Course notes: Fast Methods in Scientific Computation

Per-Gunnar Martinsson

March 11, 2011

Contents

CHAPTER 1

Introduction

CHAPTER 2

Basic concepts

This chapter reviews some basic tools from numerical analysis, and provides some pointers on Matlab programming. Prior knowledge of the material is very much expected!

2.1. Quadrature

Consider the task of approximating the integral:

$$I = \int_a^b f(x)\,dx$$

where f is a given (reasonably smooth) function. Fix an integer n, set

$$h = \frac{b-a}{n}$$

and define the points $\{x_j\}_{j=0}^n$ via

$$x_j = a + j\,h.$$

In particular,

$$x_0 = a, \qquad \text{and} \qquad x_n = b.$$

One of the simplest approximations is via the *trapezoidal rule:*

$$I_1 = \frac{h}{2}\left(f(x_0) + \sum_{j=1}^{n-1} 2\,f(x_j) + f(x_n) \right).$$

If f is a C^2 function, then

$$|I - I_1| = O(h^2),$$

as $h \to 0$.

A slightly better approximation is obtained from *Simpson's rule*. Supposing that n is an even number, it reads:

$$\begin{aligned} I_2 &= \frac{h}{3}\left(f(x_0) + \sum_{j=1}^{n/2} \big(4\,f(x_{2j-1}) + 2\,f(x_{2j})\big) + f(x_n) \right) \\ &= \frac{h}{3}\left(f(x_0) + 4\,f(x_1) + 2\,f(x_2) + 4\,f(x_3) + 2\,f(x_4) + \cdots + 2\,f(x_{n-2}) + 4\,f(x_{n-1}) + f(x_n) \right) \end{aligned}$$

If f is a C^4 function, then

$$|I - I_2| = O(h^4),$$

as $h \to 0$.

The points $\{x_j\}_{j=0}^n$ are called *quadrature points* and the factors $\{w_j\}_{j=0}^n$ multiplying them are called *quadrature weights*. For instance, in Simpsons rule, we have

$$w_0 = \frac{h}{3}, \quad w_1 = \frac{4h}{3}, \quad w_2 = \frac{2h}{3}, \quad w_3 = \frac{4h}{3}, \quad \cdots \quad w_{n-1} = \frac{4h}{3}, \quad w_n = \frac{h}{3}.$$

Both the trapezoidal rule and the Simpson rule are examples of quadrature from the class of rules called *Newton-Cotes formulas*. These are handy when the integrand is given at equispaced points. If you can pick the quadrature points yourself, then there are more accurate methods available. Of particular use in practical work are the so called *Gaussian quadrature rules*. For a given *order* p, a Gaussian rule has

$$\begin{aligned} \text{nodes:} \quad & \{x_j\}_{j=1}^p, \\ \text{weights:} \quad & \{w_j\}_{j=1}^p. \end{aligned}$$

The convergence order is very high. If f is sufficiently smooth, then for some $\xi \in (a,\, b)$:

$$\int_a^b f(x)\, dx - \sum_{j=1}^p w_j\, f(x_j) = \frac{(b-a)^{2n+1}\,(n!)^4}{(2n+1)\,\big((2n)!\big)^3}\, f^{(2n)}(\xi) \tag{2.1}$$

If the interval $I = (a,\, b)$ is large, one typically uses so called *composite* quadratures. We then split I into n parts, each of length h,

$$h = \frac{b-a}{n}, \qquad x_j = a + h\, j, \qquad I_j = [x_{j-1},\, x_j],$$

and write the integral as a sum over the sub intervals:

$$\int_a^b f(x)\, dx = \sum_{j=1}^n \int_{x_{j-1}}^{x_j} f(x)\, dx.$$

Each term in the integral is then evaluated using a p'th order Gaussian quadrature.

2.2. The FFT

The *Fast Fourier Transform (FFT)* is an algorithm (with many variations) for rapidly evaluating the *Discrete Fourier Transform (DFT)*. It was published in 1965 and has become an extremely useful tool in computing. Curiously, it was originally discovered by Gauss around 1805.

For a vector $\mathbf{x} \in \mathbb{C}^N$, the DFT is an $N \times N$ matrix $\mathbf{F}_N$ (or simply $\mathbf{F}$ if the dimension is unambiguous) defined via

$$\mathbf{F}_{N,mn} = e^{-i\,2\,\pi\,m\,n/N}.$$

Given a vector $\mathbf{x} = [\mathbf{x}(m)]_{m=0}^{N-1} \in \mathbb{C}^N$, its *discrete Fourier transform* $\hat{\mathbf{x}}$ is then defined as

$$\hat{\mathbf{x}}(n) = [\mathbf{F}_N\, \mathbf{x}](n) = \sum_{m=0}^{N-1} e^{-i\,2\,\pi\,m\,n/N}\, \mathbf{x}(m), \qquad n = 0,\, 1,\, 2,\, \ldots,\, N-1. \tag{2.2}$$

The DFT preserves Euclidean distance up to a multiplicative factor of $\sqrt{N}$,

$$||\hat{\mathbf{x}}|| = ||\mathbf{F}_N\, \mathbf{x}|| = \sqrt{N}\, ||\mathbf{x}||, \qquad \forall \mathbf{x} \in \mathbb{C}^N,$$

where $||\cdot||$ is the usual Euclidean norm

$$||\mathbf{x}|| = \left(\sum_{m=0}^{N-1} |\mathbf{x}(m)|^2 \right)^{1/2}.$$

Since $\mathbf{F}_N$ is dense, the direct evaluation of the sum (2.2) would require $O(N^2)$ operations. In contrast, the FFT does the job in $O(N \log N)$ operations.

The inverse of the DFT is given by

$$\mathbf{x}(m) = [\mathbf{F}_N^{-1}\hat{\mathbf{x}}](m) = \frac{1}{N} \sum_{n=0}^{N-1} e^{2\,\pi\, i\, m\, n/N}\, \hat{\mathbf{x}}(n).$$

REMARK 2.1. *The inverse of* $\mathbf{F}$ *is in fact its adjoint (or the complex conjugate of its transpose). Observe that for any unitary matrix* $\mathbf{U}$*, it is the case that* $\mathbf{U}\,\mathbf{U}^* = \mathbf{U}^*\,\mathbf{U} = \mathbf{I}$*. The Fourier transform is a unitary map up to scaling by a factor of* $\sqrt{N}$*. In other words,* $\mathbf{F}\,\mathbf{F}^* = N\,\mathbf{I}$*.*

The FFT is a classical example of a *divide-and-conquer* algorithm. It takes a problem of size N, and splits it into two analogous problems of size $N/2$, these are then split further into four analogous problems of size $N/4$, etc. To see how it works, suppose that N is an even number, set $N = 2P$, and split the formula for the DFT into two halves, one that includes the odd indices, and one that includes the even indices:

$$\begin{aligned}
\hat{\mathbf{x}}(n) =& [\mathbf{F}_N \mathbf{x}](n) \\
=& \sum_{m=0}^{N-1} e^{-2\pi i m\, n/N}\, \mathbf{x}(m) \\
=& \sum_{m=0}^{P-1} e^{-2\pi i 2m\, n/(2P)}\, \mathbf{x}(2m) + \sum_{m=0}^{P-1} e^{-2\pi i (2m+1)\, n/(2P)}\, \mathbf{x}(2m+1) \\
=& \sum_{m=0}^{P-1} e^{-2\pi i m\, n/P}\, \mathbf{x}(2m) + e^{-2\pi i n/N} \sum_{m=0}^{P-1} e^{-2\pi i m\, n/P}\, \mathbf{x}(2m+1) \\
=& [\mathbf{F}_{N/2} \mathbf{x}_{\text{even}}](n) + e^{-2\pi i n/N}\, [\mathbf{F}_{N/2} \mathbf{x}_{\text{odd}}](n),
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{x}_{\text{even}} &= \mathbf{x}(0:2:(N-2)) = [\mathbf{x}(0), \mathbf{x}(2), \mathbf{x}(4), \ldots, \mathbf{x}(N-2)]^{\text{t}}, \\
\mathbf{x}_{\text{odd}} &= \mathbf{x}(1:2:(N-1)) = [\mathbf{x}(1), \mathbf{x}(3), \mathbf{x}(5), \ldots, \mathbf{x}(N-1)]^{\text{t}}.
\end{aligned}$$

We see how the task of applying one matrix of size $N \times N$ and been reduced to the task of applying two matrices of size $(N/2) \times (N/2)$.

How much does the FFT cost? We find that the cost is about the same on every level, and since there are roughly $\log N$ levels, the total cost is $O(N \log N)$.

The FFT as described above is a precise algorithm — there is no approximation involved. In practice, it will be executed in finite precision arithmetic and there will be some round-off errors. Curiously, the output of the FFT is often *more* accurate than what you would get be evaluating (2.2) directly!

Some comments:

- The speed of a program depends greatly on how it is implemented. It often pays off to avoid recursions if possible. The reason is that moving data is often more expensive than executing the floating point operations.
- In the simplistic version described here, we assume that N is a power of 2. There exist very effective versions of the method for any N. (The notion that that the performance is poor unless N could be factored into small integers, is no longer accurate — even prime N works fine.)
- The DFT inherently operates in complex arithmetic. There exist analogous transforms that are purely real. For instance, in the *discrete cosine transform (DCT)* a vector is represented as a sum of cosines. Fast transforms very similar to the FFT exist for the DCT.
- The data flow can be organized as a *butterfly.* This is a very powerful idea for which many areas of application have recently been found — look it up!
- The method can be generalized to form *non-uniform* Fourier transforms where the points in neither physical space nor Fourier space need be equispaced.

Most important fact about the FFT: It is ***very*** fast. If you can solve a problem using the FFT to the accuracy required, then you probably should.

REMARK 2.2. *The Matlab definition of* $\mathbf{F}_N$ *is given by*

$$\mathbf{F}_{mn} = e^{-i2\pi(m-1)(n-1)/N}.$$

This definition is convenient when the indices of a vector $\mathbf{x}$ *run from* 1 *to* N*, rather than from* 0 *to* $N-1$*. The difference is purely cosmetic and it is trivial to switch between the definitions.*

2.3. Accuracy and complexity of an algorithm

As we saw in the section on quadratures, it is common that the error of a numerical algorithm depends polynomially on some "discretization parameter." For classical methods for simple tasks, one can often derive theorems that precisely give expressions for the error. In practice, this is sometimes very hard, or, perhaps more frequently, simply not worth the effort. We then estimate the function numerically. Suppose that the error E is a function of a discretization parameter h so that $E = E(h)$. We believe that

$$E(h) \sim c \cdot h^r, \qquad \text{as } h \to 0,$$

where c is some constant and r is some positive integer. Estimating these is in principle easy — simply generate a table of E for some suitably chosen values of h, and plot these in a log-log diagram. We see that

$$\log(E) \sim \log(c) + r\,\log(h),$$

so the convergence order r can be read off by estimating the slope of the graph. (In order to get good grid spacing in the graph, it is useful to space the sample points geometrically, for instance: $h = 1,\ 1/2,\ 1/4,\ 1/8,\ 1/16,\ \ldots$) In practice, complications may arise:

- The asymptotics might not "kick in" until h is very small, making it costly to estimate. But this is useful information in its own right! If you don't see the convergence until h is extremely small, then the method is probably not very good.

- Sometimes convergence is "too fast." You hit round-off errors before you can determine the slope. Well, in this case the method is typically good enough. (And who cares about the exact order in this case?)

A quick-and-dirty way of estimating the convergence order is to cut the discretization parameter in half and take the 2-logarithm of the results. Note that

$$\log_2\left(\frac{E(h)}{E(2h)}\right) \sim \log_2\left(\frac{c\,h^r}{c\,(h/2)^r}\right) \sim \log_2\left(2^r\right) = r.$$

[Rewrite this section — start with a brief review of the Cauchy remainder theorem, and then explain how this applies (or not) in practice.]

[Also comment on parallelization.]

2.4. Good programming habits

2.5. Review of the Gauss and Stokes' theorems

2.6. Exercises:

Question 2.1. Derive the Simpson rule and prove that it has an $O(h^4)$ error.

Hint: To determine the coefficients in the rule, consider the formula

$$\int_{-h}^{h} f(x)\,dx \approx a\,f(-h) + b\,f(0) + c\,f(h).$$

Making the formula exact for $f(x) = 1$, $f(x) = x$, $f(x) = x^2$ yields three equations for three unknowns. Then to estimate the error, notice that if f is any function with four continuous derivatives, then

$$f(x) = f(0) + x\,f'(0) + \frac{1}{2}\,x^2\,f''(0) + \frac{1}{6}\,x^3\,f^{(3)}(0) + \frac{1}{24}\,x^4\,f^{(4)}(\xi)$$

for some ξ such that $|\xi| \le |x|$.

Question 2.2. The purpose of this question is to demonstrate the importance of avoiding loops when writing Matlab programs. Consider the trapezoidal rule

$$I \approx h\left(\frac{1}{2}f(x_0) + \sum_{j=1}^{n-1} f(x_j) + \frac{1}{2}f(x_n)\right),$$

where n is the number of quadrature points and

$$h = \frac{b-a}{n}, \qquad x_j = a + h\,j.$$

Consider the following two Matlab functions:

```
function I = trapezoidal1(f,a,b,n)
h = (b-a)/n;
I = 0.5*h*(f(a) + f(b));
for icount = 1:(n-1)
   I = I + h*f(a+icount*h);
end
return
```

```
function I = trapezoidal2(f,a,b,n)
h = (b-a)/n;
w = h*[0.5,ones(1,n-1),0.5];
x = linspace(a,b,n+1);
I = sum(w.*f(x));
return
```

Let $t_j = t_j(n)$ denote the time required for Matlab to execute "version j" of the program. Show via numerical examples that the ratios $t_j(n)/n$ converge to some numbers c_j as n grows, and estimate these numbers.

Hint: The timing functions `tic` and `toc` will not give accurate answers when the interval between them is very short. A way around this is to execute the function you want to time (say) 1000 times, and then divide the measured time by 1000.

Question 2.3. Implement the trapezoidal rule and the Simpson rule efficiently. Then *measure* the convergence order of your functions for a variety of different (You know them, of course, but let us measure them as an exercise.) In this exercise, you may use the built-in function

```
I = quad(f,a,b,tol)
```

to compute the "exact" answer. Use your functions to estimate the integral

$$I = \int_0^1 \cos\left(\frac{1}{0.01 + x^2}\right)\, dx.$$

Produce (loglog) graphs that plot the error in your function versus h, where $h = 1/n$ and

$$n = 100,\ 200,\ 400,\ 800,\ 1600,\ \ldots,\ 51\,200.$$

How would you estimate the convergence order if you do not have a "reference" function to compute the "exact" answer?

Question 2.4. Repeat Question **2.3**, but now use Gaussian quadrature. In other words, construct a function with calling sequence

```
function I = gaussquad(f,a,b,n,p)
```

that uses Gaussian quadrature on n panels, with p points on each panel. Estimate computationally the "order" of the method as n and p increase. Produce a graph that plots the accuracy of the Simpson rule versus the accuracy of the Gaussian rule for $p = 5$, 10, 15, 20 (for the same number of function evaluations).

Note: I believe that there is no built-in Matlab function for generating Gaussian quadrature nodes and weights. On the webpage you'll find a simple function `lgwt.m` that does this task.

Question 2.5. Write a function for estimating the integral

$$I = \int_c^d \int_a^b f(x_1, x_2)\, dx_1\, dx_2.$$

You may use either Gaussian or Newton-Cotes quadrature. Then estimate the integrals

$$I_1(k) = \int_0^1 \int_0^1 \cos(k\, x_1\, x_2^2) \frac{1}{1+\cos(x_1)}\, dx_1\, dx_2,$$
$$I_2(k) = \int_0^1 \int_0^1 |\cos(k\, x_1\, x_2^2)| \frac{1}{1+\cos(x_1)}\, dx_1\, dx_2,$$

for different values of the positive integer k. Roughly how large can k be for you to quickly (say within a second) get 10 accurate digits?

Question 2.6. Consider three different ways of computing the DFT:

(a) Apply it directly via the formula

$$\hat{\mathbf{x}}(n) = [\mathbf{F}_N\, \mathbf{x}](n) = \sum_{m=0}^{N-1} e^{-i\,2\,\pi\, m\, n/N}\, \mathbf{x}(m), \qquad n = 0,\, 1,\, 2,\, \ldots,\, N-1.$$

(It is strongly recommended to *vectorize* the matrix-vector product.)

(b) Build your own implementation of the basic FFT algorithm described in Section 2.2. (This is the *Cooley-Tukey* algorithm.)

(c) Apply the built-in Matlab command FFT. (Last time I checked, Matlab used the *FFTW* package, but this may have changed.)

Measure the asymptotic timing for each of the methods, and comment on what you observe.

CHAPTER 3

The Laplace and Poisson equations — basic solution strategies

This chapter reviews some of the classical solution strategies for the Laplace and Poisson equations. These techniques lead to extremely efficient numerical solvers when they work, but the caveat is that they do not work very often — only for trivial geometries such as a line, a disc, a rectangle, all of $\mathbb{R}^d$, etc. We will review them anyway since learning about these methods will help us understand basic properties of the equations, and enable us to derive equally efficient methods for more generic situations in later chapters.

3.1. Problems on finite domains in $\mathbb{R}$

3.1.1. A 1D problem with periodic boundary conditions. Consider the differential equation

$$-u''(x) = f(x), \qquad x \in (-\pi, \pi) \tag{3.1}$$

with periodic boundary conditions

$$u(-\pi) = u(\pi), \qquad u'(-\pi) = u'(\pi). \tag{3.2}$$

First we construct an analytic solution. Suppose

$$u(x) = \sum_{n=-\infty}^{\infty} \hat{u}_n \frac{e^{i\,n\,x}}{\sqrt{2\pi}},$$
$$f(x) = \sum_{n=-\infty}^{\infty} \hat{f}_n \frac{e^{i\,n\,x}}{\sqrt{2\pi}},$$

where $\{\hat{u}_n\}_{n=-\infty}^{\infty}$ and $\{\hat{f}_n\}_{n=-\infty}^{\infty}$ are the *Fourier coefficients* of u and f, respectively,

$$\hat{u}_n = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{-i\,n\,x}\, u(x)\, dx,$$
$$\hat{f}_n = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{-i\,n\,x}\, f(x)\, dx.$$

We find that

$$-u''(x) = \sum_{n=-\infty}^{\infty} n^2\, \alpha_n \frac{e^{i\,n\,x}}{\sqrt{2\pi}},$$

so (3.1) takes the form

$$\sum_{n=-\infty}^{\infty} n^2\, \hat{u}_n \frac{e^{i\,n\,x}}{\sqrt{2\pi}} = \sum_{n=-\infty}^{\infty} \hat{f}_n \frac{e^{i\,n\,x}}{\sqrt{2\pi}}. \tag{3.3}$$

We see that the solution is

$$u(x) = \sum_{n\neq 0} \frac{\hat{f}_n}{n^2}\,\frac{e^{i\,n\,x}}{\sqrt{2\pi}}, \tag{3.4}$$

where

$$\hat{f}_n = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{-i\,n\,x} f(x)\,dx.$$

Next, let us discretize (2.2) using the standard difference formula

$$-u''(x) = \frac{1}{h^2}\big(-u(x-h) + 2\,u(x) - u(x+h)\big) + O(h^2).$$

We split the interval $(-\pi,\,\pi)$ into N pieces of length

$$h = \frac{2\pi}{N},$$

and set

$$x_m = -\pi + h\,m.$$

Further, let $\mathbf{u}, \mathbf{f} \in \mathbb{C}^N$ denote approximations to u and f,

$$\mathbf{u}(m) \approx u(x_m),$$
$$\mathbf{f}(m) = f(x_m).$$

The discrete analog of (3.1) is

$$\frac{1}{h^2}\big(-\mathbf{u}(n-1) + \mathbf{u}(2n) - \mathbf{u}(n+1)\big) = \mathbf{f}(n). \tag{3.5}$$

Define a shift operator

$$[\tau_p \mathbf{u}](m) = \mathbf{u}(m-p).$$

Observe that

$$\begin{aligned}
[\mathsf{F}(\tau_p \mathbf{u})](n) &= \sum_{m=0}^{N-1} e^{-i\,2\,\pi\,m\,n/N}\,\mathbf{u}(m-p) \\
&= \sum_{k=0}^{N-1} e^{-i\,2\,\pi\,(k+p)\,n/N}\,\mathbf{u}(k) \\
&= e^{-i\,2\,\pi\,p\,n/N} \sum_{k=0}^{N-1} e^{-i\,2\,\pi\,k\,n/N} \mathbf{u}(k) \\
&= e^{-i\,2\,\pi\,p\,n/N}\,\hat{\mathbf{u}}(n).
\end{aligned}$$

In Fourier space, (3.5) takes the form

$$\frac{1}{h^2}\big(-e^{i\,2\,\pi\,n/N} + 2 - e^{-i\,2\,\pi\,n/N}\big)\,\hat{\mathbf{u}}(n) = \hat{\mathbf{f}}(n), \qquad n = 0,\,1,\,2,\,\ldots,\,N-1. \tag{3.6}$$

Now recall that for any $z \in \mathbb{C}$:

$$-e^{iz} + 2 - e^{-iz} = -(e^{iz/2} - e^{-iz/2})^2 = -(i\sin(z/2))^2 = 4\,\sin^2(z/2).$$

Consequently, (3.6) simplifies to

$$\sigma(n)\,\hat{\mathbf{u}}(n) = \hat{\mathbf{f}}(n), \tag{3.7}$$

where

$$\sigma(n) = \frac{4}{h^2}\,\sin^2(\pi n/N).$$

The key point here is that the difference equation (3.5) is by the Fourier transform converted to the algebraic equation (3.7). This equation can easily be solved,

$$\hat{\mathbf{u}}(n) = \begin{cases} \frac{1}{\sigma(n)}\,\hat{\mathbf{f}}(n), & n \neq 0 \\ 0 & n = 0. \end{cases}$$

and then the approximation to u is recovered via the inverse Fourier transform

$$u(x_m) \approx \mathbf{u}(m) = [\mathbf{F}^{-1}\,\hat{\mathbf{u}}](m) = \frac{1}{N}\sum_{n=0}^{N-1} \mathbf{u}(n)\, e^{2\,\pi\, i\, m\, n/N} = \frac{1}{N}\sum_{n=0}^{N-1} \frac{\mathbf{f}(n)}{\sigma(n)}\, e^{2\,\pi\, i\, m\, n/N}.$$

REMARK 3.1. *When N is large, the low frequency modes of the DFT in a sense "converge" to the continuous Fourier transform. For instance, since $h = 2\pi/N$, we see that*

$$\sigma(n) = \frac{4}{(2\pi/N)^2}\,\sin^2(\pi n/N) = \frac{N^2}{\pi^2}\,\sin^2(\pi n/N) \approx n^2$$

when n/N is small. The factor "n^2" is precisely what we saw for in the continuous case, cf. (3.3).

REMARK 3.2. *The solution formula (3.4) can be rewritten as a convolution:*

$$\begin{aligned} u(x) &= \sum_{n\neq 0} \frac{1}{n^2}\frac{e^{i\,n\,x}}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \frac{e^{-i\,n\,y}}{2\pi}\, f(y)\, dy \\ &= \int_{-\pi}^{\pi} \left(\sum_{n\neq 0} \frac{e^{i\,n\,(x-y)}}{2\,\pi\,n^2} \right) f(y)\, dy \\ &= \int_{-\pi}^{\pi} G(x-y)\, f(y)\, dy, \end{aligned}$$

where G is the fundamental solution of the equation:

$$G(x) = \sum_{n\neq 0} \frac{e^{i\,n\,x}}{2\,\pi\,n^2}$$

REMARK 3.3. *How many Fourier modes do you need to get a requested accuracy? Suppose you approximate u by the first N modes of the solution*

$$u_{\text{approx}}(x) = \sum_{n=-N}^{N} \hat{u}_n\, \frac{e^{inx}}{\sqrt{2\pi}}. \tag{3.8}$$

Then using the Parseval formula, we see that

$$||u - u_{\text{approx}}||^2_{L^2(I)} = \int_{-\pi}^{\pi} |u(x) - u_{\text{approx}}(x)|^2\, dx = \sum_{|n|>N} |\hat{u}_n|^2 = \sum_{|n|>N} \frac{|\hat{f}_n|^2}{n^4}.$$

In other words, the error depends on how rapidly the Fourier modes of f decay — the slower the decay, the more terms we need to achieve any given accuracy. And the speed of decay in $(\hat{f}_n)_{n=-\infty}^{\infty}$ in turn depends on how smooth f is. For instance, if $f \in C^p(I)$ for some integer p, then using

partial integration, we see that

$$\begin{aligned}
|\hat{f}_n| &= \frac{1}{\sqrt{2\pi}} \left| \int_{-\pi}^{\pi} e^{inx}\, f(x)\, dx \right| \\
&= \frac{1}{\sqrt{2\pi}} \left| \int_{-\pi}^{\pi} \frac{1}{inx} e^{inx}\, f'(x)\, dx \right| \\
&= \frac{1}{\sqrt{2\pi}} \left| \int_{-\pi}^{\pi} \frac{1}{(inx)^2} e^{inx}\, f''(x)\, dx \right| = \cdots \\
&= \frac{1}{\sqrt{2\pi}} \left| \int_{-\pi}^{\pi} \frac{1}{(inx)^p} e^{inx}\, f^{(p)}(x)\, dx \right| \\
&\leq \frac{1}{\sqrt{2\pi}} \frac{1}{|n|^p} \int_{-\pi}^{\pi} |f^{(p)}(x)|\, dx.
\end{aligned}$$

(Periodicity makes the boundary terms in the partial integrations vanish.)

