.70}$$

where $g_{C,z}$ is from (8.57), and P_C is defined in (8.55).

(2) We start by marking as processed all source cubes that have no children, and marking as unprocessed all other cubes. Then, we sweep through all source cubes multiple times, until all are processed. During every sweep, for each unprocessed source cube C whose children have all been processed, we compute the value of the function ρ_C defined on U_C via the formula

$$\rho_C(z) = \sum_B \sum_{x \in U_B} \rho_B(x)\ g_{C,z}(x), \tag{8.71}$$

where the leftmost sum in the right-hand side of (8.71) is taken over all children of C, and where $g_{C,z}$ is from (8.57), and U_C and U_B are defined in Section 8.7. We then mark the cube C as processed.

(3) For each target/test cube C, we compute the values of the function ψ_C that is defined on Q_C (defined in (8.59) and (8.60)) via the formula

$$\psi_C(y) = \sum_B \sum_{x \in U_B} \rho_B(x)\ G(x, y), \tag{8.72}$$

where the leftmost sum in the right-hand side of (8.72) is taken over all patrons of C (see Figure 8.4), and where ρ_B is defined in (8.70) and (8.71), and U_B is defined in Section 8.7.

(4) We start by defining the function φ_C on Q_C (defined in (8.59) and (8.60)) for the largest target/test cube C, via the formula

$$\varphi_C(y) = \psi_C(y), \tag{8.73}$$

where ψ_C is defined in (8.72). Then, we mark the largest target/test cube as processed and all other cubes as unprocessed. Finally, we sweep through all target/test cubes multiple times, until all are processed. During every sweep, for each unprocessed target/test cube C whose parent has been processed, we compute the value of the function φ_C that is defined on Q_C (defined in (8.59) and (8.60)) via the formula

$$\varphi_C(y) = \psi_C(y) + \sum_{z \in V_C} \varphi_B(z)\ h_{C,z}(y), \tag{8.74}$$

where B is the parent of C, ψ_C is defined in (8.72), $h_{C,z}$ is from (8.61), and V_C is defined in Section 8.8. We then mark the cube C as processed.

(5) For each target/test cube C that has no children, we compute the value of the function Φ_C that is defined on Q_C (defined in (8.59)) via the formula

$$\Phi_C(y) = \varphi_C(y) + \Sigma_1(y) + \Sigma_2(y), \tag{8.75}$$

where φ_C is defined in (8.73) and (8.74), and where

$$\Sigma_1(y) = \sum_B \sum_{x \in U_B} \rho_B(x)\ G(x,y), \tag{8.76}$$

with the leftmost sum in the right-hand side of (8.76) being taken over all childless source cubes that do not touch C but that do lie in the cubes of the same size as C that touch C, with ρ_B defined in (8.70), and

$$\Sigma_2(y) = \sum_B \sum_{x \in P \cap B} \rho(x)\ G(x,y), \tag{8.77}$$

with the leftmost sum in the right-hand side of (8.77) being taken over both C itself and all childless source cubes which are not separated from C by at least a cube of their own size.

The desired sums in (8.54) are the values of the result Φ from the fifth, last step (5) (the function Φ_C in (8.75) is just the restriction of Φ to $Q_C = Q \cap C$).

For an alternative using the singular value decomposition in place of interpolation, see Gimbutas and Rokhlin [21]. For (sometimes dramatic) accelerations, see Cheng, Greengard, and Rokhlin [10], and Martinsson and Rokhlin [39].

Figure 8.2 displays the squares in the matrix representation of a linear operator formed by the Cartesian products of the pairs of source and target/test intervals involved in the third step (3) (the undarkened squares), as well as by the pairs of source and target/test intervals involved in the fifth, last step (5). Obviously, the Cartesian products of the pairs of intervals from the third step (3) and fifth step (5) cover the matrix representation exactly once, with no overlap.

8.10 Computational costs

In this section, we tabulate the numbers of floating-point operations and words of memory required by the algorithm described in Section 8.9.

First, we estimate the number of source and target cubes. Suppose that there are n source points, *i.e.*, P consists of n points. Then, it follows from the fact that the parent of any childless source cube contains at least l points (see Section 8.6), that there are at most n/l such parent cubes. Therefore, since any such parent cube has at most 2^d children, there are at most $2^d\, n/l$ source cubes that do not have any children.