REMARK 3.4. *We are sometimes interested in evaluating the derivative of the approximation* u_{approx} *as defined by (3.8). In a sense, this is easy since*

$$u'_{\text{approx}}(x) = \sum_{n=-N}^{N} in\, \hat{u}_n \frac{e^{inx}}{\sqrt{2\pi}} = \sum_{n=-N}^{N} \frac{i}{n}\, \hat{f}_n \frac{e^{inx}}{\sqrt{2\pi}}. \tag{3.9}$$

However, it is important to note that $||u' - u'_{\text{approx}}||$ *converges more slowly than* $||u - u_{\text{approx}}||$*. In fact:*

$$||u' - u'_{\text{approx}}||^2_{L^2(I)} = \sum_{|n|>N} |in\, \hat{u}_n|^2 = \sum_{|n|>N} \frac{|\hat{f}_n|^2}{n^2}.$$

The observation that the computed derivatives are less accurate than the computed function values is very common in the context of numerical methods for PDEs (finite elements, finite differences, fourier methods, boundary integral equation methods, etc.). You would typically aim for approximating u well, do an OK job of approximating the first derivative, and not even hope for a good approximation to the higher derivatives.

3.1.2. A 1D problem with homogeneous Dirichlet boundary conditions. Consider the differential equation

$$\begin{cases} -u''(x) = & f(x), \qquad x \in (0, \pi), \\ u(0) = & 0, \\ u(\pi) = & 0. \end{cases} \tag{3.10}$$

This equation can also be solved via Fourier methods. A suitable basis is now

$$\varphi_n(x) = \sqrt{\frac{2}{\pi}}\, \sin(n\, x), \qquad n = 1,\, 2,\, 3,\, \ldots \tag{3.11}$$

We find that

$$\begin{cases} -\varphi''_n(x) = & n^2\, \varphi_n(x), \qquad x \in (0, \pi), \\ \varphi_n(-\pi) = & 0, \\ \varphi_n(\pi) = & 0. \end{cases} \tag{3.12}$$

Moreover, $(\varphi_n)_{n=1}^{\infty}$ is an orthonormal basis for $L^2(I)$ — the set of square integrable functions on $I = (0, \pi)$. We let $(\hat{u}_n)$ and $(\hat{f}_n)$ again be the expansion coefficients

$$u(x) = \sum_{n=1}^{\infty} \hat{u}_n \, \varphi_n(x),$$
$$f(x) = \sum_{n=1}^{\infty} \hat{f}_n \, \varphi_n(x)$$

where $\{\hat{u}_n\}_{n=-\infty}^{\infty}$ and $\{\hat{f}_n\}_{n=-\infty}^{\infty}$ are the *Fourier coefficients* of u and f, respectively,

$$\hat{u}_n = \int_0^{\pi} \varphi_n(x) \, u(x) \, dx,$$
$$\hat{f}_n = \int_0^{\pi} \varphi_n(x) \, f(x) \, dx.$$

The sequences $(\hat{u}_n)$ and $(\hat{f}_n)$ can be computed via the FFT, and so can the map back to physical space.

REMARK 3.5. *It seems very fortuitous that we should have such convenient sets of orthogonal functions for expanding the solution. This is not a coincidence! Many of the differential equations that arise are for physical reasons "self-adjoint." And one can prove that for many self-adjoint operators you can construct a sequence of functions that is both a orthonormal basis for the relevant function space, and eigenfunctions of the differential operator. In fact all self-adjoint operators admit a somewhat weaker statement: namely that there is some unitary transform that maps them to diagonal operators. For details, look up the spectral theorem.*

REMARK 3.6. *The eigenfunctions φ_n defined by (3.11) can easily be determined by considering the differential equation*

$$\begin{cases} -\varphi''(x) = \lambda \varphi(x), & x \in (0, \pi), \\ \varphi(0) = 0, \\ \varphi(\pi) = 0. \end{cases}$$

The solutions to the equation $\varphi'' + \lambda \varphi = 0$ depend on the sign of λ:

(1) *If $\lambda = 0$, the solutions are $\varphi(x) = A + B\,x$.*
(2) *If $\lambda = -\alpha^2$, the solutions are $\varphi(x) = A\,e^{\alpha x} + B\,e^{-\alpha x}$.*
(3) *If $\lambda = \alpha^2$, the solutions are $\varphi(x) = A\,\cos(\alpha x) + B\,\sin(\alpha x)$.*

The only solutions that satisfy the boundary conditions take the form

$$\varphi(x) = A\,\sin(n\,x),$$

where n is a positive integer. The choice $A = \sqrt{2/\pi}$ is made so that

$$1 = ||\varphi||_{L^2(I)} = \left(\int_{-\pi}^{\pi} |\varphi(x)|^2 \, dx \right)^{1/2}.$$

A slightly different line of attack would be to separate the issues of (i) body load and (ii) boundary conditions. To this end, let us first construct a function v that satisfies

$$-v''(x) = f(x), \qquad x \in I,$$

and not care about the boundary conditions. This can easily be done via Fourier methods as outlined in Section 3.1.1. Now let us see if we can find a solution u of (3.10) by adding a correction w to the solution v:

$$u = v + w.$$

We find that w must satisfy the equation

$$\left\{ \begin{array}{rcll} -w''(x) = & 0, & & x \in (-\pi, \pi), \\ w(-\pi) = & -v(-\pi), & & \\ w(\pi) = & -v(\pi). & & \end{array} \right. \tag{3.13}$$

Solving this equation is easy. Since $w'' = 0$, we know that w must be linear:

$$w(x) = A\,x + B.$$

Using the boundary conditions, we easily determine A and B and find that

$$w(x) = -v(-\pi) - \frac{x+\pi}{2\pi}\, v(\pi).$$

REMARK 3.7. *The general technique of first finding a solution that satisfies the body load, and then to find a "correction" that solved a homogeneous problem and that takes care of the boundary conditions is often applicable. The function v is called a "particular" solution.*

3.2. Poisson's equation on $\mathbb{R}^d$ (no boundary conditions)

Next consider the equation

$$-\Delta u(x) = f(x), \qquad x \in \mathbb{R}^d. \tag{3.14}$$

For this equation, it is slightly tricky to formulate proper boundary conditions, but you have to do *something*, otherwise the problem is not well-posed: For any given solution u of (3.14), the function $v(x) = u(x) + a + b \cdot x$ would also be a solution for any $a \in \mathbb{R}$ and $b \in \mathbb{R}^d$. We will avoid this discussion for now.

First we solve (3.14) via Fourier methods. The proper transform is now the *continuous Fourier transform:*

$$\hat{u}(t) = [\mathcal{F}u](t) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\,t\cdot x}\, u(x)\, dx, \qquad t \in \mathbb{R}^d.$$

The inverse transform is given by

$$u(x) = [\mathcal{F}^* \hat{u}](x) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{i\,t\cdot x}\, \hat{u}(t)\, dt, \qquad x \in \mathbb{R}^d.$$

Via partial integrations, we see that for $j = 1, 2, \ldots, d$:

$$\begin{aligned} [\mathcal{F}(\partial_{x_j} u)](t) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\,t\cdot x}\, (\partial_{x_j} u(x))\, dx &= -\frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} (\partial_{x_j} e^{-i\,t\cdot x})\, u(x)\, dx \\ &= -\frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} (-i\,t_j)\, e^{-i\,t\cdot x}\, u(x)\, dx = i\,t_j\, \hat{u}(t). \end{aligned}$$

In consequence, $[\mathcal{F}(\partial_{x_j}^2 u)](t) = (i\,t_j)^2\, \hat{u}(t) - t_j^2\, \hat{u}(t)$, and so

$$[\mathcal{F}(-\Delta u)](t) = (t_1^2 + t_2^2 + \cdots + t_d^2)\, \hat{u}(t) = |t|^2\, \hat{u}(t).$$

Now (3.14) in Fourier space takes the form

$$|t|^2\, \hat{u}(t) = \hat{f}(t), \qquad t \in \mathbb{R}^d.$$

This is an algebraic equation that we easily solve:

$$\hat{u}(t) = \frac{1}{|t|^2}\,\hat{f}(t),$$

and then u is recovered via the inverse Fourier transform

$$u(x) = [\mathcal{F}^*\hat{u}](x) = \left[\mathcal{F}^*\frac{\hat{f}(t)}{t^2}\right](x) = \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} e^{i\,t\,x}\frac{\hat{f}(t)}{|t|^2}\,dt.$$

Some comments:

- The utility of the Fourier method depends on how smooth f is and how rapidly it decays.
- The continuous Fourier transform can sometimes be evaluated efficiently by combining the FFT with a quadrature rule. This requires some care when f is non-smooth, or when f does not decay rapidly (of course, these situations make things more complicated for any numerical method).

Alternatively, one could solve the equation via a so called *fundamental solution*. To get there, recall that a multiplication in Fourier space corresponds to a convolution in physical space. In other words, if $\hat{f}$ and $\hat{g}$ are the Fourier transforms of two functions f and g, and

$$\hat{h}(t) = \hat{f}(t)\,\hat{g}(t),$$

then[1]

$$h(x) = [\mathcal{F}^*\hat{h}](x) = [\mathcal{F}^*(\hat{f}\,\hat{g})](x) = [f * g](x) = \int_{\mathbb{R}^d} f(x-y)\,g(y)\,dy. \tag{3.15}$$

(To make (3.15) mathematically rigorous, one has to either assume that f and g are both sufficiently smooth and sufficiently decaying to make all integrals make sense in a classical way, or reinterpret the integrals as more general linear operators.) Specifically, with

$$\phi = \mathcal{F}^*\left[\frac{1}{|t|^2}\right] \tag{3.16}$$

we find that

$$u(x) = \left[\mathcal{F}^*\left(\frac{1}{|t|^2}\,\hat{f}\right)\right](x) = [\phi * f](x) = \int_{\mathbb{R}^d}\phi(x-y)\,f(y)\,dy.$$

The formula (3.16) is deceptively simple. Observe that the Fourier integral

$$\phi(x) = \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} e^{ix\cdot t}\frac{1}{|t|^2}\,dt$$

is not integrable (in the Lebesgue sense) for any d! (When $d = 1$, it is not integrable at the origin, when $d \geq 3$, it is not integrable at infinity, and when $d = 2$, it is integrable neither at the origin, nor at infinity.) This problem can be overcome, and with suitable generalization of the Fourier transform one can show that

$$\begin{aligned}
\phi(x) &= -\frac{1}{2}|x| && d = 1\\
\phi(x) &= -\frac{1}{2\pi}\log|x| && d = 2\\
\phi(x) &= \frac{1}{4\pi|x|} && d = 3.
\end{aligned}$$

[1]There is a factor of 2π missing here - check!

REMARK 3.8 (Mathematical properties of $\mathcal{F}$). *The map $\mathcal{F}$ is a unitary transform of $L^2(\mathbb{R}^d)$ to $L^2(\mathbb{R}^d)$, meaning that*

$$\mathcal{F}^*\mathcal{F} = \mathcal{F}\mathcal{F}^* = I.$$

As a consequence

$$||u||^2_{L^2(\mathbb{R}^d)} = (\mathcal{F}^*\mathcal{F}u,\, u) = (\mathcal{F}u,\, \mathcal{F}u) = ||\hat{u}||^2_{L^2(\mathbb{R}^d)}.$$

The Fourier transform maps smooth functions to rapidly decaying functions, and vice versa:

$$\begin{aligned} f \text{ smooth} \quad &\Leftrightarrow \quad \hat{f} \text{ decays rapidly} \\ f \text{ decays rapidly} \quad &\Leftrightarrow \quad \hat{f} \text{ smooth} \end{aligned}$$

REMARK 3.9. *Properly defining the Fourier transform on $\mathbb{R}$ takes some work since the integral in the definition is typically not absolutely integrable. For instance, if you try to take the Fourier transform of the function $x \mapsto (1/2)|x|$ you'll have problems at infinity (the integrand grows!) and if you try to take the Fourier transform of the function $t \mapsto 1/t^2$ you'll have problems at the origin. One way around this difficulty is to consider a class of generalized functions often called "distributions." Well-known distributions include:*

The Dirac δ-function: *The function δ is a function such that (loosely speaking) $\delta(x) = 0$ for all x, but*

$$\int_{-\infty}^{\infty} f(x)\,\delta(x)\,dx = f(0),$$

for any sufficiently smooth function f.

The Heaviside function: *The Heaviside function H is simply*

$$H(x) = \begin{cases} -1 & x < 0, \\ 1 & 0 \geq x. \end{cases}$$

PV$(1/x)$: *The "principal value" function of $1/x$ is a "function" that satisfies function satisfies:*

$$\int_{-\infty}^{\infty} PV(1/x)\, f(x)\,dx = \lim_{\varepsilon \searrow 0} \left(\int_{-\infty}^{-\varepsilon} \frac{1}{x}\, f(x)\,dx + \int_{\varepsilon}^{\infty} \frac{1}{x}\, f(x)\,dx \right).$$

The point of distribution theory is to rigorously define such functions, to specify how to compute their derivatives, their Fourier transforms, etc. In this framework, you can properly formulate an equation such as

$$-\phi'' = \delta$$

and find that the solution is

$$\phi(x) = \frac{1}{2}|x|.$$

The solution to the equation $-u'' = f$ is then given as

$$u(x) = [\phi * f](x) = \int_{-\infty}^{\infty} \phi(x-y)\, f(y)\,dy$$

since

$$-u''(x) = -\frac{d^2}{dx^2} \int_{-\infty}^{\infty} \phi(x-y)\, f(y)\,dy = \int_{-\infty}^{\infty} (-\phi''(x-y))\, f(y)\,dy = \int_{-\infty}^{\infty} \delta(x-y)\, f(y)\,dy = f(x).$$

WARNING:*Most equations in this remark are "improper" in the sense that they must be interpreted in a distributional sense. This material is provided as an encouragement to study distributional theory; reading this remark puts you at risk of "knowing just enough to be dangerous."*

REMARK 3.10. *The solution formula*

$$u(x) = \int_{\mathbb{R}^d} \phi(x-y)\, f(y)\, dy$$

can often be evaluated directly via a quadrature rule. There is however no good "generic" quadrature rule for this case since (a) the domain of integration is infinite, (b) the function f may or may not be smooth, and (c) the function $\phi(x-y)$ is singular as $y \to x$. However, assuming that we have somehow resolved these issues, we would be left with the task of evaluating a sum such as

$$u(x_i) = \sum_{n=1}^{N} w_j\, \phi(x-y_j)\, f(y_j), \qquad i = 1,\, 2,\, \ldots,\, M \tag{3.17}$$

where $\{y_j\}_{j=1}^N$ are the quadrature nodes, $\{w_j\}_{j=1}^N$ are the quadrature weights, and $\{x_i\}_{i=1}^M$ is the set of points where the solution is sought. Evaluating the sum (3.17) directly requires $O(M\,N)$ operations, which quickly gets prohibitive when M and N grow. However, this task is exactly what the Fast Multipole Method (see Chapter 4) is designed for. It evaluates the sum in $O(M+N)$ operations.

3.3. Problems on finite domains in $\mathbb{R}^2$

3.3.1. The square. Let $\Omega = [0,\, \pi] \times [0,\, \pi]$ be a square in the plane with boundary $\Gamma = \partial\Omega$. In this section, we will briefly review techniques for solving the equation

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ \quad u(x) = g(x), & x \in \Gamma. \end{cases}$$

We consider first the two special cases where either $g = 0$ or $f = 0$.

3.3.1.1. *Zero Dirichlet data.* Consider the equation

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ \quad u(x) = 0, & x \in \Gamma. \end{cases} \tag{3.18}$$

Define the index set

$$I = \{(n_1,\, n_2) :\ n_1 \text{ and } n_2 \text{ are positive integers}\},$$

and the basis functions

$$\varphi_n(x_1, x_2) = \frac{2}{\pi} \sin(n_1 x_1)\, \sin(n_2 x_2), \qquad n \in I.$$

These basis functions have three important properties:

(1) The basis functions form an *orthonormal set* in $L^2(\Omega)$:

$$\int_\Omega \varphi_n(x)\, \varphi_n(x)\, dA(x) = \begin{cases} 1 & \text{when } m = n \\ 0 & \text{when } m \neq n. \end{cases}$$

(2) The basis functions *span* $L^2(\Omega)$ which means that any function $f \in L^2(\Omega)$ can be written

$$f = \sum_{n \in I} \hat{f}_n\, \varphi_n(x),$$

where $\hat{f}_n$ are the expansion coefficients

$$\hat{f}_n = \int_\Omega \varphi_n(x)\, f(x)\, dA(x).$$

(3) The basis functions are *eigenfunctions* of (3.18):

$$\begin{cases} -\Delta\varphi_n(x) = (n_1^2 + n_2^2)\,\varphi_n(x), & x \in \Omega, \\ \varphi_n(x) = 0, & x \in \Gamma. \end{cases}$$

We now look for a solution of the form

$$u = \sum_{n\in I} \hat{u}_n\,\varphi_n.$$

Since

$$-\Delta u = \sum_{n\in I} (n_1^2 + n_2^2)\hat{u}_n\,\varphi_n,$$

and

$$f = \sum_{n\in I} \hat{f}_n\,\varphi_n(x),$$

we see that the expansion coefficients of u are

$$\hat{u}_n = \frac{1}{n_1^2 + n_2^2}\,\hat{f}_n,$$

and so

$$u = \sum_{n\in I} \frac{1}{n_1^2 + n_2^2}\,\hat{f}_n\,\varphi_n.$$

REMARK 3.11. *How do you find the basis functions φ_n and the index set I? This is done by separating variables. We look for an eigenvector φ of the form*

$$\varphi(x_1, x_2) = \psi_1(x_1)\,\psi_2(x_2)$$

that satisfies

$$\begin{cases} -\Delta\varphi(x) = \varphi(x), & x \in \Omega, \\ \varphi(x) = 0, & x \in \Gamma. \end{cases}$$

We find

$$-\psi_1''(x_1)\,\psi_2(x_2) - \psi_1(x_1)\,\psi_2''(x_2) = \lambda\,\psi_1(x_1)\,\psi_2(x_1).$$

Dividing by $\psi_1\psi_2$ we obtain the equation

$$-\frac{\psi_1''(x_1)}{\psi_1(x_1)} - \frac{\psi_2''(x_2)}{\psi_2(x_2)} = \lambda.$$

Since the first term on the LHS is the only one that depends on x_1, we find that it must be constant:

$$-\frac{\psi_1''(x_1)}{\psi_1(x_1)} = \alpha$$

for some α. Using also that $\psi_1(0) = \psi_1(\pi) = 0$, we find that (cf. Remark 3.6) $\alpha = n_1^2$ for some positive integer, and

$$\psi_1 = A\,\sin(n_1\,x_1)$$

for some constant A. Analogously, one of course determines that $\psi_2 = B\,\sin(n_2\,x_2)$ and then the constants A and B are determined via the condition that

$$1 = ||\varphi||^2_{L^2(\Omega)} = \int_\Omega |\varphi(x)|^2\,dx.$$

3.3.1.2. *Zero body load.* Now consider the equation

$$\begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ u(x) = g(x), & x \in \Gamma. \end{cases} \tag{3.19}$$

Let us split the boundary data g into four separate function, one on each side (the south, east, west, and north sides, respectively):

$$\begin{aligned} g(t,0) =& g^{(\mathrm{s})}(t), \\ g(\pi,t) =& g^{(\mathrm{e})}(t), \\ g(t,\pi) =& g^{(\mathrm{n})}(t), \\ g(0,t) =& g^{(\mathrm{w})}(t). \end{aligned}$$

Then write the solution u of (3.19) as

$$u = v + w$$

where

$$\begin{cases} -\Delta v(x) = 0, & x \in \Omega, \\ v(0,x_2) = g^{(\mathrm{w})}(x), & 0 \le x_2 \le \pi \\ v(\pi,x_2) = g^{(\mathrm{e})}(x), & 0 \le x_2 \le \pi \\ v(x_1,0) = 0, & 0 \le x_1 \le \pi \\ v(x_1,0) = \pi, & 0 \le x_1 \le \pi, \end{cases} \tag{3.20}$$

and

$$\begin{cases} -\Delta w(x) = 0, & x \in \Omega, \\ w(0,x_2) = 0, & 0 \le x_2 \le \pi \\ w(\pi,x_2) = 0, & 0 \le x_2 \le \pi \\ w(x_1,0) = g^{(\mathrm{s})}(x), & 0 \le x_1 \le \pi \\ w(x_1,\pi) = g^{(\mathrm{n})}(x), & 0 \le x_1 \le \pi. \end{cases} \tag{3.21}$$

In order to solve (3.20), we look for a solution of the form

$$v(x_1,\, x_2) = \sum_{n=1}^{\infty} v_n(x_1)\, \sin(nx_2).$$

We find that

$$0 = -\Delta v = \sum_{n=1}^{\infty} \left(-v_n''(x_1) + n^2\, v_n(x_1)\right)\, \sin(nx_2)$$

so each v_n must satisfy the differential equation

$$-v_n'' + n^2\, v_n = 0.$$

The solution is

$$v_n(x_1) = A_n\, \cosh(nx_1) + B_n\, \sinh(nx_1). \tag{3.22}$$

Now we need boundary conditions for $v_n(0)$ and $v_n(\pi)$ to determine A_n and B_n. To this end, not that $\{\sin(nx_2)\}_{n=1}^{\infty}$ forms an orthogonal basis for $L^2([0,\,\pi])$, so we can expand $g^{(\mathrm{w})}$ and $g^{(\mathrm{e})}$:

$$g^{(\mathrm{w})}(t) = \sum_{n=1}^{\infty} \hat{g}_n^{(\mathrm{w})} \sin(nt),$$
$$g^{(\mathrm{e})}(t) = \sum_{n=1}^{\infty} \hat{g}_n^{(\mathrm{e})} \sin(nt).$$

Inserting the boundary conditions at $x_1 = 0, \pi$, we find

$$v(0,\,x_2) = \sum_{n=1}^{\infty} v_n(0) \sin(nx_2) = g^{(\mathrm{w})}(x_2)$$
$$v(\pi,\,x_2) = \sum_{n=1}^{\infty} v_n(\pi) \sin(nx_2) = g^{(\mathrm{e})}(x_2).$$

So the functions v_n have boundary conditions

$$v_n(0) = \hat{g}_n^{(\mathrm{w})} \tag{3.23}$$
$$v_n(\pi) = \hat{g}_n^{(\mathrm{e})}. \tag{3.24}$$

Combining (3.22) and (3.23), we find that

$$v_n(x_1) = \hat{g}_n^{(\mathrm{w})} \cosh(nx_1) + \frac{\hat{g}_n^{(\mathrm{e})} - \hat{g}_n^{(\mathrm{w})} \cosh(n\pi)}{\sinh(n\pi)} \sinh(nx_1).$$

Finally, the solution of (3.20) is

$$v(x_1, x_2) = \sum_{n=1}^{\infty} \left(\hat{g}_n^{(\mathrm{w})} \cosh(nx_1) + \frac{\hat{g}_n^{(\mathrm{e})} - \hat{g}_n^{(\mathrm{w})} \cosh(n\pi)}{\sinh(n\pi)} \sinh(nx_1) \right) \sin(n\,x_2).$$

Of course, equation (3.21) is solved analogously.

3.3.1.3. *Non-zero boundary data, and non-zero body load.*

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ u(x) = g(x), & x \in \Gamma. \end{cases} \tag{3.25}$$

Simply combine, let v be the solution to

$$\begin{cases} -\Delta v(x) = f(x), & x \in \Omega, \\ v(x) = 0, & x \in \Gamma, \end{cases}$$

let w be the solution to

$$\begin{cases} -\Delta w(x) = 0, & x \in \Omega, \\ w(x) = g(x), & x \in \Gamma, \end{cases}$$

and then set

$$u = v + w.$$

The function v can be determined using the methods of Section 3.3.1.2 and w can be determined using the methods of Section 3.3.1.1.

Alternatively, you could find a solution v to

$$-\Delta v = f$$

with *periodic* boundary data:

$$\begin{aligned} v(0,x_2) =& v(\pi,x_2), && 0 \le x_2 \le \pi, \\ [\partial_1 v](0,x_2) =& [\partial_1 v](\pi,x_2), && 0 \le x_2 \le \pi, \\ v(x_1,0) =& v(x_1,\pi), && 0 \le x_1 \le \pi, \\ [\partial_2 v](x_1,0) =& [\partial_2 v](x_1,\pi), && 0 \le x_1 \le \pi. \end{aligned}$$

This is easier technically since the plain FFT applies. Then for the homogeneous problem, you'd need to solve

$$\begin{cases} -\Delta w(x) = 0, & x \in \Omega, \\ \quad w(x) = -v(x) + f(x), & x \in \Gamma. \end{cases}$$

3.3.2. The disc. Next let Ω denote a disc with boundary Γ:

$$\begin{aligned} \Omega =& \{x \in \mathbb{R}^2 : \ |x| < 1\} \\ \Gamma =& \{x \in \mathbb{R}^2 : \ |x| = 1\}, \end{aligned}$$

and consider the boundary value problem

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ \quad u(x) = g(x), & x \in \Gamma, \end{cases}$$

It is now convenient to switch to *polar coordinates* (r,θ) which are defined by

$$x_1 = r\,\cos\theta \qquad x_2 = r\,\sin\theta.$$

Then we think of u as a function of r and θ, and the Laplace operator takes the form

$$-\Delta u = -\partial_r^2 u - \frac{1}{r}\partial_r u - \frac{1}{r^2}\partial_\theta^2 u.$$

In this section, we will look for solutions of the form

$$u(r,\theta) = \sum_{n=0}^{\infty} \big(v_n(r)\,\cos(n\theta) + w_n(r)\,\sin(n\theta)\big). \tag{3.26}$$

3.3.2.1. *Zero body load.* Consider the equation

$$\begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ \quad u(x) = g(x), & x \in \Gamma. \end{cases} \tag{3.27}$$

We look for a solution of the form

$$u(r,\theta) = \sum_{n=-\infty}^{\infty} c_n\, r^{|n|}\, e^{in\theta}. \tag{3.28}$$

Then

$$-\Delta u = 0$$

for any numbers $(c_n)_{n=-\infty}^{\infty}$ such that the sum converges sufficiently fast. To determine the numbers (c_n), we apply the boundary condition. For $r = 1$, we obtain

$$g(\cos\theta,\sin\theta) = u(1,\theta) = \sum_{n=-\infty}^{\infty} c_n\, e^{in\theta},$$

so the c_n's are simply the Fourier coefficients of g:

$$c_n = \frac{1}{2\pi}\int_0^{2\pi} e^{-in\theta} g(\cos\theta,\sin\theta)\,d\theta.$$

REMARK 3.12. *How did we know to make the Ansatz (3.28)? It can again be obtained from separation of variables: We look for a solution of the form* $u(r,\theta) = v(r)\,w(\theta)$ *that satisfies* $-\Delta u = 0$. *This translates to the equation*

$$-v''(r)\,w(\theta) - \frac{1}{r}\,v'(r)\,w(\theta) - \frac{1}{r^2}\,v(r)\,w''(\theta) = 0.$$

Now "separate the variables" by multiplying the equation by r^2/vw*:*

$$-r^2\frac{v''(r)}{v(r)} - r\,\frac{v'(r)}{v(r)} = \frac{w''(\theta)}{w(\theta)}.$$

Since the LHS depends only on r*, and the RHS depends only on* θ*, there must be some number* α *such that*

$$\frac{w''(\theta)}{w(\theta)} = \alpha. \tag{3.29}$$

The solutions to (3.29) depend on the sign of α*:*

(1) *If* $\alpha = -\beta^2$*, then the solutions are* $w(\theta) = A\,e^{i\beta\theta} + B\,e^{-i\beta\theta}$.
(2) *If* $\alpha = 0$*, then the solutions are* $w(\theta) = A + B\,\theta$.
(3) *If* $\alpha = \beta^2$*, then the solutions are* $w(\theta) = A\,e^{\beta\theta} + B\,e^{-\beta\theta}$.

Since w *must be* 2π*-periodic, the only possibility is that* $\alpha = -n^2$ *for some integer* n*, and the corresponding solution is*

$$w(\theta) = e^{\pm in\theta}.$$

Returning to the separated equation, we now find that v *must satisfy*

$$-r^2\frac{v''(r)}{v(r)} - r\,\frac{v'(r)}{v(r)} = -n^2$$

for some integer n*. This simplifies to the Euler equation*

$$r^2 v''(r) + r\,v'(r) - n^2\,v(r) = 0$$

which has solutions

$$v(r) = A\,r^n + B\,r^{-n}$$

if n *is non-zero, and*

$$v(r) = A + B\,\log(r)$$

if $n = 0$*. Using that the solution has to be bounded as* $r \to 0$ *we discard the* $\log(r)$ *and the negative powers of* r *and are left with the basis function in (3.28).*

REMARK 3.13. *The Laplace problem on a disc can alternatively be solved very easily using complex variables. (This remark assumes some knowledge of harmonic analysis.) If* u *is a function that is continuous on* $\overline{\Omega}$*, and harmonic on* Ω*, then there is an analytic function* $w = w(z)$ *such that*

$$u(z) = Re(w(z)).$$

The disc is now parameterized with the complex variable $z = x_1 + i\,x_2$*. Since* w *is analytic, it has a Taylor expansion*

$$w(z) = \sum_{n=0}^{\infty} \alpha_n\,z^n.$$

Now simply observe that

$$Re(z^n) = r^n\,\cos(n\theta), \qquad Im(z^n) = r^n\,\sin(n\theta).$$

3.3.2.2. *Zero Dirichlet data.* Consider the equation

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ u(x) = 0, & x \in \Gamma. \end{cases} \tag{3.30}$$

The eigenfunction of (3.30) are

$$\varphi_{n,j}(x) = e^{i\,n\,\theta}\, J_{|n|}(k_j^{|n|}\, r), \qquad n \in \mathbb{Z} \qquad j = 1,\, 2,\, 3,\, \ldots$$

where, for $m = 0,\, 1,\, 2,\, 3,\, \ldots$ the functions J_m are *Bessel functions*, and $k_1^m,\, k_2^m,\, k_3^m,\, \ldots$ are the zeros of the m'th Bessel function. One can show that

$$-\Delta \varphi_{n,j} = (k_j^{|n|})^2.$$

Consequently, the solution to (3.30) is

$$u(r,\theta) = \sum_{n\in\mathbb{Z}} \sum_{j=1}^{\infty} \frac{1}{(k_j^{|n|})^2}\, \hat{f}_{n,j} e^{i\,n\,\theta}\, J_{|n|}(k_j^{|n|}\, r),$$

where

$$\hat{f}_{n,j} = \frac{1}{||\varphi_{n,j}||^2_{L^2(\Omega)}} \int_\Omega \overline{\varphi_{n,j}(r,\theta)}\, f(r,\theta)\, dA.$$

REMARK 3.14. *How find the eigenfunctions? Again use separation of variables. Set* $u(r,\theta) = v(r)\, w(\theta)$*, then the eigenvalue problem*

$$-\Delta u = \lambda u$$

takes the form

$$-r^2 \frac{v''(r)}{v(r)} - r\frac{v'(r)}{v(r)} - \frac{w''(\theta)}{w(\theta)} = \lambda\, r^2.$$

As earlier, we find that by enforcing periodic boundary conditions on w*, the only possible solutions are* $w(\theta) = e^{\pm i\,n\,\theta}$*, and so the equation for* v *reduces to*

$$-r^2 \frac{v''(r)}{v(r)} - r\frac{v'(r)}{v(r)} + n^2 = \lambda\, r^2,$$

which simplifies to the Bessel differential equation:

$$-r^2\, v''(r) - r\, v'(r) + \left(n^2 - \lambda\, r^2\right) v(r) = 0.$$

The solutions are

$$v(r) = A\, J_n(\sqrt{\lambda}\, r) + B\, Y_n(\sqrt{\lambda}\, r),$$

where J_n *and* Y_n *are Bessel functions. Using that* v *should be bounded on* Ω*, we drop the* Y_n*'s. From the boundary value* $v(1) = 0$*, we finally find that the eigenvalues* λ *are determined by the equation*

$$J_n(\sqrt{\lambda}) = 0.$$

3.4. Green's functions

For the case of free space problems such as

$$-\Delta u = f \qquad \mathbb{R}^d,$$

we saw in Section 3.2 that the solution can be written as a convolution:

$$u(x) = [\phi * f](x) = \int_{\mathbb{R}^d} \phi(x-y)\, f(y)\, dy.$$

For a boundary value problems such as

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ u(x) = 0, & x \in \Gamma, \end{cases} \tag{3.31}$$

the corresponding result is that there exists a so called *Green's functions* $F = F(x, y)$ with the property that the solution to (3.31) is given by

$$u(x) = \int_\Omega F(x, y)\, f(y)\, dy.$$

If we know the eigenfunctions of (3.31), we can actually write an explicit formula for F. To this end, note that

$$\begin{aligned} u(x) &= \sum_{n=1}^{\infty} \frac{1}{\lambda_n} (\varphi_n,\, f)\, \varphi_n(x) \\ &= \sum_{n=1}^{\infty} \frac{1}{\lambda_n} \left(\int_\Omega \overline{\varphi_n(y)}\, f(y)\, dy \right) \varphi_n(x) \\ &= \int_\Omega \left(\sum_{n=1}^{\infty} \frac{1}{\lambda_n} \overline{\varphi_n(y)}\, \varphi_n(x) \right),\, f(y)\, dy, \end{aligned}$$

so

$$F(x, y) = \sum_{n=1}^{\infty} \frac{1}{\lambda_n} \overline{\varphi_n(x)}\, \varphi_n(y). \tag{3.32}$$

Evaluating F via the sum (3.32) is not easy computationally. However, for many simple geometries, analytic expressions for F are known, and the Green's function can in these contexts be very useful.

There also exists Green's function $G = G(x, y)$ for the homogeneous problem

$$\begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ u(x) = g(x), & x \in \Gamma. \end{cases} \tag{3.33}$$

The function G constructs a solution to (3.33) via

$$u(x) = \int_\Gamma G(x, y)\, g(y)\, dy.$$

The solution to an equation with both a body load and boundary data

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ u(x) = g(x), & x \in \Gamma \end{cases} \tag{3.34}$$

is of course directly obtained via superposition

$$u(x) = \int_\Omega F(x, y)\, f(y)\, dy + \int_\Gamma G(x, y)\, g(y)\, dy. \tag{3.35}$$

Remark 3.15. *While the concept of a fundamental solution is very restrictive (they exist only for constant coefficient problems with no boundary data), the concept of a Green's function is very general. Solution formulas such as (3.35) exist even for many variable coefficient problems such as*

$$\begin{cases} -\nabla \cdot (a(x)\nabla u(x)) + b(x) \cdot \nabla u(x) + c(x)u(x) = f(x), & x \in \Omega, \\ u(x) = g(x), & x \in \Gamma \end{cases} \tag{3.36}$$

where a, b, and c are functions on Ω. (The key condition is that the function a has to be a positive definite matrix.) Of course, this is not much help computationally since determining G and H is in general much harder than solving (3.34)! (However, there have been exciting developments in this direction recently...)

3.5. Time-dependent problems

If you know the eigenfunctions of the Laplace operator, you can very easily solve time-dependent equations such as the *heat equation*

$$\Delta u = \frac{\partial u}{\partial t} \tag{3.37}$$

and the *wave equation*

$$\Delta u = \frac{\partial^2 u}{\partial t^2}. \tag{3.38}$$

To illustrate how this works, suppose that we know an orthonormal basis $(\varphi_n)_{n=1}^{\infty}$ of eigenvectors for the Laplace operator on some domain Ω

$$\begin{cases} -\Delta\varphi_n = \lambda_n \varphi_n, & \text{on } \Omega, \\ \varphi_n = 0 & \text{on } \Gamma. \end{cases}$$

The eigenvalues satisfy

$$\lambda_n \to \infty, \qquad \text{as } n \to \infty.$$

The idea is now simply to look for a solution to the time-dependent problem of the form

$$u(x,t) = \sum_{n=1}^{\infty} \alpha_n(t)\, \varphi_n(x). \tag{3.39}$$

3.5.1. The heat equation. Insert (3.39) into (3.37) to find that

$$\underbrace{\sum_{n=1}^{\infty} -\lambda_n\, \alpha_n(t)\, \varphi_n(x)}_{=\Delta u} = \underbrace{\sum_{n=1}^{\infty} \alpha_n'(t)\, \varphi_n(x)}_{=\frac{\partial u}{\partial t}}$$

We see that α_n must satisfy

$$-\lambda_n\, \alpha_n = \alpha_n'$$

and so

$$\alpha_n(t) = c_n\, e^{-\lambda_n t}.$$

The constants c_n are determined via an *initial condition* given at time t_0:

$$u(x,t_0) = v(x).$$

We find that

$$u(x,t) = \sum_{n=1}^{\infty} \hat{v}_n\, e^{-\lambda_n (t-t_0)}\, \varphi_n(x),$$

where

$$\hat{v}_n = \int_{\Omega} \overline{\varphi_n(x)}\, v(x)\, dx.$$

3.5.2. The wave equation. Plug (3.39) into (3.38) to find that

$$\underbrace{\sum_{n=1}^{\infty} -\lambda_n \, \alpha_n(t) \, \varphi_n(x)}_{=\Delta u} = \underbrace{\sum_{n=1}^{\infty} \alpha_n''(t) \, \varphi_n(x)}_{=\frac{\partial^2 u}{\partial t^2}}$$

We see that α_n must satisfy

$$-\lambda_n \, \alpha_n = \alpha_n''$$

and so

$$\alpha_n(t) = a_n \, \cos(\omega_n \, t) + b_n \, \sin(\omega_n \, t),$$

where

$$\omega_n = \sqrt{\lambda_n}.$$

The constants a_n and b_n are again determined via an *initial conditions* given at time t_0:

$$\begin{aligned} u(x, t_0) =& v(x). \\ u_t(x, t_0) =& w(x). \end{aligned}$$

We find that

$$u(x,t) = \sum_{n=1}^{\infty} \left(\hat{v}_n \, \cos(\omega_n \, (t - t_0)) + \frac{\hat{w}_n}{\omega_n} \, \sin(\omega_n \, (t - t_0)) \right) \, \varphi_n(x),$$

where

$$\hat{v}_n = \int_\Omega \overline{\varphi_n(x)} \, v(x) \, dx, \qquad \text{and} \qquad \hat{w}_n = \int_\Omega \overline{\varphi_n(x)} \, w(x) \, dx.$$

3.6. Some remarks on eigenfunctions and Fourier methods

Many of the techniques in this chapter are based on computing the *eigenfunctions* of a differential operator such as the Laplace operator. This is conceptually a very gratifying approach, and is a reasonably general technique in the sense that many problems of practical interest in principle can be solved using eigen-expansions. However, it is in practice often *very hard* to compute these functions. Say you need 1000 modes to approximate a solution to reasonable accuracy; then the cost of computing the eigenfunctions is 1000 times as large as the cost of solving a single problem. Moreover, the larger the eigenvalue, the harder the eigenvector is to compute accurately. (Note that in all examples we have seen, the eigenvectors start to oscillate more and more rapidly as the eigenvalue increases — this is true for almost all "Laplace-like" eigenvalue problems.)

For the basic task of solving a given boundary value problem, it is rare that computing the eigenvectors is the best way to go about it. However, there are many situations in which computing eigenvectors is actually the problem one wants to solve. Examples include:

- Acoustics.
- Design of optical devises such as fibre optic cables.
- Quantum physics (each eigenvector is a "state").
- Mechanics (each eigenvector is a "buckling mode").

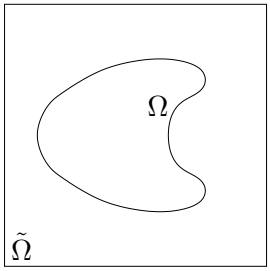


FIGURE 3.1. A general domain Ω is enclosed by a square $\tilde{\Omega}$. We solve the Poisson equation with periodic boundary conditions on $\tilde{\Omega}$ using the FFT (very rapidly!).

While constructing eigenfunctions that satisfy boundary conditions on a "general" domain Ω is quite hard, it is much easier to construct transforms that handle only the differential operator while ignoring the boundary conditions. This is how Fourier methods are often used. To illustrate, suppose we seek to solve the equation

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ u(x) = 0, & x \in \Gamma, \end{cases} \tag{3.40}$$

on some domain Ω with boundary Γ. We can then enclose Ω is a box $\tilde{\Omega}$, cf. Figure 3.1, and solve the equation

$$-\Delta v(x) = g(x), \qquad x \in \tilde{\Omega}, \tag{3.41}$$

where g is a function on $\tilde{\Omega}$ such that $f(x) = g(x)$ for $x \in \Omega$. The equation (3.41) can then be solved using the FFT to very rapidly construct the function v. The final step is then to construct a "correction" w that satisfies

$$\begin{cases} -\Delta w(x) = 0, & x \in \Omega, \\ w(x) = -v(x), & x \in \Gamma, \end{cases} \tag{3.42}$$

whence

$$u = v + w.$$

This strategy is often used in practice. Note however that the construction of the extension g of f is not trivial. You could of course just define

$$g(x) = \chi_\Omega(x) = \begin{cases} f(x) & x \in \Omega, \\ 0 & x \notin \Omega. \end{cases}$$

The drawback is that this function g is discontinuous even when f is smooth (unless of course f is zero near Γ) which means that the Fourier series will converge slowly.

REMARK 3.16. *Techniques for computing large collections of eigenvectors of a given differential operator are presently in a rapid state of development. This is a very exciting and active are of research.*

3.7. Variational formulations

3.8. Energy minimizing formulations

3.9. Conformal maps

3.10. Exercises

Question 3.1. Solve (analytically) the Laplace equation with Dirichlet boundary data on a domain exterior to a unit disc. In other words, let Ω and Γ be defined by

$$\begin{aligned}\Omega =& \{x \in \mathbb{R}^2 : \ |x| > 1\} \\ \Gamma =& \{x \in \mathbb{R}^2 : \ |x| = 1\},\end{aligned}$$

and consider the boundary value problem

$$\begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ \quad u(x) = g(x), & x \in \Gamma. \end{cases}$$

You may assume that $g \in L^1 \cap L^2$, that

$$\int_0^{2\pi} g(\cos\theta, \sin\theta) = 0,$$

and then require the solution u to decay at infinity.

Question 3.2. In this problem, we will numerically solve the equation

$$\begin{cases} -u''(x) = & f(x), & x \in (0, \pi), \\ \quad u(0) = & 0, \\ \quad u(\pi) = & 0, \end{cases} \tag{3.43}$$

for a few different right hand sides. For each given f, compute the solution using either the basis functions $\{\sqrt{2/\pi}\,\sin(n\,x)\}_{n=1}^{\infty}$, or the basis functions $\{\sqrt{1/\pi}\,e^{2inx}\}_{n\in\mathbb{Z}}$ (with appropriate corrections for the boundary conditions). Investigate how rapidly the partial Fourier sums converge. Consider the following choices of f:

(a) $f(x) = e^{(\sin x)^2}$.
(b) $f(x) = e^{\sqrt{x}}$.
(c) $f(x) = \cos(100\,x)$.
(d) $f(x) = \cos(100.1\,x)$.
(e) $f(x) = |x-1|^{-0.25}$.
(f) $f(x) = \begin{cases} 0 & 0 < x \le 1 \\ 1 & 1 < x \le 2 \\ 2 & 1 < x < \pi \end{cases}$

Optional extra questions: Can you construct an analytic solution for any of the given functions? Check how quickly the *derivatives* of the partial Fourier series converge. Can you think of some other "interesting" choices of body loads?

Question 3.3. Consider the Poisson equation

$$-\Delta u(x) = f(x), \qquad x \in \Omega \tag{3.44}$$

where

$$f(x) = \begin{cases} x_1\, e^{-1/(1-|x|^2)}, & |x| < 1, \\ 0 & |x| \geq 1, \end{cases}$$

and where

$$\Omega = (-\pi,\, \pi)^2.$$

(a) Use the FFT to construct computationally a function u that solves (3.44).
(b) Find (numerically) a solution to (3.44) that also satisfies homogeneous Dirichlet boundary conditions.

Question 3.4. Consider the Poisson equation

$$-\Delta u(x) = f(x), \qquad x \in \mathbb{R}^2 \tag{3.45}$$

where

$$f(x) = \begin{cases} x_1\, e^{-1/(1-|x|^2)}, & |x| < 1, \\ 0 & |x| \geq 1. \end{cases}$$

As a boundary condition, require u to converge to zero at infinity (observe that $\int f = 0$).

(a) Switch to polar coordinates, and solve (numerically or otherwise) the equations you obtain.
(b) Can you devise a solution strategy based on the FFT that works for any smooth compactly supported body load?

Question 3.5. Pick a simple domain Ω in the plane (say either a square or a circle) and solve the equation

$$\begin{aligned} -\Delta u + c\, u &= f && \Omega, \\ u &= 0 && \Gamma. \end{aligned}$$

where f is a "point source" placed at some interesting point in the domain. Solve the equation using Fourier methods for different values of c (positive or negative, large or small in magnitude). Plot some representative solutions.

Optional extra question: How would you handle

$$\begin{aligned} -\Delta u + b \cdot \nabla u + c\, u &= f && \Omega, \\ u &= 0 && \Gamma. \end{aligned}$$

Question 3.6. Construct a program that uses Fourier methods to solve the wave equation

$$\begin{cases} \frac{d^2u}{dx^2} = \frac{d^2u}{dt^2}, & x \in (0,\pi), t > 0, \\ u(0,t) = 0, & t > 0 \\ u(\pi,t) = 0, & t > 0 \\ u(x,0) = v, & \\ u_t(x,0) = w, & \end{cases} \tag{3.46}$$

where v and w are given function (play with different choices). Use the Matlab movie function to create an animation of the solution as t increases.

Harder version: Generalize to the wave equation on a square or a disc.

CHAPTER 4

The Fast Multipole Method and other tree codes

4.1. Problem formulation

In this chapter, we describe fast methods for rapidly evaluating the sum

$$u_i = \sum_{j=1}^{N} G(\mathbf{x}_i, \mathbf{x}_j)\, q_j, \qquad i = 1, 2, \ldots, N, \tag{4.1}$$

where $\{\mathbf{x}_i\}_{i=1}^N$ is a given set of points in a square Ω in the plane, where $\{q_i\}_{i=1}^N$ is a given set of real numbers which we call *charges*, and where $\{u_i\}_{i=1}^N$ is a set of real numbers we seek to compute (*potentials*). The *kernel* G is given by

$$G(\mathbf{x}, \mathbf{y}) = \begin{cases} -\frac{1}{2\pi} \log |\mathbf{x} - \mathbf{y}|, & \text{when } \mathbf{x} \neq \mathbf{y} \\ 0 & \text{when } \mathbf{x} = \mathbf{y}. \end{cases} \tag{4.2}$$

Using straight-forward summation techniques, the evaluation of the sum (4.1) requires $\mathrm{O}(N^2)$ floating point operations.

We will describe two techniques for evaluating the sum (4.1) to within a given preset precision ε. The first is a simple technique which is a version of the so called *Barnes-Hut* method that evaluates the sum in $\mathrm{O}(N \log(N) \log(1/\varepsilon))$ floating point operations, see Section 4.5. The second one is a version of the *Fast Multipole Method.* This method is slightly more elaborate, but solves the task in $\mathrm{O}(N \log(1/\varepsilon))$ floating point operations, see Section 4.7. Before we describe either method, however, we describe in Section 4.3 an accelerated technique for solving a much simpler problem in which the charges are contained in one set, and the potentials to be evaluated are in a different one. The techniques used to solve this toy problem form a core part of the more sophisticated summation techniques in Section 4.5 and 4.7.

[... discuss electric charges / gravity / Stokes / other potentials ...]