Now, suppose also that there are at most $\log_2(1/\varepsilon)$ levels in the 2^d-ary tree of source cubes, where ε is the precision of computations, the precision of the approximations (8.57) and (8.61). (See Nabors, Korsmeyer, Leighton, and White [47] for a superior but more complicated algorithm that has similar operation counts even without the assumption that

there are only $\log_2(1/\varepsilon)$ levels in the 2^d-ary trees of source and target/test cubes. The assumption is valid in most applications, however, since it is usually possible to arrange for the closest pair of source points to be separated by at least ε, and for the closest pair of target/test points to be separated by at least ε, as well.) Combining the fact that there are at most $2^d\, n/l$ source cubes that have no children, and the fact that each such cube has at most $\log_2(1/\varepsilon)$ ancestors, yields that there are at most

$$n_P = 2^d\, (n/l)\, \log_2(1/\varepsilon) \tag{8.78}$$

source cubes in total.

Next, suppose for simplicity that, as with the source points, there are n target/test points, *i.e.*, Q consists of n points. Suppose also that there are at most $\log_2(1/\varepsilon)$ levels in the 2^d-ary tree of target/test cubes. Then, similarly, there are at most

$$n_Q = 2^d\, (n/l)\, \log_2(1/\varepsilon) \tag{8.79}$$

target/test cubes in total.

Keeping in mind Remarks 36 and 37, we obtain the following costs for the corresponding steps of the algorithm described in Section 8.9.

(1) There at most $2^d\, n/l$ source cubes that have no children, at most m points in each U_C, and at most l points in each P_C. Thus, Step 1 costs at most $\mathcal{O}(2^d\, l\, m\, n/l)$.

(2) There at most n_P source cubes, at most 2^d children of any cube, and at most m points in any U_C. Thus, Step 2 costs at most $\mathcal{O}(2^d\, m^2\, n_P)$.

(3) There are at most n_Q target/test cubes, at most $2^d\, m$ points in each Q_C, at most m points in each U_B, and at most $6^d - 3^d$ patrons of C (see Figure 8.4). Thus, Step 3 costs at most $\mathcal{O}((6^d - 3^d)\, 2^d\, m^2\, n_Q)$.

(4) There at most n_Q target/test cubes, at most $2^d\, m$ points in each Q_C, and at most m points in each V_C. Thus, Step 4 costs at most $\mathcal{O}(2^d\, m^2\, n_Q)$.

(5) For any source cube C that has no children, and any fixed size of cube at least as large as C, there are at most 3^d test cubes of that size for which C lies in a cube of that same size that touches one of the 3^d cubes. Therefore, since there are at most $2^d\, n/l$ source cubes that have no children, and at most $\log_2(1/\varepsilon)$ different sizes of cubes, it follows from (8.78) that there are at most $3^d\, n_P$ pairs of source and target/test cubes in total involved in the sums (8.76). Similarly, there are at most $3^d\, n_P$ pairs of source and target/test cubes in total involved the sums (8.77). Moreover, there are at most l points in each Q_C, at most m points in each U_B, and at most l points in each $P \cap B$. Thus, Step 5 costs at most $\mathcal{O}(3^d\, l\, (l+m)\, n_P)$.

Overall, if we take $l = m$, then the whole procedure costs $\mathcal{O}((6^d - 3^d)\, 4^d\, m\, n\, \log_2(1/\varepsilon))$. Suppressing the dependence on the dimension d, we find that the whole procedure costs

$$\mathcal{O}(m\, n\, \log_2(1/\varepsilon)). \tag{8.80}$$

In many circumstances, m is proportional to $\log_2(1/\varepsilon)$. The extra factor of $\log_2(1/\varepsilon)$ is usually not realized in practice. Clearly, the algorithm is efficient whenever m is not too large; for many matrices from physics and engineering, m is not too large when the underlying physical phenomena are not too oscillatory (for analogous algorithms that are efficient for the matrices associated with wave equations that are highly oscillatory, see Cheng et al. [8]).

Remark 39 It is always more efficient to move to earlier steps some of the computations associated with Step 5 described above. However, the more efficient algorithm is somewhat more complicated.

8.11 Chebyshev series

The multipoles described in Section 8.1 are not the only efficient means for obtaining low-rank approximations to the undarkened blocks in Figure 8.2. Often the matrix displayed in Figure 8.2 arises from the discretization of an integral kernel whose blocks associated with the undarkened blocks in Figure 8.2 are smooth as a function of the row and column indices. In such cases, we may compress the undarkened blocks via any procedure for approximating smooth functions. In this section, we describe Chebyshev series, which produce low-rank approximations to smooth blocks.

Before discussing functions of both the row and column indices, we discuss functions of just a single variable: Suppose that f is a uniformly smooth function on $[-1, 1]$. Then, we define a function g on the whole real line via the formula

$$g(t) = f(\cos t). \tag{8.81}$$

It is easy to check that g is uniformly smooth on the whole real line, and, therefore, its Fourier series (of period 2π) converges rapidly. For any positive integer m, we define g_m to be the sum of the first $2m - 1$ terms in the Fourier series for g, the terms involving the constant function, $\cos(t)$, $\sin(t)$, $\cos(2t)$, $\sin(2t)$, $\ldots$, $\cos((m-2)t)$, $\sin((m-2)t)$, $\cos((m-1)t)$, $\sin((m-1)t)$. (In fact, the coefficients of the sines in g_m are all zeros, since g is even.) We define f_m on $[-1, 1]$ via the formula

$$g_m(t) = f_m(\cos t) \tag{8.82}$$

for any real number t. f_m is known as the m^{th} Chebyshev approximation to f.

It follows from the fact that g is uniformly smooth that $g - g_m$ is uniformly small when m is sufficiently large. Combining this fact, (8.81), and (8.82) yields that $f - f_m$ is uniformly small when m is sufficiently large. Thus, the Chebyshev approximation f_m approximates f well when m is sufficiently large.

Suppose now that h is a uniformly smooth function on $[-1, 1] \times [-1, 1]$. For example, h could be a function whose values on a Cartesian grid are the matrix entries in an undarkened block from Figure 8.2. For any $y \in [-1, 1]$, we define $f(x) = h(x, y)$, and observe as above that f_m is a good approximation to f when m is sufficiently large. Moreover, what constitutes sufficiently large is independent of y. Therefore, there exist functions c_0, c_1, $\ldots$, c_{m-2}, c_{m-1}

on $[-1, 1]$ (namely, the coefficients in the Fourier series or Chebyshev approximations) such that

$$h(x, y) \approx \sum_{j=0}^{m-1} T_j(x)\, c_j(y) \tag{8.83}$$

for any $x \in [-1, 1]$ and any $y \in [-1, 1]$, where

$$\cos(jt) = T_j(\cos t) \tag{8.84}$$

for any real number t and $j = 0, 1, \ldots, m-2, m-1$. (8.83) provides a highly accurate approximation of rank at most m to any matrix whose entries are the values of h on a Cartesian grid. Thus, Chebyshev series provide a convenient and reasonably efficient means for constructing low-rank approximations to blocks of a matrix that comes from the discretization of an integral kernel that is smooth away from the diagonal.

Remark 40 The functions T_0, T_1, T_2, ... defined in (8.84) are "Chebyshev polynomials" (the use of "T" stems from alternative transliterations from the Cyrillic of "Chebyshev," such as "Tchebycheff"). In fact, T_j is a polynomial of degree j for $j = 0, 1, 2, \ldots$: Clearly, T_0 and T_1 are polynomials of degree 0 and 1, namely $T_0(x) = 1$ and $T_1(x) = x$. The fact that T_j is a polynomial of degree j for $j = 2, 3, 4, \ldots$ then follows by induction from the recurrence relation

$$T_j(x) = 2x\, T_{j-1}(x) - T_{j-2}(x) \tag{8.85}$$

for $j = 2, 3, 4, \ldots$. It is easy to verify (8.85), using (8.84) and the angle addition and subtraction formulae (for cosine) to derive the equivalent recurrence relation

$$T_{j+1}(\cos t) + T_{j-1}(\cos t) = 2(\cos t)\, T_j(\cos t) \tag{8.86}$$

for any real number t and $j = 1, 2, 3, \ldots$.

Remark 41 Given any (possibly suboptimal) low-rank approximation to a matrix, there exists an efficient algorithm for computing an optimal ID approximating the same matrix to nearly the same precision; see Liberty et al. [37] or Halko et al. [26].

Chapter 9

General references

In this chapter, we provide a sampling of references on background material.