4.2. Multipole expansions

A multipole expansion is an efficient way of representing a harmonic potential. To be precise, suppose that a box shaped domain Ω holds a set of N sources with charges $\{q_j\}_{j=1}^N$ at locations $\{\mathbf{y}_j\}_{j=1}^N \subset \Omega$. At a point $\mathbf{x}$, the electric potential $u(\mathbf{x})$ caused by these charges is given by

$$u(\mathbf{x}) = \sum_{j=1}^{N} G(\mathbf{x}, \mathbf{y}_j)\, q_j,$$

where G is the kernel defined in (4.2). A very simple approximation to the potential $u(\mathbf{x})$ is obtained by evaluating the potential caused by a single charge at the center $\mathbf{y}_{\mathrm{c}}$ of the box,

$$u(\mathbf{x}) \approx \hat{q}_0 \log |\mathbf{x} - \mathbf{y}_{\mathrm{c}}|. \tag{4.3}$$

In (4.3), $\hat{q}_0$ is simply the net charge of the box,

$$\hat{q}_0 = \sum_{j=1}^{N} q_j. \tag{4.4}$$

The approximation error in (4.3) tends to zero as $|\mathbf{x} - \mathbf{y}_c| \to \infty$, but can be quite large when $\mathbf{x}$ is close to the box Ω. The approximation can be improved by adding a dipole charge at $\mathbf{y}_c$ so that

$$u(\mathbf{x}) \approx \hat{q}_0 \log|\mathbf{x} - \mathbf{y}_c| - \mathbf{d} \cdot \frac{\mathbf{x} - \mathbf{y}_c}{|\mathbf{x} - \mathbf{y}_c|^2}. \tag{4.5}$$

The optimal dipole charge vector $\mathbf{d}$ is given by

$$\mathbf{d} = \sum_{j=1}^{N} q_j \left(\mathbf{y}_j - \mathbf{y}_c\right).$$

The approximations (4.3) and (4.5) form the first two terms in the so called *multipole expansion* of the harmonic function u. The full expansion is conveniently expressed using polar coordinates

$$\mathbf{x} - \mathbf{y}_c = r\,(\cos\theta,\, \sin\theta), \qquad \mathbf{y} - \mathbf{y}_c = r'\,(\cos\theta',\, \sin\theta'). \tag{4.6}$$

Then

$$u(\mathbf{x}) = \hat{q}_0 \log r + \sum_{p=1}^{\infty} \left(\hat{q}_{2p-1} \frac{\cos(p\,\theta)}{r^p} + \hat{q}_{2p} \frac{\sin(p\,\theta)}{r^p} \right), \tag{4.7}$$

where $\hat{q}_0$ is defined by (4.4), and for $p \geq 1$

$$\hat{q}_{2p-1} = -\sum_{j=1}^{N} q_j\,(r')^p \frac{1}{p} \cos(p\,\theta'),$$

$$\hat{q}_{2p} = -\sum_{j=1}^{N} q_j\,(r')^p \frac{1}{p} \sin(p\,\theta').$$

The sum in (4.7) converges exponentially fast whenever $|\mathbf{x} - \mathbf{y}_c| > \sqrt{2}\,a$. Specifically,

$$\left| u(\mathbf{x}) - \hat{q}_0 \log r - \sum_{p=1}^{P} \left(\hat{q}_{2p-1} \frac{\cos(p\,\theta)}{r^p} + \hat{q}_{2p} \frac{\sin(p\,\theta)}{r^p} \right) \right| \sim \left(\frac{\sqrt{2}\,a}{|\mathbf{x} - \mathbf{y}_c|} \right)^P. \tag{4.8}$$

This claim follows directly from the following lemma about the logarithmic kernel:

LEMMA 4.2.1. *Let* $\mathbf{x}$, $\mathbf{y}$, *and* $\mathbf{y}_c$ *be vectors in* $\mathbb{R}^2$ *such that* $|\mathbf{x} - \mathbf{y}_c| > |\mathbf{y} - \mathbf{y}_c|$. *Let for* $p = 0, 2, 3, \ldots$ *the functions* $\tilde{B}_p$ *and* C_p *be defined by*

$$\tilde{B}_0(\mathbf{x}, \mathbf{y}_c) = \log r, \tag{4.9}$$

$$\tilde{B}_{2p-1}(\mathbf{x}, \mathbf{y}_c) = r^{-p} \cos(p\,\theta), \tag{4.10}$$

$$\tilde{B}_{2p}(\mathbf{x}, \mathbf{y}_c) = r^{-p} \sin(p\,\theta), \tag{4.11}$$

and

$$C_0(\mathbf{y}, \mathbf{y}_c) = 1, \tag{4.12}$$

$$C_{2p-1}(\mathbf{y}, \mathbf{y}_c) = -\rho^p \cos(p\,\theta')/p, \tag{4.13}$$

$$C_{2p}(\mathbf{y}, \mathbf{y}_c) = -\rho^p \sin(p\,\theta')/p, \tag{4.14}$$

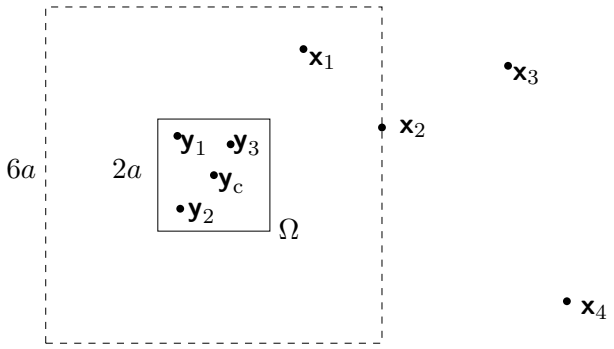


FIGURE 4.1. Geometry of Section 4.2. Any point on or outside of the dashed square is *well-separated* from Ω. Consequently, the points $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$ are well-separated from Ω, but $\mathbf{x}_1$ is not.

where polar coordinates (r, θ) and (r', θ') are defined by (4.6). Let P be a positive integer. Then

$$\left| \log|\mathbf{x} - \mathbf{y}| - \sum_{r=0}^{2P} \tilde{B}_r(\mathbf{x}, \mathbf{y}_c)\, C_r(\mathbf{y}_c, \mathbf{y}) \right| \leq K \left(\frac{r'}{r} \right)^P ,$$

where $K = \cdots$.

A proof of Lemma ?? is given in Appendix ?.

In what follows, we will often replace the potential induced by a source collection by a truncated multipole expansion of the form

$$u(\mathbf{x}) \approx \hat{q}_0 \log r + \sum_{p=1}^{P} \left(\hat{q}_{2p-1} \frac{\cos(p\,\theta)}{r^p} + \hat{q}_{2p} \frac{\sin(p\,\theta)}{r^p} \right). \tag{4.15}$$

To ensure that accuracy is maintained, we will use the approximation only for points $\mathbf{x}$ that are adequately removed from Ω, which we define as follows:

DEFINITION 4.1. Let Ω be a square with side length $2\,a$. we say that a point $\mathbf{x}$ is *well-separated* from Ω if it is outside of the (open) box of side-length $6\,a$ concentric to Ω, see Figure 4.1.

When $\mathbf{y} \in \Omega$ we have $\rho \leq \sqrt{2}a$, and when $\mathbf{x}$ is well-separated from Ω, we have $r \geq 3a$ and ρ, so Lemma 4.2.1 assures us that

$$\left| u(\mathbf{x}) - \hat{q}_0 \log r - \sum_{p=1}^{P} \left(\hat{q}_{2p-1} \frac{\cos(p\,\theta)}{r^p} + \hat{q}_{2p} \frac{\sin(p\,\theta)}{r^p} \right) \right| \leq K \left(\frac{\sqrt{2}}{3} \right)^P . \tag{4.16}$$

Choosing

$$P \sim \log(\varepsilon)/\log(\sqrt{2}/3)$$

we see that the approximation error remains bounded by ε. [Add details.]

We refer to the truncated vector $\hat{\mathbf{q}} = [\hat{q}_j]_{j=0}^{2P}$ as the *multipole expansion* of the charge distribution. It contains all the information required to evaluate the potential to within precision ε at any point that is well-separated from Ω.

REMARK 4.1. *The multipole expansion can also represent continuum charge distributions. Given a charge distribution $q(\mathbf{y})$ in Ω, and a harmonic potential*

$$u(\mathbf{x}) = \int_\Omega \log|\mathbf{x}-\mathbf{y}|\, q(\mathbf{y})\, dA(\mathbf{y}),$$

we find that u satisfies (4.7) if we define the multipole coefficients $\hat{q}_j$ via the formulas

$$\begin{aligned}
\hat{q}_0 &= \int_\Omega q(\mathbf{y})\, dA(\mathbf{y}),\\
\hat{q}_{2p-1} &= \int_\Omega q(\mathbf{y})\, C_{2p-1}(\mathbf{y},\mathbf{y}_c)\, dA(\mathbf{y}) = -\int_\Omega q(\mathbf{y})\,(r')^p \frac{1}{p}\cos(p\,\theta')\, dA(\mathbf{y}),\\
\hat{q}_{2p} &= \int_\Omega q(\mathbf{y})\, C_{2p}(\mathbf{y},\mathbf{y}_c)\, dA(\mathbf{y}) = -\int_\Omega q(\mathbf{y})\,(r')^p \frac{1}{p}\sin(p\,\theta')\, dA(\mathbf{y}).
\end{aligned}$$

4.3. Separation of variables and the Outgoing Expansion

The multipole expansion in Section 4.2 provides an approximation of the kernel $G(\mathbf{x},\mathbf{y})$ of the form

$$G(\mathbf{x},\mathbf{y}) \approx \sum_{p=1}^{P} f_p(\mathbf{x})\, g_p(\mathbf{y}). \tag{4.17}$$

The sum in (4.17) is made up of a sum of products in which one factor depends only on $\mathbf{x}$ and one depends only on $\mathbf{y}$; we call such an approximation a *separation of variables*, or a *tensor product* approximation of the kernel. For the specific case when $G(\mathbf{x},\mathbf{y}) = \log|\mathbf{x}-\mathbf{y}|$, Lemma 4.2.1 provides the form of the functions, but similar expansions can be constructed for a wide range of kernels. Appendix ?? lists some known analytic expansions, and we describe in Chapter ? techniques for numerically constructing the functions.

We illustrate the value of a separation of variables such as (4.17) by a simple model problem. Suppose that for some given kernel G admitting an approximation (4.17), we seek to evaluate the sum

$$u_i = \sum_{j=1}^{N} G(\mathbf{x}_i,\, \mathbf{y}_j)\, q_j, \qquad i = 1,\, 2,\, \ldots,\, M, \tag{4.18}$$

where $\{q_j\}_{j=1}^N$ is a given set of *sources*, and we week to evaluate the *potentials* $\{u_i\}_{i=1}^M$. We suppose that the *source locations* $\{\mathbf{y}_j\}_{j=1}^N$ are contained in a box Ω_τ that is separate from a box Ω_σ that holds the *target locations* $\{\mathbf{x}_i\}_{i=1}^M$, see Figure ?. Using straight-forward summation, evaluating (4.18) requires $\mathrm{O}(M\,N)$ operations. Now suppose that we replace the kernel in (4.18) with the approximation (4.17) and seek to compute the numbers

$$u_i^{\text{approx}} = \sum_{j=1}^{N} \left(\sum_{p=1}^{P} f_p(\mathbf{x}_i)\, g_p(\mathbf{y}_j) \right) q_j.$$

We observe that by interchanging the summation order, we can evaluate the vector $\{u_i^{\text{approx}}\}_{i=1}^M$ in two steps: We first evaluate the numbers

$$\hat{q}_p = \sum_{j=1}^{N} q_j\, g_p(\mathbf{y}_j), \qquad p = 1,\, 2,\, \ldots,\, P, \tag{4.19}$$

at cost $\mathrm{O}(P\,N)$, and then evaluate

$$u_i^{\text{approx}} = \sum_{p=1}^{P} \hat{q}_p\, f_p(\mathbf{x}), \qquad i = 1,\, 2,\, \ldots,\, M, \tag{4.20}$$

at cost $\mathrm{O}(P\,M)$. We see that the cost is now $\mathrm{O}(P\,(M+N))$, instead of $\mathrm{O}(M\,N)$. If P is significantly smaller than M and N, the savings can be substantial. The number of terms P controls the accuracy: Higher P leads to a lower accuracy, but a higher cost.

The vector $\hat{\mathbf{q}} = [\hat{q}_p]_{p=1}^{P}$ is the *outgoing expansion* of the box τ. The potential generated by the charges in τ can (to a given precision) be constructed from the information contained in $\hat{\mathbf{q}}_\tau$ alone at any point sufficiently far removed from the source box. In this book, the condition "sufficiently far removed" will almost always mean that every point in the target box is *well-separated* from the source box in the sense described in Definition 4.1.

REMARK 4.2. *It will be convenient to express the various operations performed using matrix notation. The sum (4.18) is simply a matrix-vector multiplication*

$$\mathbf{u}_\sigma = \mathbf{A}_{\sigma,\tau}\, \mathbf{q}_\tau,$$

where $\mathbf{A}_{\sigma,\tau}$ *is the* $M \times N$ *matrix whose* ij *entry is given by*

$$\mathbf{A}_{\sigma,\tau}(i,j) = G(\mathbf{x}_i, \mathbf{y}_j).$$

Lemma 4.2.1 implies that $\mathbf{A}_{\sigma,\tau}$ *admits a factorization*

$$\underset{M \times N}{\mathbf{A}_{\sigma,\tau}} \approx \underset{M \times P}{\tilde{\mathbf{B}}_{\sigma,\tau}} \underset{P \times N}{\mathbf{C}_\tau},$$

where $\tilde{\mathbf{B}}_{\sigma,\tau}$ *is an* $M \times P$ *matrix whose* ip *entry is given by*

$$\tilde{\mathbf{B}}_{\sigma,\tau}(i,p) = B_p(\mathbf{x}_i, \mathbf{y}_\tau).$$

Analogously, $\mathbf{C}_\tau$ *is an* $P \times N$ *matrix whose* pj *entry is given by*

$$\mathbf{C}_\tau(p,j) = C_p(\mathbf{y}_\tau, \mathbf{y}_j).$$

The acceleration is now obtained by first evaluating

$$\hat{\mathbf{q}}_\tau = \mathbf{C}_\tau\, \mathbf{q}_\tau$$

at a cost of $N\,R$ *operations, and then evaluating*

$$\mathbf{u}_\sigma^{\text{approx}} = \tilde{\mathbf{B}}_{\sigma,\tau}\, \hat{\mathbf{q}}_\tau,$$

at a cost of $M\,R$ *operations. Then clearly*

$$\mathbf{u}_\sigma^{\text{approx}} = \tilde{\mathbf{B}}_{\sigma,\tau}\, \hat{\mathbf{q}}_\tau = \tilde{\mathbf{B}}_{\sigma,\tau}\, \mathbf{C}_\tau\, \mathbf{q}_\tau \approx \mathbf{A}_{\sigma,\tau}\, \mathbf{q}_\tau = \mathbf{u}_\sigma.$$

4.4. A basic tree structure

The separation of variables in the kernel that was described in Section 4.3 is as we saw sufficient for effectively evaluate a potential field whenever the set of target locations is *well-separated* from the set of source locations. When the two sets are identical, we need to tessellate the box containing them into smaller boxes, and use the expansion only for interactions between boxes that are. In this section, we describe the simplest such tessellation.

Suppose that we are given a set of points $\{\mathbf{x}_i\}_{i=1}^{N}$ in a box Ω. Given an integer n_0, we pick the smallest integer L such that when the box Ω is split into 4^L equisized smaller boxes, no box holds

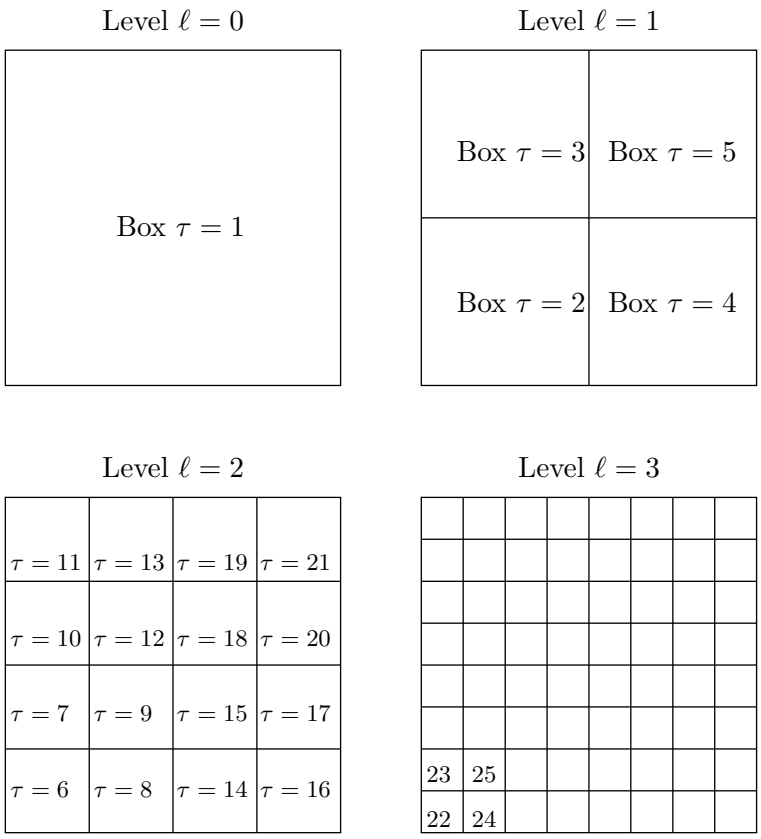


FIGURE 4.2. A binary tree with 4 levels of uniform refinement.

more than n_0 points. These 4^L equisized small boxes form the *leaves* of the tree. We merge the leaves by sets of four to form 4^{L-1} boxes of twice the side-length, and then continue merging by pairs until we recover the original box Ω, which we call the *root*.

The set consisting of all boxes of the same size forms what we call a *level*. We label the levels using the integer $\ell = 0, 1, 2, \ldots, L$, with $\ell = 0$ denoting the root, and $\ell = L$ denoting the leaves. See Figure 4.2.

DEFINITION 4.2. Let τ be a box in a hierarchical tree.

- The parent of τ is the box on the next coarser level that contains τ.
- The ancestors of τ is the set $\mathcal{L}_\tau^{\text{anc}}$ consisting of the parent of τ, the parent of the parent, *etc.*
- The children of τ is the set $\mathcal{L}_\tau^{\text{child}}$ of boxes whose parent is τ.
- The neighbors of τ is the set $\mathcal{L}_\tau^{\text{nei}}$ of boxes that are on the same level as τ and are directly adjacent to it.
- The interaction list of τ is the set $\mathcal{L}_\tau^{\text{int}}$ of all boxes σ such that:
  (1) σ and τ are on the same level.
  (2) σ and τ are not directly adjacent.
  (3) The parents of σ and τ are directly adjacent.

EXAMPLE 4.1. For the tree shown in Figure 5.1, we have, *e.g.*,

$$\begin{aligned}
\mathcal{L}_{56}^{\text{anc}} &= \{1,\, 4,\, 14\},\\
\mathcal{L}_{14}^{\text{child}} &= \{54,\, 55,\, 56,\, 57\},\\
\mathcal{L}_{23}^{\text{nei}} &= \{22,\, 24,\, 25,\, 26,\, 28\},\\
\mathcal{L}_{59}^{\text{nei}} &= \{36,\, 37,\, 48,\, 58,\, 60,\, 61,\, 70,\, 72\},\\
\mathcal{L}_{7}^{\text{int}} &= \{11, 13, 14\colon 21\},\\
\mathcal{L}_{37}^{\text{int}} &= \{22\colon 29,\, 30\colon 33,\, 38\colon 41,\, 47,\, 49,\, 54\colon 57,\, 60,\, 61,\, 71,\, 72,\, 73\}.
\end{aligned}$$

For the moment, we are assuming that the given point distribution is sufficiently uniform that all the leaves hold roughly the same number of points. In this case, one easily sees that

$$L \sim \log \frac{N}{n_0}.$$

For non-uniform distributions of points, a uniform subdivision of Ω into 4^L boxes of equal length would be inefficient since many of the leaves would hold few or no points. In such cases, adaptive subdivisions should be used, see Chapter ?. This chapter also contains a description of algorithms for determining all lists.

4.5. The Barnes-Hut Method

In this section, we describe a relatively simple scheme for rapidly evaluating the sum (4.1). This is a harder problem than the simple one considered in Section 4.3 since the sets of source points and target points are not well-separated (in fact, they are identical). The scheme uses only the simple multipole expansions described in Section 4.2 to reduce the asymptotic complexity of evaluating (4.1) from $\mathrm{O}(N^2)$ to $\mathrm{O}(N \log N)$. The material in this section is included primarily for illustrative purposes since it forms a slight deviation from the main narrative of the chapter; the algorithm described is simple and somewhat useful in certain contexts, but is far less powerful than the full FMM, which will be described in Sections 4.6 and 4.7. That presentation is largely independent of the material in this section.

The first step of the Barnes-Hut algorithm is to organize the points into a hierarchical tree as described in Section 4.4. Then the potentials u_i are evaluated via three steps: (1) Compute the outgoing expansion for every box. (2) Evaluate all interactions between well-separated boxes from the outgoing expansions. (3) Evaluate the near-field interactions via direct computation.

We will describe the three steps in details, but first we introduce some notation. For each box τ, the index vector J_τ marks all points contained in the box τ. In other words:

$$j \in J_\tau \qquad \Leftrightarrow \qquad \mathbf{x}_j \in \Omega_\tau.$$

Next we split the potential into a far-field and a near-field part. To be precise, for a point $\mathbf{x}_i$ in a leaf box τ, we set

$$u_i = u_i^{\text{near}} + u_i^{\text{far}},$$

where u_i^{near} is the potential caused by charges that are not well-separated from τ.

Step 1 — compute the outgoing expansions: For each box τ in the tree, we compute its outgoing expansion $\hat{\mathbf{q}}_\tau \in \mathbb{R}^P$ via the formula

$$\hat{\mathbf{q}}_\tau(p) = \sum_{j \in J_\tau} C_p(\mathbf{y}_\tau, \mathbf{x}_j)\, q_j, \qquad p = 1,\, 2,\, \ldots,\, P. \tag{4.21}$$

(We recall that the information in $\hat{\mathbf{q}}_\tau$ is sufficient to evaluate the potential induced by the charges in τ *at all points that are well-separated from* τ.) The calculation (4.21) can be expressed using matrix notation as

$$\hat{\mathbf{q}}_\tau = \mathbf{C}_\tau\, \mathbf{q}(J_\tau), \tag{4.22}$$

where $\mathbf{C}_\tau$ is the matrix with entries $\mathbf{C}_\tau(p, j) = C_p(\mathbf{y}_\tau, \mathbf{x}_j)$.

Step 2 — evaluate long range interactions: Fix a leaf node τ, fix a point $\mathbf{x}_i \in \Omega_\tau$, and let τ_L, τ_{L-1}, $\ldots$, τ_2 be an enumeration of the ancestors of τ so that $\tau_L = \tau$, and $\tau_{\ell-1}$ is the parent of τ_ℓ. Now observe that every charge $\mathbf{x}_j$ that is well-separated from τ is contained in the interaction list of precisely one of the ancestors of τ. Consequently,

$$u_i^{\text{far}} = \sum_{\ell=2}^{L} \sum_{\sigma \in \mathcal{L}_{\tau_\ell}} \sum_{j \in J_\sigma} G(\mathbf{x}_i, \mathbf{x}_j)\, q_j. \tag{4.23}$$

In (4.23), all kernel evaluations are performed for points in boxes that are well-separated. Consequently, we can to within precision ε replace them with multipole expansions,

$$\begin{aligned} u_i^{\text{far}} &= \sum_{\ell=2}^{L} \sum_{\sigma \in \mathcal{L}_{\tau_\ell}} \sum_{j \in J_\sigma} \sum_{p=1}^{P} \tilde{B}_p(\mathbf{x}_i, \hat{\mathbf{y}}_\sigma)\, C_p(\mathbf{x}_j, \hat{\mathbf{y}}_\sigma)\, q_j \\ &= \sum_{\ell=2}^{L} \sum_{\sigma \in \mathcal{L}_{\tau_\ell}} \sum_{p=1}^{P} \tilde{B}_p(\mathbf{x}_i, \hat{\mathbf{y}}_\sigma)\, \hat{\mathbf{q}}_\sigma(p). \end{aligned} \tag{4.24}$$

The expression (4.24) can be written compactly as

$$\mathbf{u}_\tau^{\text{far}} \approx \sum_{\ell=2}^{L} \sum_{\sigma \in \mathcal{L}_\tau} \tilde{\mathbf{B}}_{\tau,\sigma}\, \hat{\mathbf{q}}_\sigma,$$

where $\tilde{\mathbf{B}}_{\tau,\sigma}$ is the matrix with entries $\tilde{\mathbf{B}}_{\tau,\sigma}(i, p) = \tilde{B}_p(\mathbf{x}_i, \mathbf{y}_\sigma)$.

Step 3 — evaluate short range interactions: Fix a leaf node τ, and a point $\mathbf{x}_i \in \Omega_\tau$. The near field u_i^{near} has contributions from charges both in τ itself, and in its direct neighbors. Consequently,

$$u_i^{\text{near}} = \sum_{j \in J_\tau} G(\mathbf{x}_i, \mathbf{x}_j)\, q_j + \sum_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \sum_{j \in J_\sigma} G(\mathbf{x}_i, \mathbf{x}_j)\, q_j. \tag{4.25}$$

Using matrix notation, we express (4.25) compactly as

$$\mathbf{u}_\tau^{\text{near}} = \mathbf{A}(J_\tau, J_\tau)\, \mathbf{q}(J_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \mathbf{A}(J_\tau, J_\sigma)\, \mathbf{q}(J_\sigma).$$

In a final (fourth?) step, we simply loop over all leaf nodes τ, and combine the far field and the near field,

$$\mathbf{u}(J_\tau) = \mathbf{u}_\tau^{\text{far}} + \mathbf{u}_\tau^{\text{near}}.$$

Compute the outgoing expansions on all boxes:

loop over all boxes τ
    $\hat{\mathbf{q}}_\tau = \mathbf{C}_\tau \, \mathbf{q}(J_\tau)$
end loop

Evaluate the far field potentials. Each box τ broadcasts its outgoing representation to all boxes in its interaction list:

$\mathbf{u} = 0$
loop over all boxes τ
    loop over all $\sigma \in \mathcal{L}_\tau$
        $\mathbf{u}(J_\sigma) = \mathbf{u}(J_\sigma) + \tilde{\mathbf{B}}_{\sigma,\tau} \, \hat{\mathbf{q}}_\tau$
    end loop
end loop

Evaluate the near field interactions:

loop over all leaf boxes τ
    $\mathbf{u}(J_\tau) = \mathbf{u}(J_\tau) + \mathbf{A}(J_\tau, J_\tau) \, \mathbf{q}(J_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \mathbf{A}(J_\tau, J_\sigma) \, \mathbf{q}(J_\sigma)$
end

FIGURE 4.3. The Barnes-Hut algorithm.

The Barnes-Hut algorithm is summarized as Algorithm 4.3. For simplicity, the steps have been slightly rearranged from the presentation above, but it is easily seen that the output is exactly the same.

It only remains to estimate the cost of the Barnes-Hut algorithm. We let T_j denote the cost of Step j. Then

$$T_1 \sim N \, L \, P,$$

since in Step 1 each of the N particles broadcasts P pieces of information to L boxes. In Step 2, we need to evaluate the sum (4.24) for each of the N particles. There are at most 27 boxes in each list $\mathcal{L}_{\tau_\ell}$, so

$$T_2 \sim 27 \, N \, L \, P.$$

Finally, Step 3 involves the evaluation of the sum (4.25) for each of the N particles. Since each box τ has at most 8 neighbors, and since each box contains at most n_0 particles, we find that

$$T_3 \sim 9 \, N \, n_0.$$

Recalling that $L \sim \log(N/n_0)$, and that n_0 is a small fixed number (say $n_0 = 100$), we find that the total cost $T_{\text{total}} = T_1 + T_2 + T_3$ satisfies

$$T \sim N \, \log(N) \, \log(1/\varepsilon).$$

as $N \to \infty$ and $\varepsilon \to 0$.

4.6. Incoming Expansions and Translation Operators

A reason for the $\mathrm{O}(N \, \log(N))$ asymptotic cost of the Barnes-Hut algorithm is that every one of the N charges directly communicates its contribution to all of the $\mathrm{O}(\log(N))$ boxes that contain it. We can eliminate the $\log(N)$ factor in the cost of computing the outgoing expansions by computing them *recursively*, as follows:

On level L:: Compute the outgoing expansions directly using (4.21).

On level $\ell = L-1, L-2, \ldots, 2$:: Compute the outgoing expansion for each box τ on level ℓ by aggregating the outgoing expansions of the four children of τ on level $\ell+1$ using a so called *outgoing-to-outgoing translation operator.*

The potentials will be calculated via an analogous process in which *incoming expansions* (to be defined shortly) are aggregated in a dual sweep from coarser to finer levels. In this section, we introduce the incoming expansion, and describe a number of operators for shifting expansions from one box to another.

To introduce the concepts of an incoming expansion and a translation operator, let us return to the model problem described in Section 4.3 in which we were given two well-separated boxes Ω_σ and Ω_τ, a set of source points $\{\mathbf{y}_j\}_{j=1}^N$ with associated charges $\{q_j\}_{j=1}^N$ in Ω_τ and a set of target points $\{\mathbf{x}_i\}_{i=1}^M$ in Ω_σ. Our task is to rapidly evaluate the map

$$\mathbf{u}_\sigma = \mathbf{A}_{\sigma,\tau}\, \mathbf{q}_\tau, \tag{4.26}$$

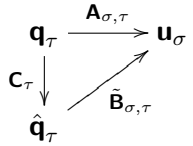
where the $M \times N$ matrix $\mathbf{A}_{\sigma,\tau}$ has entries

$$\mathbf{A}_{\sigma,\tau}(i,j) = G(\mathbf{x}_i, \mathbf{y}_j).$$

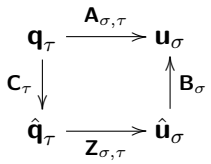
The purpose of Section 4.3 was to demonstrate that the map (4.26) could be evaluated rapidly via the approximate factorization

$$\mathbf{A}_{\sigma,\tau} \approx \tilde{\mathbf{B}}_{\sigma,\tau}\, \mathbf{C}_\tau,$$

where the $M \times P$ matrix $\tilde{\mathbf{B}}_{\sigma,\tau}$ and the $P \times N$ matrix $\mathbf{C}_\tau$ are defined in Remark 4.2. The process is illustrated by the following diagram:



We first evaluate the *outgoing expansion* $\hat{\mathbf{q}}$ via application of the map $\mathbf{C}_\tau$ at a cost of $N\,P$ operations. Then the potential on σ is evaluated via application of the map $\tilde{\mathbf{B}}_{\sigma,\tau}$ at a cost of $M\,P$ operations. In this section, we will add one more leg to the diagram:



(4.27)

The vector $\hat{\mathbf{u}}_\sigma$ is the *incoming expansion* on σ. In our model problem it will be a vector of P numbers $\{\hat{\mathbf{u}}_\sigma(p)\}_{p=1}^P$ such that

$$u_i \approx \hat{\mathbf{u}}_\sigma(0) + \sum_{r=1}^{P/2} \hat{\mathbf{u}}_\sigma(2p-1)\,(r')^p\,\cos(p\,\theta) + \hat{\mathbf{u}}_\sigma(2p)\,(r')^p\,\sin(p\,\theta).$$

In other words, the numbers $\hat{\mathbf{u}}_\sigma(p)$ are the coefficients in a harmonic expansion of u, and we see that the $M \times P$ matrix $\mathbf{B}_\sigma$ has coefficients

$$\mathbf{B}_\sigma(i,p) = \cdots$$

We can now state our goals for this section:

(1) Demonstrate that there exists a matrix $\mathbf{Z}$, called the "outgoing-to-incoming translation operator" such that $\mathbf{A} \approx \mathbf{B}\,\mathbf{Z}\,\mathbf{C}$.
(2) Demonstrate that if all the source points in τ are contained in a box ν that is contained in τ, and if the outgoing expansion $\hat{q}_\nu$ for ν is known, then there is a linear map $\mathbf{F}$ such that $\hat{q}_\tau \approx \mathbf{F}\,\hat{q}_\nu$. We call the map $\mathbf{F}$ the "outgoing-to-outgoing translation operator".
(3) Demonstrate that if all target points in σ are contained in a box ν that is fully contained in σ, and if the incoming expansion for σ is known, then there is a linear map $\mathbf{E}$ such that $\hat{u}_\nu \approx \mathbf{E}\,\hat{u}_\sigma$. We call the map $\mathbf{E}$ the "incoming-to-incoming translation operator".

We start with a theorem asserting the existence of the outgoing-to-incoming translation operator.

LEMMA 4.6.1. *Let Ω_σ and Ω_τ be two well-separated intervals on the line with centers $\mathbf{x}_\sigma$ and $\mathbf{y}_\tau$, respectively. Let $\{\mathbf{y}_j\}_{j=1}^N$ be a set of source points in Ω_τ, and let $\{\mathbf{x}_i\}_{i=1}^M$ be a set of target points in Ω_σ. Let $\mathbf{A}$ be the $M \times N$ matrix with entries*

$$\mathbf{A}_{ij} = |\mathbf{x}_i - \mathbf{y}_j|^{-1}.$$

Then $\mathbf{A}$ admits the approximate factorization $\mathbf{A} \approx \mathbf{B}\,\mathbf{Z}\,\mathbf{C}$ where

$$\begin{aligned}
\mathbf{B}_{is} &= (\mathbf{x}_i - \mathbf{x}_\sigma)^{s-1}, \\
\mathbf{Z}_{sr} &= (-1)^{s-1}\binom{r+s-2}{r-1}\frac{1}{|\mathbf{x}_\sigma - \mathbf{y}_\tau|\,(\mathbf{x}_\sigma - \mathbf{y}_\tau)^{r+s-2}}, \\
\mathbf{C}_{rj} &= (\mathbf{y}_j - \mathbf{y}_\tau)^{r-1}.
\end{aligned}$$

Specifically, for any i, j,

$$|\mathbf{A}_{ij} - [\mathbf{B}\,\mathbf{Z}\,\mathbf{C}]_{ij}| \leq \cdots (1/3)^R.$$

PROOF. It is sufficient to demonstrate that for any $x \in \Omega_\sigma$, and for any $\mathbf{y} \in \Omega_\tau$, we have

$$\left| \frac{1}{|\mathbf{x} - \mathbf{y}|} - \sum_{s=1}^{R} (\mathbf{x} - \mathbf{x}_\sigma)^{s-1} \sum_{r=1}^{R} \mathbf{Z}_{sr}\,(\mathbf{y} - \mathbf{y}_\tau)^{r-1} \right| \leq \cdots \left(\frac{1}{3}\right)^R.$$

To this end, note that

$$\frac{1}{|\mathbf{x} - \mathbf{y}|} = \frac{1}{|(\mathbf{x} - \mathbf{y}_\tau) - (\mathbf{y} - \mathbf{y}_\tau)|} = \frac{1}{|\mathbf{x} - \mathbf{y}_\tau|}\,\frac{1}{\left|1 - \frac{\mathbf{y} - \mathbf{y}_\tau}{x - \mathbf{y}_\tau}\right|}.$$

Since $|\mathbf{y} - \mathbf{y}_\tau| < |\mathbf{x} - \mathbf{y}_\tau|$, and

$$(1-t)^{-1} = \sum_{s=1}^{\infty} t^{s-1}, \qquad \text{whenever } |t| < 1,$$

we find that

$$\begin{aligned}
\frac{1}{|\mathbf{x} - \mathbf{y}|} &= \frac{1}{|\mathbf{x} - \mathbf{y}_\tau|} \sum_{r=1}^{\infty} \left(\frac{\mathbf{y} - \mathbf{y}_\tau}{x - \mathbf{y}_\tau}\right)^{r-1} \\
&= \sum_{r=1}^{\infty} \frac{1}{|\mathbf{x} - \mathbf{y}_\tau|(\mathbf{x} - \mathbf{y}_\tau)^{r-1}} (\mathbf{y} - \mathbf{y}_\tau)^{r-1}.
\end{aligned} \tag{4.28}$$

Set $\kappa = \mathrm{Sign}(\mathbf{x}_\sigma - \mathbf{y}_\tau)$. Then

$$\frac{1}{|\mathbf{x}-\mathbf{y}_\tau|(\mathbf{x}-\mathbf{y}_\tau)^{r-1}} = \kappa\,(\mathbf{x}-\mathbf{y}_\tau)^{-r} = \kappa\,\big((\mathbf{x}_\sigma-\mathbf{y}_\tau)-(\mathbf{x}_\sigma-x)\big)^{-r}$$
$$= \kappa\,(\mathbf{x}_\sigma-\mathbf{y}_\tau)^{-r}\left(1-\frac{\mathbf{x}_\sigma-x}{\mathbf{x}_\sigma-\mathbf{y}_\tau}\right)^{-r}.$$

Next we note that $|\mathbf{x}_\sigma - x| < |\mathbf{x}_\sigma - \mathbf{y}_\tau|$ and that

$$(1-t)^{-r} = \sum_{s=1}^{\infty}\binom{r+s-2}{r-1}t^{s-1} \qquad \text{whenever } |t|<1.$$

It follows that

$$\frac{1}{|\mathbf{x}-\mathbf{y}_\tau|(\mathbf{x}-\mathbf{y}_\tau)^{r-1}} = \kappa\,(\mathbf{x}_\sigma-\mathbf{y}_\tau)^{-r}\sum_{s=1}^{\infty}\binom{r+s-2}{r-1}\left(\frac{\mathbf{x}_\sigma-x}{\mathbf{x}_\sigma-\mathbf{y}_\tau}\right)^{s-1}$$
$$= \sum_{s=1}^{\infty}\underbrace{(-1)^{s-1}\binom{r+s-2}{r-1}\frac{\kappa}{(\mathbf{x}_\sigma-\mathbf{y}_\tau)^{r+s-1}}}_{=\mathbf{Z}_{sr}}(\mathbf{x}-\mathbf{x}_\sigma)^{r-1}. \tag{4.29}$$

Combining (4.28) and (4.29), and noting that since the sums are absolutely convergent, we can interchange the order of summation, we find that

$$\frac{1}{|\mathbf{x}-\mathbf{y}|} = \sum_{s=1}^{\infty}(\mathbf{x}-\mathbf{x}_\sigma)^{s-1}\sum_{r=1}^{\infty}\mathbf{Z}_{sr}\,(\mathbf{y}-\mathbf{y}_\tau)^{r-1}.$$

It remains to bound the errors introduced by truncation.
FILL IN DETAILS... □

Next we prove the existence of the "outgoing-to-outgoing translation operator":

LEMMA 4.6.2. *Let $\mathbf{y}^\tau$ and $\mathbf{y}_\nu$ denote two points on the line. Let $\{\mathbf{y}_j\}_{j=1}^N$ denote a number of source locations with associated charges $\{q_j\}_{j=1}^N$. Define for $r = 1, 2, \ldots, R$ the "outgoing expansions"*

$$\hat{q}_r^\tau = \sum_{j=1}^{N} q_j\,(\mathbf{y}_j-\mathbf{y}_\tau)^{r-1}, \qquad \textit{and} \qquad \hat{q}_r^\nu = \sum_{j=1}^{N} q_j\,(\mathbf{y}_j-\mathbf{y}_\nu)^{r-1}.$$

Let $\mathbf{F}$ denote the $R\times R$ lower triangular matrix defined by

$$\mathbf{F}_{rs} = \begin{cases} \binom{r-1}{s-1}(\mathbf{y}_\nu-\mathbf{y}_\tau)^{r-s}, & \text{when } s\le r,\\ 0 & \text{when } s>r.\end{cases}$$

Then

$$\hat{q}^\tau = \mathbf{F}\,\hat{q}^\nu. \tag{4.30}$$

PROOF. The equation (4.30) follows directly from the binomial identity:

$$\begin{aligned}\hat{q}_r^\tau = \sum_{j=1}^N q_j \left(\mathbf{y}_j - \mathbf{y}_\tau\right)^{r-1} &= \sum_{j=1}^N q_j \left[\left(\mathbf{y}_j - \mathbf{y}_\nu\right) + \left(\mathbf{y}_\nu - \mathbf{y}_\tau\right)\right]^{r-1} \\ &= \sum_{j=1}^N q_j \sum_{s=1}^r \binom{r-1}{s-1} \left(\mathbf{y}_j - \mathbf{y}_\nu\right)^{s-1} \left(\mathbf{y}_\nu - \mathbf{y}_\tau\right)^{r-s} \\ &= \sum_{s=1}^r \binom{r-1}{s-1} \hat{q}_s^\nu \left(\mathbf{y}_\nu - \mathbf{y}_\tau\right)^{r-s}.\end{aligned}$$

□

The construction of the incoming-to-incoming operator is almost entirely analogous:

LEMMA 4.6.3. *Let $\mathbf{x}_\sigma$ and $\mathbf{x}_\nu$ denote two points on the real line. Suppose that $\{\hat{u}_r^\sigma\}_{r=1}^R$ is a given vector of coefficients in a polynomial expansion around $\mathbf{x}_\sigma$. Define the $R \times R$ upper triangular matrix $\mathbf{E}$ via*

$$\mathbf{E}_{sr} = \begin{cases} \binom{r-1}{s-1} \left(\mathbf{x}_\nu - \mathbf{x}_\sigma\right)^{r-s}, & \text{when } r \geq s, \\ 0, & \text{when } r < s, \end{cases}$$

and define the vector $\hat{u}^\nu$ via

$$\hat{u}^\nu = \mathbf{E}\, \hat{u}^\sigma.$$

Then for any x,

$$\sum_{r=1}^R \hat{u}_r^\sigma \left(\mathbf{x} - \mathbf{x}_\sigma\right)^{r-1} = \sum_{s=1}^R \hat{u}_s^\nu \left(\mathbf{x} - \mathbf{x}_\nu\right)^{s-1}.$$

PROOF. A simple application of the binomial identity yields

$$\begin{aligned}\sum_{r=1}^R \hat{u}_r^\sigma \left(\mathbf{x} - \mathbf{x}_\sigma\right)^{r-1} &= \sum_{r=1}^R \hat{u}_r^\sigma \left[\left(\mathbf{x} - \mathbf{x}_\nu\right) + \left(\mathbf{x}_\nu - \mathbf{x}_\sigma\right)\right]^{r-1} \\ &= \sum_{r=1}^R \hat{u}_r^\sigma \sum_{s=1}^r \binom{r-1}{s-1} \left(\mathbf{x} - \mathbf{x}_\nu\right)^{s-1} \left(\mathbf{x}_\nu - \mathbf{x}_\sigma\right)^{r-s} \\ &= \sum_{s=1}^R \underbrace{\left\{ \sum_{r=s}^R \binom{r-1}{s-1} \left(\mathbf{x}_\nu - \mathbf{x}_\sigma\right)^{r-s} \hat{u}_r^\sigma \right\}}_{=\hat{u}_s^\nu} \left(\mathbf{x} - \mathbf{x}_\nu\right)^{s-1}.\end{aligned}$$

□

REMARK 4.3. *The manipulation of the kernel at the beginning of this subsection constitutes an analytic proof that the matrix $\mathbf{A}$ has approximate rank R. It also provides explicit formulas for the factors $\mathbf{B}$ and $\mathbf{C}$. As an alternative approach, one can construct the matrix $\mathbf{A}$ first, and then numerically derive factors $\mathbf{B}$ and $\mathbf{C}$ such that $||\mathbf{A} - \mathbf{B}\,\mathbf{C}|| \leq \varepsilon$. This approach has the advantage that detailed information about the kernel is not required à priori, and moreover, that the rank R used is optimal. (The analytic derivation above gives only an upper bound for the rank R — it does not say that other factorizations cannot do better.) Against these advantages stands the cost of first constructing the full matrix $\mathbf{A}$ at a cost of $M\,N$ kernel evaluations, and then decomposing it at a cost of $\mathrm{O}(M\,N\,R)$ floating point operations. This amount of work can be justified if the evaluation problem (4.18) is to be evaluated for many charge distributions $\{q_j\}_{j=1}^N$ for a given set*

of source and target locations. In practice, hybrid methods that exploit both analytic knowledge and numerical compression are frequently the method of choice.

4.7. The Classical FMM in Two Dimensions

In this section, we will describe modifications to the Barnes-Hut method described in Section 4.5 that improve the asymptotic CPU time requirement from $\mathrm{O}(N \log(N))$ to $\mathrm{O}(N)$. The problem considered is the same (fast evaluation of the sum (4.1)), and the basic methodology is very similar: we start with organizing the points in a tree structure, we construct outgoing expansions for all boxes, and evaluate all long-range interactions via these expansions. A key difference is how the outgoing expansions are computed. In the Barnes-Hut method, every point directly broadcasts its contribution to the $\mathrm{O}(\log(N))$ boxes that contain it. In contrast, in the FMM, each point contributes directly only to the outgoing expansions on the leaves. The outgoing expansions on the next coarser level are constructed from the outgoing expansions on the leaves. The process then continues upwards through the tree with the outgoing expansion for any box being computed from the outgoing expansions of its two children. This process reduces the cost of computing the outgoing expansions from $\mathrm{O}(N \log(N))$ to $\mathrm{O}(N)$. The analogous reduction in cost in computing the potentials is obtained by similarly aggregating "incoming expansions" for each box τ.

We start the detailed description of the method by describing the process by which the outgoing expansions are computed hierarchically. Using the "outgoing-to-outgoing translation operators" introduced in Section 4.6, this is straightforward. Let τ denote a non-leaf node whose children are listed in $\mathcal{L}_\tau^{\text{child}}$. Then Lemma 5.4.3 demonstrates that there exists matrices $\mathbf{F}_{\tau,\sigma}$ such that

$$\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\text{child}}} \mathbf{F}_{\tau,\sigma}\, \hat{\mathbf{q}}_\sigma. \tag{4.31}$$

In order to derive the corresponding formula for hierarchically computing the incoming expansions, we first define for each box τ its *far-field potential* u_τ^{far} as the part of the potential caused by all sources that are well-separated from τ:

$$u_\tau^{\text{far}}(\mathbf{x}) = \sum_{j \in J_\tau^{\text{far}}} G(\mathbf{x}, \mathbf{x}_j)\, q_j, \qquad \mathbf{x} \in \Omega_\tau,$$

where J_τ^{far} is an index vector marking the source points that are well-separated from τ:

$$j \in J_\tau^{\text{far}} \qquad \Leftrightarrow \qquad \mathbf{x}_j \text{ is well-separated from } \tau.$$

The *incoming expansion* for τ is then a vector $\hat{\mathbf{u}}^\tau = [\hat{u}_r^\tau]_{r=1}^R$ such that

$$u_\tau^{\text{far}}(\mathbf{x}) \approx \sum_{r=1}^{R} \hat{u}_r^\tau\, (\mathbf{x} - \mathbf{x}_\tau)^{r-1}.$$

Now observe that if ν is the parent of τ, then the far-field potential for τ has two terms:

$$u_\tau^{\text{far}}(\mathbf{x}) = u_\nu^{\text{far}}(\mathbf{x}) + \sum_{\sigma \in \mathcal{L}_\tau} \sum_{j \in J_\sigma} G(\mathbf{x}, \mathbf{x}_j)\, q_j. \tag{4.32}$$

The first term in (4.32) represents the contributions to the potential from all charges that are well-separated from both τ and its parent ν. The second term in (4.32) represents contributions from points that are well-separated from τ, but not from ν. According to Lemma 5.4.2, these contributions can all be evaluated from the outgoing expansions on the boxes in $\mathcal{L}_\tau$,

$$\sum_{\sigma \in \mathcal{L}_\tau} \sum_{j \in J_\sigma} G(\mathbf{x}, \mathbf{x}_j)\, q_j \approx \sum_{\sigma \in \mathcal{L}_\tau} \mathbf{B}_\tau\, \mathbf{Z}_{\tau,\sigma}\, \hat{\mathbf{q}}_\sigma.$$

Letting $\mathbf{E}_{\tau,\nu}$ denote the incoming-to-incoming translation operator defined by Lemma 5.4.4, we find that equation (4.32) can be expressed

$$\hat{\mathbf{u}}_\tau = \mathbf{E}_{\tau,\nu}\,\hat{\mathbf{u}}_\nu + \sum_{\sigma\in\mathcal{L}_\tau} \mathbf{Z}_{\tau,\sigma}\,\hat{\mathbf{q}}_\sigma.$$

The Fast Multipole Method is now obtained by modifying the Barnes-Hut method to use equations (4.31) and (4.7) to hierarchically compute all incoming and outgoing expansions. The full algorithm is given in Figure 4.4.

4.8. The General Structure of an FMM

4.9. The Classical FMM in Three Dimensions

4.9.1. Problem formulation.

4.9.2. Translation operators.

4.9.3. Tree structure.

4.9.4. The Algorithm.

4.10. FMMs for other Kernel Functions

4.10.1. Stokes Equation.

4.10.2. The equations of linear elasticity.

4.10.3. Helmholtz equation.

4.11. Bibliographical notes

Compute the outgoing expansion on each leaf:

loop over all leaf nodes τ
    $\hat{\mathbf{q}}_\tau = \mathbf{C}_\tau \, \mathbf{q}(J_\tau)$
end loop

Computing the outgoing expansions on all parents by merging the outgoing expansions of their children by applying the "outgoing-to-outgoing translation operators":

loop over levels ℓ, finer to coarser, $l = L : (-1) : 2$
    loop over all nodes τ on level ℓ
        $\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\text{child}}} \mathbf{F}_{\tau,\sigma} \, \hat{\mathbf{q}}_\sigma$
    end loop
end loop

For every box τ, collect the contribution to its incoming potential that is due to boxes on the same level as itself (that is, all boxes that are well-separated from it, but are not well-separated from its parent). This is done by applying the "outgoing-to-incoming translation operators" to the outgoing expansions for all boxes σ in the "interaction list" of box τ:

loop over all boxes τ
    $\hat{\mathbf{u}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau} \mathbf{Z}_{\tau,\sigma} \, \hat{\mathbf{q}}_\sigma$
end loop

Complete the computation of the incoming expansions for every box by adding to the contribution from boxes in its interaction list (computed in the previous step) the contribution from its parent. Practically, we loop over all parent boxes τ, and broadcast the incoming expansion of τ to its children σ_1 and σ_2 via the application of "incoming-to-incoming translation operators":

loop over levels ℓ, coarser to finer, $l = 2 : L$
    loop over all nodes τ on level ℓ
        loop over all children σ of τ
            $\hat{\mathbf{u}}_\sigma = \hat{\mathbf{u}}_\sigma + \mathbf{E}_{\sigma,\tau} \, \hat{\mathbf{u}}_\tau$.
        end loop
    end loop
end loop

Compute the potential on every leaf τ by expanding its incoming potential and then adding the contributions from charges in τ itself and its adjacent neighbors via direct evaluation:

loop over all leaf nodes τ
    $\mathbf{u}(J_\tau) = \mathbf{B}_\tau \, \hat{\mathbf{u}}_\tau + \mathbf{A}(J_\tau, J_\tau) \, \mathbf{q}(J_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \mathbf{A}(J_\tau, J_\sigma) \, \mathbf{q}(J_\sigma)$
end loop

FIGURE 4.4. The Classical Fast Multipole Method in 2D.