Bound-state and impedance calculations: [4] [42] [54]
Calderón-Zygmund and wavelet theory: [43] [14] [38]
Complex analysis: [2]
Fourier analysis: [17] [30]
Functional analysis: [50] [34] [36] [52]
Mathematical methods in physics (including multipole/partial-wave expansions): [45][28][20]
Potential theory: [31] [44]
Scattering theory: [4] [42] [54]
Scientific computation: [13] [49] [55] [53] [22]
Special functions: [1] [56] [18]

Bibliography

[1] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions*, Dover Publications, Mineola, New York, 1972.

[2] L. V. Ahlfors, *Complex Analysis*, McGraw-Hill, New York, third ed., 1979.

[3] K. E. Atkinson, *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge University Press, Cambridge, UK, 1997.

[4] M. Born and E. Wolf, *Principles of Optics*, Cambridge University Press, Cambridge, UK, seventh ed., 2003.

[5] J. Bremer, *On the Nyström discretization of integral equations on planar curves with corners*, Appl. Comput. Harmon. Anal., 32 (2012), pp. 45–64.

[6] J. Bremer, V. Rokhlin, and I. Sammis, *Universal quadratures for boundary integral equations on two-dimensional domains with corners*, J. Comput. Phys., 229 (2010), pp. 8259–8280.

[7] O. Bruno, T. Elling, R. Paffenroth, and C. Turc, *Electromagnetic integral equations requiring small numbers of Krylov-subspace iterations*, J. Comput. Phys., 228 (2009), pp. 6169–6183.

[8] H. Cheng, W. Crutchfield, Z. Gimbutas, L. Greengard, J. F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao, *A wideband fast multipole method for the Helmholtz equation in three dimensions*, J. Comput. Phys., 216 (2006), pp. 300–325.

[9] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin, *On the compression of low rank matrices*, SIAM J. Sci. Comput., 26 (2005), pp. 1389–1404.

[10] H. Cheng, L. Greengard, and V. Rokhlin, *A fast multipole algorithm in three dimensions*, J. Comput. Phys., 155 (1999), pp. 468–498.

[11] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and efficient algorithms in computational electromagnetics*, Artech House, Boston, 2001.

[12] D. L. Colton and R. Kress, *Integral Equation Methods in Scattering Theory*, John Wiley & Sonds, New York, 1983.

[13] G. Dahlquist and Å. Björck, *Numerical Methods*, Dover Publications, Mineola, New York, 1974.

[14] I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.

[15] J. Dongarra and F. Sullivan, *Guest editors introduction: The top 10 algorithms*, AIP-IEEE Computing in Science and Engineering, 2 (2000), pp. 22–23.

[16] R. Duan and V. Rokhlin, *High-order quadratures for the solution of scattering problems in two dimensions*, J. Comput. Phys., 228 (2009), pp. 2152–2174.

[17] H. Dym and H. P. McKean, *Fourier Series and Integrals*, Academic Press, San Diego, California, 1972.

[18] Employees of Wolfram Research, et al., *The Functions Site*. Hosted and supported by Wolfram Research. Available at http://functions.wolfram.com.

[19] C. L. Epstein and L. Greengard, *Debye sources and the numerical solution of the time-harmonic Maxwell equations*, Comm. Pure Appl. Math., 63 (2010), pp. 413–463.

[20] U. H. Gerlach, *Linear mathematics in infinite dimensions: Signals, boundary-value problems, and special functions*, May 2010. Available at http://www.math.ohio-state.edu/∼gerlach/math/BVtypset/BVtypset.html.

[21] Z. Gimbutas and V. Rokhlin, *A generalized fast multipole method for nonoscillatory kernels*, SIAM J. Sci. Comput., 24 (2002), pp. 796–817.

[22] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, third ed., 1996.

[23] S. A. Goreinov and E. E. Tyrtyshnikov, *The maximal-volume concept in approximation by low-rank matrices*, in Structured Matrices in Mathematics, Computer Science, and Engineering I: Proceedings of an AMS-IMS-SIAM Joint Summer Research Conference, University of Colorado, Boulder, June 27–July 1, 1999, V. Olshevsky, ed., vol. 280 of Contemporary Mathematics, Providence, RI, 2001, AMS Publications, pp. 47–51.

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

[25] M. Gu and S. C. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.

[26] N. Halko, P. Martinsson, and J. A. Tropp, *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Review, 53 (2011), pp. 217–288.