Compute the outgoing expansion on each leaf via application of the *particle-to-outgoing operator:*

loop over all leaf nodes τ
    $\hat{\mathbf{q}}_\tau = \mathbf{C}_\tau \, \mathbf{q}(J_\tau)$
end loop

Compute the outgoing expansion of each parent by merging the expansions of its children via application of the *outgoing-to-outgoing operators:*

loop over levels ℓ, from the finest to the coarsest
    loop over all nodes τ on level ℓ
        $\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\text{child}}} \mathbf{F}_{\tau,\sigma} \, \hat{\mathbf{q}}_\sigma$
    end loop
end loop

Compute the incoming expansion for every box by combining the incoming expansion of its parent (mapped via the *incoming-to-incoming operator*) with the contributions from the outgoing expansions of all boxes in its interaction list (transferred via the *outgoing-to-incoming operator*):

$\hat{\mathbf{u}}_1 = \mathbf{0}$
loop over levels ℓ, from level $\ell = 1$ to the finest
    loop over all nodes τ on level ℓ
        Let ν denote the parent of τ.
        $\hat{\mathbf{u}}_\tau = \mathbf{E}_{\tau,\nu} \, \hat{\mathbf{u}}_\nu + \sum_{\sigma \in \mathcal{L}_\tau} \mathbf{Z}_{\tau,\sigma} \, \hat{\mathbf{q}}_\sigma$.
    end loop
end loop

Compute the potential on every leaf by expanding its incoming potential (via the *incoming-to-particle operator*) and then adding the contributions from its near field via direct evaluation:

loop over all leaf nodes τ
    $\mathbf{u}(J_\tau) = \mathbf{B}_\tau \, \hat{\mathbf{u}}_\tau + \mathbf{A}(J_\tau, J_\tau) \, \mathbf{q}(J_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \mathbf{A}(J_\tau, J_\sigma) \, \mathbf{q}(J_\sigma)$
end loop

FIGURE 4.5. The Classical Fast Multipole Method.

AAA

CHAPTER 5

A Linear Algebraic Interpretation of the FMM

5.1. Introduction

This chapter provides an alternative derivation of both the Barnes-Hut and the Fast Multipole Methods. These derivations are focused on expressing the algorithms as matrix factorizations. The chapter is self-contained, and overlaps the presentation in Chapter 4. *[Excessively so at the moment, this will be fixed later.]* However, our treatment in this chapter is quite terse. A reader unfamiliar with FMMs is encouraged to read Chapter 4 first.

To enable us to explicitly show the structure of certain blocks matrices, we consider a problem in one dimension in this chapter. However, the presentation directly generalizes to higher dimensions.

5.2. Problem formulation

In this section, we describe fast methods for rapidly evaluating the sum

$$u_i = \sum_{j=1}^{N} G(\mathbf{x}_i, \mathbf{x}_j)\, q_j, \qquad i = 1, 2, \ldots, N, \tag{5.1}$$

where $\{\mathbf{x}_i\}_{i=1}^N$ is a given set of real numbers, $\{q_i\}_{i=1}^N$ is a given set of real numbers which we call *charges*, $\{u_i\}_{i=1}^N$ is a set of real numbers we seek to compute called *potentials*, and the *kernel* G is given by

$$G(\mathbf{x}, \mathbf{y}) = \begin{cases} \frac{1}{|\mathbf{x}-\mathbf{y}|}, & \text{when } \mathbf{x} \neq y \\ 0 & \text{when } \mathbf{x} = y. \end{cases} \tag{5.2}$$

Using straight-forward summation techniques, the evaluation of the sum (4.1) requires $\mathrm{O}(N^2)$ floating point operations.

It is convenient to express the summation operation in (5.1) as a matrix-vector multiplication. To this end, we define $\mathbf{A}$ as the $M \times N$ matrix with ij entry

$$a_{ij} = G(\mathbf{x}_i, \mathbf{x}_j).$$

Setting $\mathbf{q} = [q_j]_{j=1}^N$ and $\mathbf{u} = [u_i]_{i=1}^M$, formula (5.1) can be expressed

$$\mathbf{u} = \mathbf{A}\,\mathbf{q}.$$

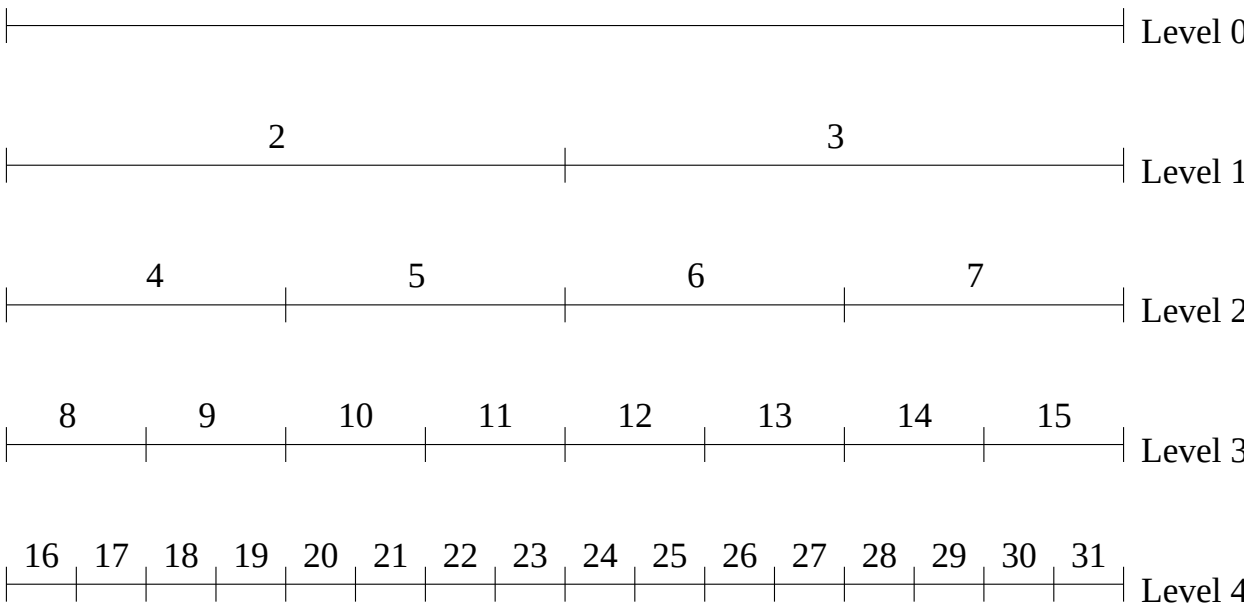


FIGURE 5.1. A binary tree with 4 levels of uniform refinement.

5.3. Hierarchical subdivisions of the line

Suppose that we are given a set of points $\{\mathbf{x}_i\}_{i=1}^N$ in an interval, or *box*, Ω on the line. In this section, we describe the most basic way of organizing these points into a hierarchical structure of subintervals: Given an integer n_0, we pick the smallest integer L such that when the given interval is subdivided into 2^L subintervals of equal length, no subinterval holds more than n_0 points. These 2^L subintervals form the *leaves* of the tree. We next merge the leaves by pairs to form 2^{L-1} intervals of twice the size, and then continue merging by pairs until we recover the full interval, which we call the *root.*

A set consisting of all boxes of the same size forms what we call a *level.* We use the index ℓ to number the levels so that level $\ell = L$ consists of all the leaves, and level $\ell = 0$ is the root box. See Figure 5.1.

The terms *parent*, *child*, *neighbor*, *etc* are defined as in Definition 4.2.

EXAMPLE 5.1. For the tree shown in Figure 5.1, we have, *e.g.*,

$$\mathcal{L}_2^{\text{child}} = \{4,\, 5\}, \quad \mathcal{L}_{14}^{\text{child}} = \{28,\, 29\},$$

$$\mathcal{L}_{10}^{\text{anc}} = \{1,\, 2,\, 5\}, \quad \mathcal{L}_{29}^{\text{anc}} = \{1,\, 3,\, 7,\, 14\}.$$

$$\mathcal{L}_2^{\text{nei}} = \{3\}, \quad \mathcal{L}_{10}^{\text{nei}} = \{9,\, 11\}, \quad \mathcal{L}_{17}^{\text{nei}} = \{16,\, 18\}.$$

$$\mathcal{L}_4^{\text{int}} = \{6,\, 7\}, \quad \mathcal{L}_{13}^{\text{int}} = \{10,\, 11,\, 15\}, \quad \mathcal{L}_{17}^{\text{int}} = \{19\},$$

Algorithms for constructing the various lists are given in Chapter ??. This chapter also describes techniques for handling non-uniform point distributions.

5.4. Translation operators

5.4.1. Problem formulation. Let us consider a simplified model problem in which we are given two separated sets, one containing charges, and one containing target points, and we seek to evaluate the potential caused by the charges at all the target points. Specifically, suppose that we are given a set of charge locations $\{\mathbf{y}_j\}_{j=1}^N$ in an interval Ω_τ and a set of target locations $\{\mathbf{x}_i\}_{i=1}^M$ in a set Ω_σ. Given a set of charges $\{q_j\}_{j=1}^N$ we now seek to evaluate the sum

$$u_i = \sum_{j=1}^{N} G(\mathbf{x}_i, \mathbf{y}_j)\, q_j. \tag{5.3}$$

In matrix notation, we set $\mathbf{q} = [q_j]_{j=1}^N$, $\mathbf{u} = [u_i]_{i=1}^M$, whence the sum (5.3) takes the form

$$\mathbf{u} = \mathbf{A}\,\mathbf{q}, \tag{5.4}$$

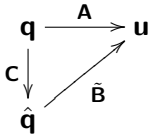
where the $M \times N$ matrix $\mathbf{A}$ has entries

$$a_{ij} = G(\mathbf{x}_i, \mathbf{y}_j).$$

5.4.2. Outgoing expansion. We will first demonstrate that the matrix $\mathbf{A}$ admits an approximate factorization

$$\mathbf{A} \approx \tilde{\mathbf{B}}\,\mathbf{C}, \tag{5.5}$$

where the product on the right hand side has rank R. The factorization (5.5) allows us to rapidly evaluate (5.4) via the diagram



LEMMA 5.4.1. *Let Ω_σ and Ω_τ be two separated intervals on the line with centers $\mathbf{x}_\sigma$ and $\mathbf{y}_\tau$. Suppose that $\mathbf{x} \in \Omega_\sigma$ and $\mathbf{y} \in \Omega_\tau$, and that the kernel G is defined by (5.2). Let R be a positive integer. Set*

$$\tilde{B}_r(\mathbf{x}, \mathbf{y}_\tau) = \frac{1}{|\mathbf{x} - \mathbf{y}_\tau|\,(\mathbf{x} - \mathbf{y}_\tau)^{r-1}}, \qquad \text{and} \qquad C_r(\mathbf{y}_\tau, \mathbf{y}) = (\mathbf{y}_j - \mathbf{y}_\tau)^{r-1}.$$

Then

$$\left| G(\mathbf{x}, \mathbf{y}) - \sum_{r=1}^{R} B_r(\mathbf{x}, \mathbf{y}_\tau)\, C_r(\mathbf{y}_\tau, \mathbf{y}) \right| \leq C\,\alpha^R,$$

were $C = \cdots$ and

$$\alpha = \frac{|\mathbf{y} - \mathbf{y}_\tau|}{|\mathbf{x} - \mathbf{x}_\tau|}.$$

Now just set

$$\tilde{\mathbf{B}}_{ir} = \frac{1}{|\mathbf{x}_i - \mathbf{y}_\tau|\,(\mathbf{x}_i - \mathbf{y}_\tau)^{r-1}}, \qquad \text{and} \qquad \mathbf{C}_{rj} = (\mathbf{y}_j - \mathbf{y}_\tau)^{r-1},$$

and observe that if Ω_σ and Ω_τ are well-separated, then $\alpha \leq 1/3$.

5.4.3. The incoming expansion. We next add one more leg to the diagram:

(5.6)
$$\begin{array}{ccc} \mathbf{q} & \xrightarrow{\mathbf{A}} & \mathbf{u} \\ {\scriptstyle \mathbf{C}}\downarrow & & \uparrow{\scriptstyle \mathbf{B}} \\ \hat{\mathbf{q}} & \xrightarrow[\mathbf{Z}]{} & \hat{\mathbf{u}} \end{array}$$

The vector $\hat{\mathbf{u}}$ is the *incoming expansion* on σ. In our model problem it will be a vector of R numbers $\{\hat{u}_r\}_{r=1}^R$ such that

$$u_i \approx \sum_{r=1}^{R} \hat{u}_r \, (\mathbf{x}_i - \mathbf{x}_\sigma)^{r-1}.$$

In other words, the numbers $\hat{u}_r$ are the coefficients in a polynomial expansion of u, and we see that the $M \times R$ matrix $\mathbf{B}$ has coefficients

$$\mathbf{B}_{ir} = (\mathbf{x}_i - \mathbf{x}_\sigma)^{r-1}.$$

We can now state our goals for this section:

We can alternatively formulat (5.6) as a matrix approximation as

$$\mathbf{A} \approx \mathbf{B}\,\mathbf{Z}\,\mathbf{C}.$$

The existence of such a factorization is assured by the following result:

LEMMA 5.4.2. *Let Ω_σ and Ω_τ be two well-separated intervals on the line with centers $\mathbf{x}_\sigma$ and $\mathbf{y}_\tau$, respectively. Let $\{\mathbf{y}_j\}_{j=1}^N$ be a set of source points in Ω_τ, and let $\{\mathbf{x}_i\}_{i=1}^M$ be a set of target points in Ω_σ. Let $\mathbf{A}$ be the $M \times N$ matrix with entries*

$$\mathbf{A}_{ij} = |\mathbf{x}_i - \mathbf{y}_j|^{-1}.$$

Then $\mathbf{A}$ admits the approximate factorization $\mathbf{A} \approx \mathbf{B}\,\mathbf{Z}\,\mathbf{C}$ where

$$\begin{aligned} \mathbf{B}_{is} &= (\mathbf{x}_i - \mathbf{x}_\sigma)^{s-1}, \\ \mathbf{Z}_{sr} &= (-1)^{s-1} \binom{r+s-2}{r-1} \frac{1}{|\mathbf{x}_\sigma - \mathbf{y}_\tau| \, (\mathbf{x}_\sigma - \mathbf{y}_\tau)^{r+s-2}}, \\ \mathbf{C}_{rj} &= (\mathbf{y}_j - \mathbf{y}_\tau)^{r-1}. \end{aligned}$$

Specifically, for any i, j,

$$|\mathbf{A}_{ij} - [\mathbf{B}\,\mathbf{Z}\,\mathbf{C}]_{ij}| \leq \cdots (1/3)^R.$$

PROOF. It is sufficient to demonstrate that for any $x \in \Omega_\sigma$, and for any $\mathbf{y} \in \Omega_\tau$, we have

$$\left| \frac{1}{|\mathbf{x} - \mathbf{y}|} - \sum_{s=1}^{R} (\mathbf{x} - \mathbf{x}_\sigma)^{s-1} \sum_{r=1}^{R} \mathbf{Z}_{sr} \, (\mathbf{y} - \mathbf{y}_\tau)^{r-1} \right| \leq \cdots \left(\frac{1}{3}\right)^R.$$

To this end, note that

$$\frac{1}{|\mathbf{x} - \mathbf{y}|} = \frac{1}{|(\mathbf{x} - \mathbf{y}_\tau) - (\mathbf{y} - \mathbf{y}_\tau)|} = \frac{1}{|\mathbf{x} - \mathbf{y}_\tau|} \, \frac{1}{\left|1 - \frac{\mathbf{y} - \mathbf{y}_\tau}{x - \mathbf{y}_\tau}\right|}.$$

Since $|\mathbf{y} - \mathbf{y}_\tau| < |\mathbf{x} - \mathbf{y}_\tau|$, and

$$(1-t)^{-1} = \sum_{s=1}^{\infty} t^{s-1}, \qquad \text{whenever } |t| < 1,$$

we find that

$$\frac{1}{|\mathbf{x}-\mathbf{y}|} = \frac{1}{|\mathbf{x}-\mathbf{y}_\tau|}\sum_{r=1}^{\infty}\left(\frac{\mathbf{y}-\mathbf{y}_\tau}{x-\mathbf{y}_\tau}\right)^{r-1} \tag{5.7}$$
$$= \sum_{r=1}^{\infty}\frac{1}{|\mathbf{x}-\mathbf{y}_\tau|(\mathbf{x}-\mathbf{y}_\tau)^{r-1}}(\mathbf{y}-\mathbf{y}_\tau)^{r-1}.$$

Set $\kappa = \text{Sign}(\mathbf{x}_\sigma - \mathbf{y}_\tau)$. Then

$$\frac{1}{|\mathbf{x}-\mathbf{y}_\tau|(\mathbf{x}-\mathbf{y}_\tau)^{r-1}} = \kappa\,(\mathbf{x}-\mathbf{y}_\tau)^{-r} = \kappa\left((\mathbf{x}_\sigma-\mathbf{y}_\tau)-(\mathbf{x}_\sigma-x)\right)^{-r}$$
$$= \kappa\,(\mathbf{x}_\sigma-\mathbf{y}_\tau)^{-r}\left(1-\frac{\mathbf{x}_\sigma-x}{\mathbf{x}_\sigma-\mathbf{y}_\tau}\right)^{-r}.$$

Next we note that $|\mathbf{x}_\sigma - x| < |\mathbf{x}_\sigma - \mathbf{y}_\tau|$ and that

$$(1-t)^{-r} = \sum_{s=1}^{\infty}\binom{r+s-2}{r-1}t^{s-1} \qquad \text{whenever } |t|<1.$$

It follows that

$$\frac{1}{|\mathbf{x}-\mathbf{y}_\tau|(\mathbf{x}-\mathbf{y}_\tau)^{r-1}} = \kappa\,(\mathbf{x}_\sigma-\mathbf{y}_\tau)^{-r}\sum_{s=1}^{\infty}\binom{r+s-2}{r-1}\left(\frac{\mathbf{x}_\sigma-x}{\mathbf{x}_\sigma-\mathbf{y}_\tau}\right)^{s-1} \tag{5.8}$$
$$= \sum_{s=1}^{\infty}\underbrace{(-1)^{s-1}\binom{r+s-2}{r-1}\frac{\kappa}{(\mathbf{x}_\sigma-\mathbf{y}_\tau)^{r+s-1}}}_{=\mathbf{Z}_{sr}}(\mathbf{x}-\mathbf{x}_\sigma)^{r-1}.$$

Combining (4.28) and (4.29), and noting that since the sums are absolutely convergent, we can interchange the order of summation, we find that

$$\frac{1}{|\mathbf{x}-\mathbf{y}|} = \sum_{s=1}^{\infty}(\mathbf{x}-\mathbf{x}_\sigma)^{s-1}\sum_{r=1}^{\infty}\mathbf{Z}_{sr}\,(\mathbf{y}-\mathbf{y}_\tau)^{r-1}.$$

It remains to bound the errors introduced by truncation.
FILL IN DETAILS... □

Next we prove the existence of the "outgoing-to-outgoing translation operator":

LEMMA 5.4.3. *Let $\mathbf{y}_\tau$ and $\mathbf{y}_\nu$ denote two points on the line. Let $\{\mathbf{y}_j\}_{j=1}^N$ denote a number of source locations with associated charges $\{q_j\}_{j=1}^N$. Define for $r = 1, 2, \ldots, R$ the "outgoing expansions"*

$$\hat{q}_r^\tau = \sum_{j=1}^{N} q_j\,(\mathbf{y}_j-\mathbf{y}_\tau)^{r-1}, \qquad \textit{and} \qquad \hat{q}_r^\nu = \sum_{j=1}^{N} q_j\,(\mathbf{y}_j-\mathbf{y}_\nu)^{r-1}.$$

Let $\mathbf{F}$ denote the $R\times R$ lower triangular matrix defined by

$$\mathbf{F}_{rs} = \begin{cases} \binom{r-1}{s-1}(\mathbf{y}_\nu-\mathbf{y}_\tau)^{r-s}, & \text{when } s\le r,\\ 0 & \text{when } s>r. \end{cases}$$

Then

$$\hat{q}^\tau = \mathbf{F}\,\hat{q}^\nu. \tag{5.9}$$

PROOF. The equation (5.9) follows directly from the binomial identity:

$$\begin{aligned}\hat{q}_r^\tau = \sum_{j=1}^N q_j \left(\mathbf{y}_j - \mathbf{y}_\tau\right)^{r-1} &= \sum_{j=1}^N q_j \left[\left(\mathbf{y}_j - \mathbf{y}_\nu\right) + \left(\mathbf{y}_\nu - \mathbf{y}_\tau\right)\right]^{r-1} \\ &= \sum_{j=1}^N q_j \sum_{s=1}^r \binom{r-1}{s-1} \left(\mathbf{y}_j - \mathbf{y}_\nu\right)^{s-1} \left(\mathbf{y}_\nu - \mathbf{y}_\tau\right)^{r-s} \\ &= \sum_{s=1}^r \binom{r-1}{s-1} \hat{q}_s^\nu \left(\mathbf{y}_\nu - \mathbf{y}_\tau\right)^{r-s}.\end{aligned}$$

□

The construction of the incoming-to-incoming operator is almost entirely analogous:

LEMMA 5.4.4. *Let $\mathbf{x}_\sigma$ and $\mathbf{x}_\nu$ denote two points on the real line. Suppose that $\{\hat{u}_r^\sigma\}_{r=1}^R$ is a given vector of coefficients in a polynomial expansion around $\mathbf{x}_\sigma$. Define the $R \times R$ upper triangular matrix $\mathbf{E}$ via*

$$\mathbf{E}_{sr} = \begin{cases} \binom{r-1}{s-1} \left(\mathbf{x}_\nu - \mathbf{x}_\sigma\right)^{r-s}, & \text{when } r \geq s, \\ 0, & \text{when } r < s, \end{cases}$$

and define the vector $\hat{u}^\nu$ via

$$\hat{u}^\nu = \mathbf{E}\, \hat{u}^\sigma.$$

Then for any x,

$$\sum_{r=1}^R \hat{u}_r^\sigma \left(\mathbf{x} - \mathbf{x}_\sigma\right)^{r-1} = \sum_{s=1}^R \hat{u}_s^\nu \left(\mathbf{x} - \mathbf{x}_\nu\right)^{s-1}.$$

PROOF. A simple application of the binomial identity yields

$$\begin{aligned}\sum_{r=1}^R \hat{u}_r^\sigma \left(\mathbf{x} - \mathbf{x}_\sigma\right)^{r-1} &= \sum_{r=1}^R \hat{u}_r^\sigma \left[\left(\mathbf{x} - \mathbf{x}_\nu\right) + \left(\mathbf{x}_\nu - \mathbf{x}_\sigma\right)\right]^{r-1} \\ &= \sum_{r=1}^R \hat{u}_r^\sigma \sum_{s=1}^r \binom{r-1}{s-1} \left(\mathbf{x} - \mathbf{x}_\nu\right)^{s-1} \left(\mathbf{x}_\nu - \mathbf{x}_\sigma\right)^{r-s} \\ &= \sum_{s=1}^R \underbrace{\left\{ \sum_{r=s}^R \binom{r-1}{s-1} \left(\mathbf{x}_\nu - \mathbf{x}_\sigma\right)^{r-s} \hat{u}_r^\sigma \right\}}_{=\hat{u}_s^\nu} \left(\mathbf{x} - \mathbf{x}_\nu\right)^{s-1}.\end{aligned}$$

□

5.5. Barnes-Hut

In this section, we derive the Barnes-Hut algorithm as a multi-level matrix factorization.

Fix a level ℓ in the hierarchical tree. The index sets J_τ on level ℓ form a disjoint partitioning of the full index set $[1, 2, \ldots, N]$. We let $\underline{\mathbf{D}}^{(\ell)}$ denote a block tri-diagonal matrix corresponding to

this partitioning. The non-zero blocks of $\underline{\mathbf{D}}^{(\ell)}$ are identical to the corresponding blocks of $\mathbf{A}$.. To be precise, $\underline{\mathbf{D}}^{(\ell)}$ is an $N \times N$ matrix whose entries are all zeros, except that for any τ on level ℓ,

$$\underline{\mathbf{D}}^{(\ell)}(J_\tau, J_\tau) = \mathbf{A}(J_\tau, J_\tau),$$

and for every $\sigma \in \mathcal{L}_\tau^{\text{nei}}$ (that is, for every neighbor σ of τ):

$$\underline{\mathbf{D}}^{(\ell)}(J_\sigma, J_\tau) = \mathbf{A}(J_\sigma, J_\tau).$$

Using the short-hand

$$\mathbf{A}_{\sigma,\tau} = \mathbf{A}(J_\sigma, J_\tau),$$

we have, for instance,

$$\underline{\mathbf{D}}^{(2)} = \begin{bmatrix} \mathbf{A}_{(4,4)} & \mathbf{A}_{(4,5)} & 0 & 0 \\ \mathbf{A}_{(5,4)} & \mathbf{A}_{(5,5)} & \mathbf{A}_{(5,6)} & 0 \\ 0 & \mathbf{A}_{(6,5)} & \mathbf{A}_{(6,6)} & \mathbf{A}_{(6,7)} \\ 0 & 0 & \mathbf{A}_{(7,6)} & \mathbf{A}_{(7,7)} \end{bmatrix}.$$

and

$$\underline{\mathbf{D}}^{(3)} = \begin{bmatrix} \mathbf{A}_{(8,8)} & \mathbf{A}_{(8,9)} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{A}_{(9,8)} & \mathbf{A}_{(9,9)} & \mathbf{A}_{(9,10)} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{A}_{(10,9)} & \mathbf{A}_{(10,10)} & \mathbf{A}_{(10,11)} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{A}_{(11,10)} & \mathbf{A}_{(11,11)} & \mathbf{A}_{(11,12)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{A}_{(12,11)} & \mathbf{A}_{(12,12)} & \mathbf{A}_{(12,13)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{A}_{(13,12)} & \mathbf{A}_{(13,13)} & \mathbf{A}_{(13,14)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{A}_{(14,13)} & \mathbf{A}_{(14,14)} & \mathbf{A}_{(14,15)} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_{(15,14)} & \mathbf{A}_{(15,15)} \end{bmatrix}.$$

It follows that

$$\underline{\mathbf{D}}^{(2)} - \underline{\mathbf{D}}^{(3)} = \begin{bmatrix} 0 & 0 & \mathbf{A}_{(8,10)} & \mathbf{A}_{(8,11)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{A}_{(9,11)} & 0 & 0 & 0 & 0 \\ \mathbf{A}_{(10,8)} & 0 & 0 & 0 & \mathbf{A}_{(10,12)} & \mathbf{A}_{(10,13)} & 0 & 0 \\ \mathbf{A}_{(11,8)} & \mathbf{A}_{(11,9)} & 0 & 0 & 0 & \mathbf{A}_{(11,13)} & 0 & 0 \\ 0 & 0 & \mathbf{A}_{(12,10)} & 0 & 0 & 0 & \mathbf{A}_{(12,14)} & \mathbf{A}_{(12,15)} \\ 0 & 0 & \mathbf{A}_{(13,10)} & \mathbf{A}_{(13,11)} & 0 & 0 & 0 & \mathbf{A}_{(13,15)} \\ 0 & 0 & 0 & 0 & \mathbf{A}_{(14,12)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{A}_{(15,12)} & \mathbf{A}_{(15,13)} & 0 & 0 \end{bmatrix}.$$

The blocks in the formula above are precisely the blocks whose interactions are dealt with at level ℓ. The non-zero blocks of column τ are precisely the blocks in the interaction list of τ. (For instance, $\mathcal{L}_{10}^{\text{int}} = \{8, 12, 13\}$.) Recalling that if σ is in the interaction list of block τ, then

$$\mathbf{A}_{\sigma,\tau} \approx \tilde{\mathbf{B}}_{\sigma,\tau}\, \mathbf{C}_\tau,$$

we find that

$$\underline{\mathbf{D}}^{(\ell-1)} - \underline{\mathbf{D}}^{(\ell)} \approx \underline{\tilde{\mathbf{B}}}^{(\ell)}\, \underline{\mathbf{C}}^{(\ell)}, \tag{5.10}$$

provided that we define $\underline{\mathbf{C}}^{(\ell)}$ as the block diagonal matrix obtained by placing the 2^ℓ matrices $\mathbf{C}^\tau$ corresponding to boxes τ on level ℓ on the diagonal. For instance,

$$\underline{\mathbf{C}}^{(3)} = \begin{bmatrix} \mathbf{C}^8 & 0 & \cdots & 0 \\ 0 & \mathbf{C}^9 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \mathbf{C}^{15} \end{bmatrix}.$$

Moreover we define $\underline{\tilde{\mathbf{B}}}^{(\ell)}$ as a matrix with $2^\ell \times 2^\ell$ blocks. Each column corresponds to a given box τ on level ℓ. The non-zero blocks on this level are filled with the matrices $\tilde{\mathbf{B}}_{\sigma,\tau}$ for boxes σ that are in the interaction list of τ. For instance,

$$\underline{\tilde{\mathbf{B}}}^{(3)} = \begin{bmatrix} 0 & 0 & \tilde{\mathbf{B}}_{8,10} & \tilde{\mathbf{B}}_{8,11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{\mathbf{B}}_{9,11} & 0 & 0 & 0 & 0 \\ \tilde{\mathbf{B}}_{10,8} & 0 & 0 & 0 & \tilde{\mathbf{B}}_{10,12} & \tilde{\mathbf{B}}_{10,13} & 0 & 0 \\ \tilde{\mathbf{B}}_{11,8} & \tilde{\mathbf{B}}_{11,9} & 0 & 0 & 0 & \tilde{\mathbf{B}}_{11,13} & 0 & 0 \\ 0 & 0 & \tilde{\mathbf{B}}_{12,10} & 0 & 0 & 0 & \tilde{\mathbf{B}}_{12,14} & \tilde{\mathbf{B}}_{12,15} \\ 0 & 0 & \tilde{\mathbf{B}}_{13,10} & \tilde{\mathbf{B}}_{13,11} & 0 & 0 & 0 & \tilde{\mathbf{B}}_{13,15} \\ 0 & 0 & 0 & 0 & \tilde{\mathbf{B}}_{14,12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \tilde{\mathbf{B}}_{14,12} & \tilde{\mathbf{B}}_{14,13} & 0 & 0 \end{bmatrix}.$$

Since $\mathbf{A} = \mathbf{D}^{(1)}$, the Barnes-Hut method now follows from the identity

$$\mathbf{D}^{(1)} = \left(\underline{\mathbf{D}}^{(1)} - \underline{\mathbf{D}}^{(2)}\right) + \left(\underline{\mathbf{D}}^{(2)} - \underline{\mathbf{D}}^{(3)}\right) + \cdots + \left(\underline{\mathbf{D}}^{(L-1)} - \underline{\mathbf{D}}^{(L)}\right) + \underline{\mathbf{D}}^{(L)}.$$

In view of (5.10) we get

$$\mathbf{A} \approx \underline{\tilde{\mathbf{B}}}^{(2)}\, \underline{\mathbf{C}}^{(2)} + \underline{\tilde{\mathbf{B}}}^{(3)}\, \underline{\mathbf{C}}^{(3)} + \underline{\tilde{\mathbf{B}}}^{(4)}\, \underline{\mathbf{C}}^{(4)} + \underline{\mathbf{D}}^{(4)}. \tag{5.11}$$

The matrices $\underline{\mathbf{C}}^{(\ell)}$ are of size $2^\ell R \times N$ and have at most R non-zero elements in each column. Applying $\underline{\mathbf{C}}^{(\ell)}$ to a vector therefore requires RN operations. Similarly, applying $\underline{\tilde{\mathbf{B}}}^{(\ell)}$ to a vector requires at most $3RN$ operations. Since there are $L \sim \log(N)$ terms in (5.11) that relate to far-field interactions, the cost of evaluating the far-field via formula (5.11) is $\mathrm{O}(RN\log(N))$.

5.6. The Fast Multipole Method

Before we proceed to describe the Fast Multipole Method, we note that the formula (5.11) can be symmetrized by expressing it in terms of both outgoing and incoming expansions, via the factorization provided by Lemma 5.4.2. Then

$$\mathbf{A} \approx \underline{\mathbf{B}}^{(2)}\, \mathbf{Z}^{(2)}\, \underline{\mathbf{C}}^{(2)} + \underline{\mathbf{B}}^{(3)}\, \mathbf{Z}^{(3)}\, \underline{\mathbf{C}}^{(3)} + \underline{\mathbf{B}}^{(4)}\, \mathbf{Z}^{(4)}\, \underline{\mathbf{C}}^{(4)} + \underline{\mathbf{D}}^{(4)}. \tag{5.12}$$

The matrices $\underline{\mathbf{B}}^{(\ell)}$ are block-diagonal matrices whose diagonal blocks are formed by all matrices $\mathbf{B}_\tau$ for all boxes τ on level ℓ. For instance,

$$\underline{\mathbf{B}}^{(3)} = \begin{bmatrix} \mathbf{B}_8 & 0 & \cdots & 0 \\ 0 & \mathbf{B}_9 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \mathbf{B}_{15} \end{bmatrix}.$$

The matrices $\underline{\mathbf{Z}}^{(\ell)}$ are analogous to $\underline{\tilde{\mathbf{B}}}^{(\ell)}$ but are formed by the matrices $\mathbf{Z}_{\tau,\sigma}$ instead of the matrices $\tilde{\mathbf{B}}_{\tau,\sigma}$. For instance,

$$\underline{\mathbf{Z}}^{(3)} = \begin{bmatrix} 0 & 0 & \mathbf{Z}_{8,10} & \mathbf{Z}_{8,11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{Z}_{9,11} & 0 & 0 & 0 & 0 \\ \mathbf{Z}_{10,8} & 0 & 0 & 0 & \mathbf{Z}_{10,12} & \mathbf{Z}_{10,13} & 0 & 0 \\ \mathbf{Z}_{11,8} & \mathbf{Z}_{11,9} & 0 & 0 & 0 & \mathbf{Z}_{11,13} & 0 & 0 \\ 0 & 0 & \mathbf{Z}_{12,10} & 0 & 0 & 0 & \mathbf{Z}_{12,14} & \mathbf{Z}_{12,15} \\ 0 & 0 & \mathbf{Z}_{13,10} & \mathbf{Z}_{13,11} & 0 & 0 & 0 & \mathbf{Z}_{13,15} \\ 0 & 0 & 0 & 0 & \mathbf{Z}_{14,12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{Z}_{14,12} & \mathbf{Z}_{14,13} & 0 & 0 \end{bmatrix}.$$

The Fast Multipole Method can be said to achieve its acceleration over the Barnes-Hut method by expressing the matrices $\underline{\mathbf{B}}^{(\ell)}$ and $\underline{\mathbf{C}}^{(\ell)}$ in telescoping factorizations. To obtain the factorization for the operators $\underline{\mathbf{C}}^{(\ell)}$ we start by considering two ways of constructing the outgoing expansion $\hat{\mathbf{q}}_\tau$ for a non-leaf node τ. First we do it directly from the vector q via the formula

$$\hat{\mathbf{q}}_\tau = \mathbf{C}_\tau \, \mathbf{q}(J_\tau). \tag{5.13}$$

Next we note that (??) provides a way of computing $\hat{\mathbf{q}}_\tau$ from the outgoing expansions of the children σ_1 and σ_2 of τ,

$$\begin{aligned} \hat{\mathbf{q}}_\tau = \mathbf{F}_{\tau,\sigma_1}\hat{\mathbf{q}}_{\sigma_1} + \mathbf{F}_{\tau,\sigma_2}\hat{\mathbf{q}}_{\sigma_2} &= \mathbf{F}_{\tau,\sigma_1}\mathbf{C}_{\sigma_1}\,\mathbf{q}(J_{\sigma_1}) + \mathbf{F}_{\tau,\sigma_2}\mathbf{C}_{\sigma_2}\,\mathbf{q}(J_{\sigma_2}) \\ &= [\mathbf{F}_{\tau,\sigma_1}\ \mathbf{F}_{\tau,\sigma_2}] \begin{bmatrix} \mathbf{C}_{\sigma_1} & 0 \\ 0 & \mathbf{C}_{\sigma_2} \end{bmatrix} \mathbf{q}(J_\tau). \end{aligned} \tag{5.14}$$

Since (5.13) and (5.14) must hold for any vector $\mathbf{q}$, it follows that

$$\mathbf{C}_\tau = [\mathbf{F}_{\tau,\sigma_1}\ \mathbf{F}_{\tau,\sigma_2}] \begin{bmatrix} \mathbf{C}_{\sigma_1} & 0 \\ 0 & \mathbf{C}_{\sigma_2} \end{bmatrix}.$$

Letting $\underline{\mathbf{F}}^{(\ell)}$ denote the block diagonal matrix whose diagonal is formed by the blocks $\mathbf{F}_\tau$ for all τ's on level ℓ, we find that

$$\mathbf{C}^{(L-1)} = \mathbf{F}^{(L-1)}\,\mathbf{C}^{(L)}, \qquad \mathbf{C}^{(L-2)} = \mathbf{F}^{(L-2)}\,\mathbf{F}^{(L-1)}\,\mathbf{C}^{(L)}, \qquad \textit{etc.} \tag{5.15}$$

To obtain the analogous expressions for the matrices $\mathbf{B}^{(\ell)}$, let τ be a box with children σ_1 and σ_2. Let u denote a potential caused by *charges that are all well-separated from* τ. (This last assumption is made so that we do not have to worry about contributions from the second term in (4.32).) We then find that

$$\mathbf{u}(J_\tau) = \mathbf{B}_\tau\,\hat{\mathbf{u}}_\tau,$$

and that

$$\begin{aligned} \mathbf{u}(J_\tau) = \begin{bmatrix} \mathbf{u}(J_{\sigma_1}) \\ \mathbf{u}(J_{\sigma_2}) \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{\sigma_1}\,\hat{\mathbf{u}}_{\sigma_1} \\ \mathbf{B}_{\sigma_2}\,\hat{\mathbf{u}}_{\sigma_2} \end{bmatrix} &= \begin{bmatrix} \mathbf{B}_{\sigma_1}\,\mathbf{E}_{\sigma_1,\tau}\,\hat{\mathbf{u}}_\tau \\ \mathbf{B}_{\sigma_2}\,\mathbf{E}_{\sigma_2,\tau}\,\hat{\mathbf{u}}_\tau \end{bmatrix} \\ &\begin{bmatrix} \mathbf{B}_{\sigma_1} & 0 \\ 0 & \mathbf{B}_{\sigma_2} \end{bmatrix} \begin{bmatrix} \mathbf{E}_{\sigma_1,\tau} \\ \mathbf{E}_{\sigma_2,\tau} \end{bmatrix} \hat{\mathbf{u}}_\tau. \end{aligned}$$

It follows that

$$\mathbf{B}_\tau = \begin{bmatrix} \mathbf{B}_{\sigma_1} & 0 \\ 0 & \mathbf{B}_{\sigma_2} \end{bmatrix} \begin{bmatrix} \mathbf{E}_{\sigma_1,\tau} \\ \mathbf{E}_{\sigma_2,\tau} \end{bmatrix},$$

and defining $\underline{\mathbf{E}}^{(\ell)}$ as the block diagonal matrix whose diagonal blocks are the matrices $\mathbf{E}_\tau$ for all τ's on level ℓ, we find that

$$\mathbf{B}^{(L-1)} = \mathbf{B}^{(L)}\,\mathbf{E}^{(L-1)}, \qquad \mathbf{B}^{(L-2)} = \mathbf{B}^{(L)}\,\mathbf{E}^{(L-1)}\,\mathbf{E}^{(L-2)}, \qquad \textit{etc.} \tag{5.16}$$

Inserting (5.15) and (5.16) into the terms arising in (5.12), we find that

$$\begin{aligned} \underline{\mathbf{B}}^{(3)}\,\underline{\mathbf{Z}}^{(3)}\,\underline{\mathbf{C}}^{(3)} &= \underline{\mathbf{B}}^{(4)}\,\underline{\mathbf{E}}^{(3)}\,\underline{\mathbf{Z}}^{(3)}\,\underline{\mathbf{F}}^{(3)}\,\underline{\mathbf{F}}^{(4)}, \\ \underline{\mathbf{B}}^{(2)}\,\underline{\mathbf{Z}}^{(2)}\,\underline{\mathbf{C}}^{(2)} &= \underline{\mathbf{B}}^{(4)}\,\underline{\mathbf{E}}^{(3)}\,\underline{\mathbf{E}}^{(2)}\,\underline{\mathbf{Z}}^{(2)}\,\underline{\mathbf{F}}^{(2)}\,\underline{\mathbf{F}}^{(3)}\,\underline{\mathbf{C}}^{(4)}. \end{aligned}$$

This implies that (5.12) can be expressed in telescoping form as

$$\underline{\mathbf{A}} \approx \underline{\mathbf{B}}^{(4)}\left(\underline{\mathbf{E}}^{(3)}\left(\underline{\mathbf{E}}^{(2)}\,\underline{\mathbf{Z}}^{(2)}\,\underline{\mathbf{F}}^{(2)} + \underline{\mathbf{Z}}^{(3)}\right)\underline{\mathbf{F}}^{(3)} + \underline{\mathbf{Z}}^{(4)}\right)\underline{\mathbf{C}}^{(4)} + \underline{\mathbf{D}}^{(4)}. \tag{5.17}$$

The advantage of the expression (5.17) over (5.12) is that the factors get exponentially smaller as we proceed towards coarser levels. This means that the cost of processing level ℓ is now $\mathrm{O}(2^\ell R^2)$,

which should be compared to the constant $\mathrm{O}(2^L R^2)$ cost of processing level ℓ in the Barnes-Hut method.

CHAPTER 6

Boundary Integral Equation methods: How to very rapidly solve Laplace's equation on a general domain

Summary: *We describe a set of methods for computing approximate solutions to linear boundary value problems. The foundation of these methods is a reformulation of the partial differential equation as an integral equation. In many cases, the resulting integral equation can rapidly be solved to very high accuracy, and in an entirely stable manner. The computational speed of these methods compares favorably with other numerical techniques for partial differential equations. The most dramatic speed-up occurs for problems that can be reformulated as integral equations defined on the boundary alone.*

6.1. Introduction

6.1.1. The basic idea. The set of methods described in this note are applicable to a wide range of linear boundary value problems. For simplicity, we suppose that the BVP we are interested in solving takes the form

$$A\,u = 0 \qquad \text{on } \Omega, \tag{6.1}$$

$$u = f \qquad \text{on } \Gamma, \tag{6.2}$$

where A is a linear elliptic partial differential operator with constant coefficients, where Ω is a domain in $\mathbb{R}^d$ with boundary Γ, and where f is a pre-scribed function on Γ.[1] It is in many cases possible to reformulate a BVP such as (6.1), (6.2) as a boundary integral equation of the form

$$v(x) + \int_\Gamma k(x,y)\,v(y)\,dl(y) = g(x), \tag{6.3}$$

where k is a *kernel function* derived from the fundamental solution associated with the operator A, and g is a data function derived from the boundary data f (a specific example is given in Section 6.3.1). There typically exist several different BIE formulations of the same BVP; choosing the "right" one is important, *cf.* Remark 6.1.

Upon discretization, the BIE (6.3) turns into a system of N linear algebraic equations that we write

$$\mathbf{v} + \mathbf{A}\,\mathbf{v} = \mathbf{g}, \tag{6.4}$$

where $\mathbf{A}$ is a matrix approximating the integral operator in (6.3). An apparent obstacle to solving (6.4) rapidly is that the matrix $\mathbf{A}$ is almost always dense. However, this matrix typically has properties that allow the computation of products $\mathbf{w} \mapsto \mathbf{A}\,\mathbf{w}$ cheaply. For instance, there exist Fast Multipole Methods (FMMs) for a wide variety of kernels that perform such matrix-vector multiplications in $O(N)$ arithmetic operations, with a small constant. Then iterative solvers (such as GMRES) can be used to determine an approximate solution of (6.4) very rapidly.

[1] For BVP such as (6.1), (6.2), the equivalent integral equations are defined on the boundary alone. This is the environment in which FIEM has the strongest competitive advantage. However, it can profitably be applied to a wide range of other problems as well, including problems with a body load, see Section 6.1.3.

REMARK 6.1. *The number of iterations required to achieve an accurate solution of (6.4) using an iterative solver depends strongly on the choice of boundary integral equation formulation. Leaving the details to Section ?, we simply mention here that it is almost always a good idea to use a BIE of the form (6.3), in which the integral operator is compact in some function space. An equation of this form is called a second kind Fredholm equation. As shown in Section 6.3, such formulations can lead to extra-ordinarily rapid convergence (double precision accuracy in the linear solve in 15 iterations or less).*

In some environments, there also exist $O(N)$ numerical techniques that do not rely on iterative solvers, but directly compute a representation of the inverse of the operator in (6.3).

REMARK 6.2 (Terminology). *In this note, the term "fast" has a precise technical meaning: We say that a numerical method is fast if its execution time scales close to linearly with problem size. To be precise, if it has $O(N(\log N)^r)$ complexity for some real number r. (In practice, $r = 0$ can often be achieved, and we rarely need r larger than 2.) The term "Fast Integral Equation Method" (FIEM) is then taken to refer to a combination of an integral equation formulations and a fast solver. Our usage of the term "fast" is standard in this context, but the acronym FIEM is introduced purely for the purposes of this note.*

6.1.2. Benefits. When a BVP can be rewritten as a second kind Fredholm equation on the boundary, FIEM has several advantages:

(1) The condition number of the numerical problem is typically similar to the condition number of the actual "physics" of the problem. This is in contrast to discrete systems arising from discretizations of differential equations, which typically have very large condition numbers. (High condition numbers lead to several difficulties; loss of accuracy being one.)

(2) The computational complexity is asymptotically optimal in the sense that the number of floating point operations required scales linearly with the true complexity of the problem. In the example in Section 6.1.1, it scales linearly with the number of degrees of freedom required to discretized the *boundary* of the domain. The data provided is defined on a set of dimension $d - 1$ so we should not have to discretize a set of dimension d.

(3) The construction of discretization schemes that are both stable and of high order is somewhat easier in the integral equation environment than it is in the differential equation environment.

(4) Handling *exterior* problems where the domain Ω is the region outside some given contour or surface Γ is quite hard when discretizing the differential equation. Typically some artificial external boundary is introduced. In contrast, in the FIEM environment, the equation is inherently formulated on the finite domain Γ, and no such difficulties arise.

As a consequence of points 1 and 3, FIEM has the capacity to produce solutions of unparalleled accuracy; even on complicated problems, relative errors of 10^{-10} or less can be obtained.

It is worth emphasizing that FIEM is fast not only in the theoretical sense that it scales linearly with the complexity of the problem; the constant of proportionality is often small. For BVP's involving constant coefficient operators, and no body loads, FIEM can be orders of magnitude

faster than other methods. The only real competitor in this environment are methods based on the FFT (or spectral methods more generally), but such methods are limited in that they typically achieve high performance only on simple domains. They also tend to be limited in the extent to which they can handle adaptivity, and non-uniform grids more generally.

6.1.3. Generalizations. FIEM has a profound competitive advantage when applied to BVPs that can be reformulated as integral equations defined on the boundary. With some exceptions, this happens when the partial differential operator has constant coefficients, and when there is no body load. However, it is occasionally advantageous to use FIEM for problems involving body loads, and / or non-constant coefficients as well. In such environments, one must use integral equations that involve volume integrals which implies a higher computational cost. However, the unrivaled stability and accuracy of an integral equation formulation can sometimes justify the cost; for instance when dealing with scattering problems close to resonant frequencies. These matters are discussed further in Sections 6.9 and 6.10.

6.1.4. When to use FIEM. Generally speaking, FIEM works for elliptic linear BVPs involving partial differential operators whose highest order term has constant coefficients. However, they are particularly appealing for situations involving any of the following:

- Constant coefficient differential operators.
- No body load.
- Exterior problems defined on infinite domains.
- Relatively ill-conditioned physics (thin domains, scattering at short wave-lengths, etc.)
- Very high accuracy is required.

As a more practical consideration, the set-up costof using an FIEM is typically higher than that of using say a finite element discretization so in situations where you are only interested in a one-off solve at moderate accuracy, FIEM might not be the best choice. (The hardest part is usually to find a good integral equation formulation. In most standard situations, this has already been done, but if you have unusual boundary conditions, multiple types of physics going on, strange boundaries in 3D, etc., then finding the right integral equation could be *a lot* of work.)

In addition to the "fundamental limitations" described in the previous paragraph, there is also an issue of availability of software. There is any number of software packages available for solving PDEs using finite element, finite difference, or finite volume methods — some commercial and some free. Some packages are very user friendly with GUIs, the ability to import geometry information from a CAD program, the ability to handle[2] almost any differential equation thrown at them, etc. In contrast, there are relatively few packages available for discretizing and solving an equation via FIEM, and what is available is often raw Fortran code.