[27] J. Helsing, *Solving integral equations on piecewise smooth boundaries using the RCIP method: a tutorial*, Abstr. Appl. Anal., 2013 (2013), pp. 1–20.

[28] J. D. Jackson, *Classical Electrodynamics*, John Wiley & Sons, New York, third ed., 1999.

[29] S. Kapur and V. Rokhlin, *High-order corrected trapezoidal quadrature rules for singular functions*, SIAM J. Numer. Anal., 34 (1997), pp. 1331–1356.

[30] Y. Katznelson, *An Introduction to Harmonic Analysis*, Cambridge University Press, Cambridge, UK, third ed., 2004.

[31] O. Kellogg, *Foundations of Potential Theory*, Dover Publications, Mineola, New York, 1969.

[32] M. Kilian, *On the Riemann-Hilbert problem*, Master's thesis, Technical University of Berlin, 1994.

[33] A. Klöckner, A. Barnett, L. Greengard, and M. O'Neil, *Quadrature by expansion: a new method for the evaluation of layer potentials*, J. Comput. Phys., 252 (2013), pp. 332–349.

[34] A. N. Kolmogorov and S. V. Fomin, *Introductory Real Analysis*, Dover Publications, Mineola, New York, revised English ed., 1975.

[35] J. Kuczyński and H. Woźniakowski, *Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1094–1122.

[36] P. D. Lax, *Functional Analysis*, John Wiley & Sons, New York, 2002.

[37] E. Liberty, F. Woolfe, P. Martinsson, V. Rokhlin, and M. Tygert, *Randomized algorithms for the low-rank approximation of matrices*, Proc. Nat. Acad. Sci. (USA), 104 (2007), pp. 20167–20172.

[38] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, California, second ed., 1999.

[39] P.-G. Martinsson and V. Rokhlin, *An accelerated kernel-independent fast multipole method in one dimension*, SIAM J. Sci. Comput., 29 (2007), pp. 1160–1178.

[40] P.-G. Martinsson, V. Rokhlin, and M. Tygert, *On interpolation and numerical integration in finite-dimensional spaces of bounded functions*, Comm. Appl. Math. Comput. Sci., 1 (2006), pp. 133–142.

[41] A. McIntosh and M. Mitrea, *Clifford algebras and Maxwell's equations in Lipschitz domains*, Math. Meth. Appl. Sci., 22 (1999), pp. 1599–1620.

[42] E. Merzbacher, *Quantum Mechanics*, John Wiley & Sons, New York, third ed., 1998.

[43] Y. Meyer, *Wavelets and Operators*, Cambridge University Press, Cambridge, UK, English ed., 1992.

[44] S. Mikhlin, *Integral Equations and Their Applications to Certain Problems in Mechanics*, Macmillan, New York, second revised ed., 1964.

[45] P. Morse and H. Feshbach, *Methods of Theoretical Physics*, McGraw-Hill, New York, 1953.

[46] C. Müller, *Foundations of the Mathematical Theory of Electromagnetic Waves*, Springer, Berlin, revised and enlarged translation with T. P. Higgins ed., 1969.

[47] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, *Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory*, SIAM J. Sci. Comput., 15 (1994), pp. 713–735.

[48] N. Nishimura, *Fast multipole accelerated boundary integral equation methods*, Applied Mechanics Reviews, 55 (2002), pp. 299–324.

[49] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes*, Cambridge University Press, Cambridge, UK, second ed., 1992.

[50] F. Riesz and B. Sz.-Nagy, *Functional Analysis*, Dover Publications, Mineola, New York, English ed., 1990.

[51] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.

[52] W. Rudin, *Functional Analysis*, McGraw-Hill, New York, second ed., 1991.

[53] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, second ed., 2003.

[54] L. Schiff, *Quantum Mechanics*, McGraw-Hill, New York, third ed., 1990.

[55] E. Tyrtyshnikov, *A Brief Introduction to Numerical Analysis*, Birkhauser, Boston, 1997.

[56] E. Weisstein, et al., *MathWorld*. Hosted and supported by Wolfram Research. Available at http://mathworld.wolfram.com.

[57] N. Yarvin and V. Rokhlin, *Generalized Gaussian quadratures and singular value decompositions of integral operators*, SIAM J. Sci. Comput., 20 (1998), pp. 699–718.