6.1.5. Outline of chapter.

6.2. Notation, orientation of boundaries, etc

Unless otherwise noted, Ω will in this section denote a simply connected boundary domain in the plane with smooth boundary Γ. For $\boldsymbol{x} \in \Gamma$, $\boldsymbol{n}(\boldsymbol{x})$ denotes a unit normal that points out from Ω. We let Ψ denote the domain exterior to Γ. See Figure 6.1 for an illustration.

[2]The term "handle" does not necessarily imply that the output is particularly close to the correct solution

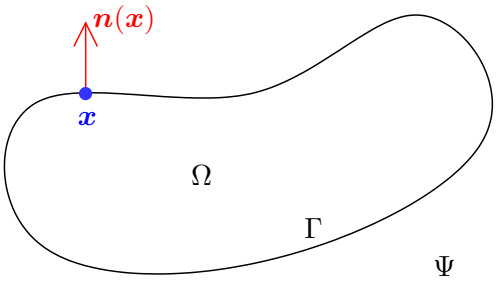


FIGURE 6.1. Notation for geometries. Γ is a smooth contour in the plane. Ω is the domain interior to Γ and Ψ is the exterior domain. $\boldsymbol{n}(\boldsymbol{x})$ denotes the outwards pointing unit normal.

6.3. A model problem

In this section, we demonstrate how FIEM can be used to solve a specific problem: The Laplace equation on a bounded smooth domain in $\mathbb{R}^2$ with Dirichlet boundary data. The objective is to demonstrate the entire solution procedure for a particularly simple example. We will then return to different points in later sections and describe how to generalize the techniques to other equations and to other geometries.

6.3.1. Reformulation of a Boundary Value Problem as a Boundary Integral Equation. We consider the BVP

$$-\Delta u = 0 \qquad \text{on } \Omega, \tag{6.5}$$

$$u = f \qquad \text{on } \Gamma, \tag{6.6}$$

where Ω is a simply connected open set in $\mathbb{R}^2$ with smooth boundary Γ, and where f is a prescribed function on Γ. We seek to rewrite (6.5, 6.6) as an equation defined on Γ only. To this end, we make the Ansatz

$$u(x) = \int_\Gamma \frac{n(x') \cdot (x - x')}{2\pi |x - x'|^2} \sigma(x')\, dl(x'), \qquad \text{for } x \in \Omega, \tag{6.7}$$

where $n(x')$ is the unit outwards pointing normal of Γ at x', and where σ is an unknown function on Γ. Note that for any σ, the function u defined by (6.7) satisfies (6.5).[3] Equation (6.6) is satisfied if and only if the function u defined by (6.7) satisfies

$$\lim_{y \to x,\, y \in \Omega} u(y) = f(x), \qquad \forall\, x \in \Gamma. \tag{6.8}$$

A simple evaluation of the limit (6.8) (see Section 6.5) yields

$$-\frac{1}{2}\sigma(x) + \int_\Gamma \frac{n(x') \cdot (x - x')}{2\pi |x - x'|^2} \sigma(x')\, dl(x') = f(x), \qquad \text{for } x \in \Gamma. \tag{6.9}$$

In a strong sense, the BVP (6.5, 6.6) is equivalent to the BIE (6.9).

[3] Conversely, it is known from classical potential theory that any solution of (6.5, 6.6) can be written in the form (6.7) for some function σ.

6.3.2. Parameterization of BIE. We next seek to approximate the BIE (6.9) by a set of linear algebraic equations. To this end, we parameterize Γ via a function $G(t) = (G_1(t), G_2(t))$ such that

$$\Gamma = \{G(t) : t \in [0, T]\}. \tag{6.10}$$

Causing hopefully not too much confusion, we think of any functions defined on Γ as functions of the parameter t:[4]

$$u = u(t), \qquad \sigma = \sigma(t), \qquad f = f(t).$$

We can also express the double layer kernel in terms of parameter values:

$$\begin{aligned} D(t,t') &= \frac{n(t') \cdot \big(G(t) - G(t')\big)}{2\pi |G(t) - G(t')|^2} \\ &= \frac{G_2'(t')\,\big(G_1(t) - G_1(t')\big) - G_1'(t')\,\big(G_2(t) - G_2(t')\big)}{2\pi\,\big((G_1'(t'))^2 + (G_2'(t'))^2\big)^{1/2}\,\big((G_1(t) - G_1(t'))^2 + (G_2(t) - G_2(t'))^2\big)}, \end{aligned} \tag{6.11}$$

where we used that

$$n(t') = \frac{\big(G_2'(t'),\, -G_1'(t')\big)}{(G_1'(t'))^2 + (G_2'(t'))^2}.$$

Inserting the parameterization into (6.9) we obtain

$$-\frac{1}{2}\sigma(t) + \int_0^T D(t,t')\,\sigma(t')\,|G'(t')|\,dt' = f(t), \qquad \text{for } \tau \in [0, T]. \tag{6.12}$$

(Observe that the formulas can be simplified slightly since the term $|G'(t')|$ appears in the denominator of $D(t,t')$. This would be beneficial for now, but would cause problems later when we consider more general quadratures.)

REMARK 6.3. *In most cases, the kernel in a BIE is singular as $t' \to t$, and at first glance it appears the the kernel $D(t,t')$ in (6.11) is singular as well. However, it turns out that when the contour is smooth, we have $n(x') \cdot (x - x') = O(|x - x'|^2)$ since the angle between $n(x')$ and $x - x'$ approach $\pi/2$. In consequence, the singularities in the numerator and the denominator precisely cancel, and the kernel is in fact a smooth function. Using l'Hôpital's rule twice, we find that*

$$\lim_{t' \to t} D(t,t') = \frac{G_2'(t)\,G_1''(t) - G_1'(t)\,G_2''(t)}{4\pi |G'(t)|^3}.$$

6.3.3. Discretization of BIE. We discretize (6.12) using the *Nyström method*, as follows: First we discretize the integral in (6.12) using a quadrature rule with weights $(w_j)_{j=1}^N$, and quadrature points $(t_j)_{n=1}^N \subset [0, T]$, and obtain the approximate equation

$$-\frac{1}{2}\sigma(t) + \sum_{j=1}^{N} w_j\, D(t,t_j)\,|G'(t_j)|\,\sigma(t_j) = f(t), \qquad \text{for } t \in [0, T]. \tag{6.13}$$

To obtain a discrete equation from (6.13), we require that (6.13) should hold for at the quadrature points, *i.e.* when $t = t_1, \ldots, t_N$. This results in the system of equations

$$-\frac{1}{2}\sigma(t_i) + \sum_{j=1}^{N} w_j\, D(t_i,t_j)\,|G'(t_j)|\,\sigma(t_j) = f(t_i), \qquad \text{for } i = 1, 2, \ldots, N. \tag{6.14}$$

[4]To be mathematically rigorous, we should have defined a new function $U = u \circ G$ (so that $U(t) = u(G(t))$), a new function $\Sigma = \sigma \circ G$, and a new function $F = f \circ G$. Then u, σ, and f would have been functions defined on Γ, and U, Σ, and F would have been functions defined on $[0, T]$.

We write (6.14) in matrix form as

$$\left(-\frac{1}{2}\mathbf{I} + \mathbf{B}\right)\boldsymbol{\sigma} = \mathbf{f}, \tag{6.15}$$

where $\mathbf{f}$ is a vector of pointwise samples from f, $\mathbf{f}(i) = f(t_i)$, and where $\boldsymbol{\sigma}$ is a vector approximating the exact solution σ, $\boldsymbol{\sigma}(i) \approx \sigma(t_i)$, and where $\mathbf{B}$ has the entries

$$\mathbf{B}(i,j) = D(t_i, t_j)\,|G'(t_j)|\,w_j.$$

Finally, we note that since the integrand in this particular case is smooth and periodic, the trapezoidal rule ($w_i = T/N$, $t_i = i\,T/N$) achieves superpolynomial convergence rates; and find that

$$\mathbf{B}(i,j) = \frac{T}{N}\,D(t_i, t_j)\,|G'(t_j)|.$$

6.3.4. Post-processing: The numerical method described so far yields as its result the dipole distribution σ, which is generally not the quantity sought. However, any other quantity of interest can be computed via the formulas:

$$u(x) = -\frac{1}{2}\sigma(x) + \int_\Gamma D(x,x')\,\sigma(x')\,dl(x'), \qquad x \in \Gamma \tag{6.16}$$

$$u(x) = \int_\Gamma D(x,x')\,\sigma(x')\,dl(x'), \qquad x \in \Omega. \tag{6.17}$$

The kernel in the first formula is smooth and it can easily be evaluated upon discretization via the trapezoidal rule. The kernel in the second formula can also be evaluated very easily whenever the target point x is not that close to the boundary. However, if x is close to the boundary, then $D(x,x')$ has very sharp derivatives when x' approaches x and some care must be exercise.

The evaluations of the layer potentials (6.16) and (6.17) can be accelerated by the Fast Multipole Method if the potential is sought at a large number of target points.

6.4. Review of the Green identities

Recall that if $\boldsymbol{f}$ is a vector valued C^1 function on Ω, then the Gauss theorem reads

$$\int_\Omega \nabla\cdot\boldsymbol{f} = \int_\Gamma \boldsymbol{n}\cdot\boldsymbol{f},$$

where $\boldsymbol{n}$ is the outward pointing unit normal to Γ. Now suppose that u and v are two C^2 scalar value functions. Applying the Gauss theorem to $\boldsymbol{f} = u\nabla v$ and $\boldsymbol{f} = (\nabla u)v$ we find

$$\int_\Omega \big(u\,\Delta v + \nabla u\cdot\nabla v\big) = \int_\Gamma u\,\frac{\partial v}{\partial n} \tag{6.18}$$

$$\int_\Omega \big(\Delta u\,v + \nabla u\cdot\nabla v\big) = \int_\Gamma \frac{\partial u}{\partial n}\,v. \tag{6.19}$$

Subtracting (6.19) from (6.18), we find

$$\int_\Omega \big(u\,\Delta v - \Delta v\,u\big) = \int_\Gamma \left(u\,\frac{\partial v}{\partial n} - \frac{\partial u}{\partial n}\,v\right).$$

Now fix a point $x \in \Omega$ and set

$$v(y) = -\frac{1}{2\pi}\log|y-x|.$$

Further, assume u satisfies

$$-\Delta u = 0 \qquad \text{in } \Omega,$$

let ε be a small number (smaller than $\operatorname{dist}(\Gamma, x)$), set

$$\Omega_\varepsilon = \Omega \backslash B_\varepsilon(x)$$

and observe that

$$\partial\Omega_\varepsilon = \Gamma \cup \Gamma_\varepsilon$$

where

$$\Gamma_\varepsilon = \partial B_\varepsilon(x).$$

Now note that $\Delta u(y) = \Delta v(y)$ for $y \in \Omega_\varepsilon$ to obtain

$$0 = \int_\Gamma \left(u \frac{\partial v}{\partial n} - \frac{\partial u}{\partial n} v \right) + \int_{\Gamma_\varepsilon} \left(u \frac{\partial v}{\partial n} - \frac{\partial u}{\partial n} v \right). \tag{6.20}$$

We next seek to evaluate the limit as $\varepsilon \to 0$. For $y \in \Gamma_\varepsilon$, we find

$$\begin{aligned} v(y) =& -\frac{1}{2\pi} \log|x-y| = -\frac{1}{2\pi} \log \varepsilon \\ \frac{\partial v}{\partial n}(y) =& n(y) \cdot \nabla v(y) = \frac{x-y}{|x-y|} \frac{x-y}{2\pi|x-y|^2} = \frac{1}{2\pi|x-y|} = \frac{1}{2\pi\varepsilon}. \end{aligned}$$

Since $u \in C(\Omega)$ we find

$$\lim_{\varepsilon\to 0} \int_{\Gamma_\varepsilon} u(y) \frac{\partial v}{\partial n}(y)\, dl(y) = \lim_{\varepsilon\to 0} \int_{\Gamma_\varepsilon} u(y) \frac{1}{2\pi\varepsilon}\, dl(y) = u(x).$$

Since $u \in C^1(\Omega)$ we know that for some finite M, we have $|\nabla u(y)| \le M$ and so

$$\lim_{\varepsilon\to 0} \left| \int_{\Gamma_\varepsilon} \frac{\partial u}{\partial n}(y)\, v(y)\, dl(y) \right| \le \limsup_{\varepsilon\to 0} \int_{\Gamma_\varepsilon} M \frac{1}{2\pi} |\log\varepsilon|\, dl(y) = \limsup_{\varepsilon\to 0} M\, \varepsilon\, |\log\varepsilon| = 0.$$

Taking the limit $\varepsilon \to 0$ in (6.20) we obtain

$$u(x) = \int_\Gamma \frac{n(y)\cdot(y-x)}{2\pi|y-x|^2} u(y)\, dl(y) - \int_\Gamma \frac{1}{2\pi} \log|y-x|\, u_n(y)\, dl(y). \tag{6.21}$$

We introduce two integral operators called *layer potentials* via

$$\text{The single layer operator:} \qquad [S\,w](x) = \int_\Gamma \frac{-1}{2\pi} \log|x-y|\, w(y)\, dl(y), \tag{6.22}$$

$$\text{The double layer operator:} \qquad [D\,w](x) = \int_\Gamma \frac{n(y)\cdot(x-y)}{2\pi|x-y|^2} w(y)\, dl(y). \tag{6.23}$$

Then (6.21) takes the form

$$u(x) = -[D\,u](x) + [S\,u_n](x) \qquad x \in \Omega. \tag{6.24}$$

An entirely analogous computation shows that

$$\frac{1}{2} u(x) = -[D\,u](x) + [S\,u_n](x) \qquad x \in \Gamma. \tag{6.25}$$

6.5. Derivation of BIEs for Laplace's equation

6.5.1. The direct formulation. We first derive a BIE for the Dirichlet problem

$$\begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ u(x) = g(x), & x \in \Gamma. \end{cases}$$

From (6.25) we find that

$$\frac{1}{2}\, g(x) + [D\, g](x) = [S\, u_n](x) \qquad x \in \Gamma. \tag{6.26}$$

Equation (6.27) can be solved for the unknown boundary function u_n. Once u_n has been determined, the potential u can be recovered from (6.24).

Next consider the Neumann problem

$$\begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ u_n(x) = h(x), & x \in \Gamma. \end{cases}$$

From (6.25) we find that

$$\frac{1}{2}\, u(x) + [D\, u](x) = [S\, h](x) \qquad x \in \Gamma. \tag{6.27}$$

Equation (6.27) can be solved for the unknown boundary function $u|_\Gamma$. Once $u|_\Gamma$ has been determined, the potential u can be recovered from (6.24).

REMARK 6.4. *The so called "direct" formulation for works very well for the Neumann problem since the operator $(1/2)I + D$ is an invertible second kind Fredholm operator. It works less well for the Dirichlet problem since S is compact, and the inverse of a compact operator is unbounded. The difficulties involved can all be resolved, but it takes some effort.*

6.5.2. The indirect formulation. Consider the Dirichlet problem

$$\begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ u(x) = g(x), & x \in \Gamma. \end{cases}$$

We look for a solution of the form

$$u(x) = [D\sigma](x),$$

where σ is an unknown boundary potential. A well-conditioned equation for σ is obtained from the formula

$$\lim_{z \in \Omega,\ z \to x} [D\sigma](z) = -\frac{1}{2}\sigma(x) + [D\sigma](x), \qquad x \in \Gamma.$$

We find

$$-\frac{1}{2}\sigma(x) + [D\sigma](x) = g(x).$$

Next consider the Neumann problem

$$\begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ u_n(x) = h(x), & x \in \Gamma. \end{cases}$$

We look for a solution of the form

$$u(x) = [S\sigma](x),$$

where σ is an unknown boundary potential. For $x \in \Gamma$, we find

$$u_n(x) = \lim_{z\in\Omega,\ z\to x} n(x)\cdot\nabla[S\sigma](z) = \frac{1}{2}\sigma(x) + [D^*\sigma](x),$$

where D^* is the *adjoint* of the double layer potential,

$$[D^*\sigma](x) = \int_\Gamma \frac{-n(x)\cdot(x-y)}{2\pi\,|x-y|^2}\,ds(y).$$

(Observe that the normal is evaluated at the target point x, not at the integration variable y.) We find that the equation for σ reads

$$\frac{1}{2}\sigma(x) + [D^*\sigma](x) = h(x), \qquad x\in\Gamma. \tag{6.28}$$

The operator $(1/2)I + D^*$ is a second kind Fredholm operator, but it has a one-dimensional null-space, and a one-dimensional co-range. This is very natural, since we know (1) that the Neumann problem does not determine u other than up to translation, and (2) that the Neumann problem is not well-posed unless

$$\int_\Gamma h(x)\,ds(x) = 0.$$

Some care must be exercised in solving (6.28). An easy way of handling this is to modify the Ansatz to read

$$u(x) = [S\sigma](x) - \frac{1}{2\pi}\log|x-\hat{x}|\int_\Gamma \sigma(x)\,ds(x), \tag{6.29}$$

where $\hat{x}$ is a point chosen in the interior of Ω. Then the new equation for σ is

$$\frac{1}{2}\sigma(x) + [D^*\sigma](x) - \frac{n(x)\cdot(x-\hat{x})}{2\pi|x-\hat{x}|^2}\int_\Gamma \sigma(x)\,ds(x) = h(x), \qquad x\in\Gamma. \tag{6.30}$$

Integrating (6.30) over Γ, we find

$$\underbrace{\int_\Gamma \left(\frac{1}{2}\sigma(x) + [D^*\sigma](x)\right) ds(x)}_{=0} - \underbrace{\int_\Gamma \frac{n(x)\cdot(x-\hat{x})}{2\pi|x-\hat{x}|^2}\,ds(x)}_{=-1}\int_\Gamma \sigma(x)\,ds(x) = \underbrace{\int_\Gamma h(x)\,ds(x)}_{=0}.$$

In other words, when (6.30) holds, we have

$$\int_\Gamma \sigma(x)\,ds(x) = 0,$$

which implies that the "new" term in (6.29) vanishes.

REMARK 6.5. *The exterior Neumann problem is easier to handle. Consider the equation*

$$\begin{cases} -\Delta u(x) = 0, & x\in\Psi, \\ \quad u_n(x) = h(x), & x\in\Gamma. \end{cases} \tag{6.31}$$

Note that we do not need h to integrate to zero for (6.31) to have a solution. We still have a problem in that the solution is not uniquely determined unless we specify some decay condition at infinity. Observing that the total charge that generates u is $\int_\Gamma h$, it is natural to require u to satisfy

$$u(x) = \frac{1}{2\pi}\log\frac{1}{|x|}\int_\Gamma h(x')\,ds(x') + O(1/|x|), \qquad \text{as } |x|\to\infty.$$

Formally, we couple (6.31) with

$$\lim_{R\to\infty}\sup_{|x|=R}\left|u(x) - \frac{1}{2\pi}\log\frac{1}{|x|}\int_\Gamma h(x')\,ds(x')\right| = 0. \tag{6.32}$$

Condition (6.32) is automatically satisfied if we look for a solution of the form

$$u(x) = [S\sigma](x),$$

where σ is an unknown boundary potential. The relevant equation for σ is now

$$-\frac{1}{2}\sigma(x) + [D^*\sigma](x) = h(x), \qquad x \in \Gamma. \tag{6.33}$$

The operator $-(1/2)I + D^*$ is a second kind Fredholm operator that is onto and one-to-one.

6.6. Discretization of BIEs

6.6.1. The Nyström method.

6.6.2. Collocation.

6.6.3. Galerkin.

6.7. Quadrature rules for smooth contours

6.8. Quadrature rules for contours with corners

6.9. FIEM for problems involving body loads

In Section 6.1.1, we described how FIEM works for a linear BVP with no body load. It is in principle trivial to extend the method to a problem with a body load by simply splitting the problem into two parts: First a particular solution that satisfies the PDE but not the boundary condition is determined, and then a homogeneous solution that corrects the boundary condition is determined using the ideas described in Section 6.1.1. However, some numerical complications may arise, as illustrated below.

Suppose that we wish to solve the equation

$$A\,u = h \qquad \text{on } \Omega, \tag{6.34}$$

$$u = f \qquad \text{on } \Gamma, \tag{6.35}$$

where h is a given body load. The first step is to construct a "particular solution", $u_{\rm p}$, with the property that

$$A\,u_{\rm p} = h, \qquad \text{on } \Omega.$$

Setting $u_{\rm h} = u - u_{\rm p}$, we find that $u_{\rm h}$ satisfies

$$A\,u_{\rm h} = 0 \qquad \text{on } \Omega,$$

$$u_{\rm h} = f \qquad \text{on } \Gamma.$$

Since $u_{\rm h}$ satisfies a homogeneous problem, it can be determined using the boundary integral technique described in Section 6.1.1.

Now, if Ω is a rectangle (or some other simple geometrical shape), and if h is a smooth function on h, then one can easily compute a particular solution $u_{\rm p}$ via

$$u_{\rm p}(x) = \int_\Omega G(x,y)\, h(y)\, dA(y).$$

When h is smooth, but Ω is not a simple geometric shape, it may be easier to compute an integral over some extension $\bar\Omega$ of Ω,

$$u_{\rm p}(x) = \int_{\bar\Omega} G(x,y)\, \bar h(y)\, dA(y),$$

where $\bar h$ is a smooth extension of h from Ω to $\bar\Omega$, see [?]. Another alternative, which is attractive when h is not easily integrable, is to use the FFT or some other fast Poisson solver so solve the equation

$$A\, u_{\rm p} = \ h \qquad \text{on } \bar\Omega, \tag{6.36}$$

$$u_{\rm p} = f \qquad \text{on } \partial\bar\Omega. \tag{6.37}$$

FIEM for problems involving body loads are discussed in [Greengard and Etheridge, Biros, etc].

6.10. FIEM for problems involving partial differential operators with non-constant coefficients

CHAPTER 7

Iterative solvers

CHAPTER 8

Extensions to other equations

Helmholtz

Maxwell

Elasticity

Stokes

CHAPTER 9

Computing in the real world

3D problems

human time vs computer time

APPENDIX A

Translation operators of the classical Laplace FMM in $\mathbb{R}^2$

1.0.1. The outgoing from sources translation operator. Let τ denote a box with center $\mathbf{c}_\tau$ and side length $2a$. It holds sources $\mathbf{q}^\tau = [q_j^\tau]_{j=1}^J$ at locations $\mathbf{X}^\tau = [\mathbf{x}_j^\tau]_{j=1}^J$. Consider the potential

$$u(\mathbf{y}) = \sum_{j=1}^{J} q_j^\tau \log(\mathbf{y} - \mathbf{x}_j^\tau).$$

The potential admits the multipole expansion

$$u(\mathbf{y}) = \hat{q}_0^\tau \log(\mathbf{y} - \mathbf{c}_\tau) + \sum_{i=1}^{\infty} \hat{q}_i^\tau \frac{1}{(\mathbf{y} - \mathbf{c}_\tau)^i},$$

where

$$\hat{q}_0^\tau = \sum_{j=1}^{J} q_j^\tau,$$

$$\hat{q}_i^\tau = \sum_{j=1}^{J} -q_j^\tau \frac{1}{i} (\mathbf{x}_j^\tau - \mathbf{c}_\tau)^i.$$

1.0.2. The outgoing from outgoing translation operator. Suppose that a child σ of a box τ holds a source distribution represented by the outgoing expansion $\hat{\mathbf{q}}^\sigma$,

$$u(\mathbf{y}) = \hat{q}_0^\sigma \log(\mathbf{y} - \mathbf{c}_\sigma) + \sum_{j=1}^{\infty} \hat{q}_j^\sigma \frac{1}{(\mathbf{y} - \mathbf{c}_\sigma)^j},$$

Then this field can alternatively be represented

$$u(\mathbf{y}) = \hat{q}_0^\tau \log(\mathbf{y} - \mathbf{c}_\tau) + \sum_{i=1}^{\infty} \hat{q}_i^\tau \frac{1}{(\mathbf{y} - \mathbf{c}_\tau)^i},$$

where

$$\hat{q}_0^\tau = \hat{q}_0^\sigma,$$

$$\hat{q}_i^\tau = -\hat{q}_0^\sigma \frac{1}{i} (\mathbf{c}_\sigma - \mathbf{c}_\tau)^i + \sum_{j=1}^{i} \hat{q}_j^\sigma \binom{i-1}{j-1} (\mathbf{c}_\sigma - \mathbf{c}_\tau)^{i-j}.$$

1.0.3. The incoming from outgoing translation operator. Suppose that τ and σ are two well-separated boxes, and that σ holds a source distribution represented by an outgoing expansion $\hat{\mathbf{q}}^\sigma$. Then the field in τ caused by these sources can be represented by an incoming expansion $\hat{\mathbf{u}}^\tau$,

$$u(\mathbf{x}) = \sum_{i=0}^{\infty} \hat{u}_i^\tau (\mathbf{x} - \mathbf{c}_\tau)^i,$$

where

$$\hat{u}_0^\tau = \hat{q}_0^\sigma \log(\mathbf{c}_\tau - \mathbf{c}_\sigma) + \sum_{j=1}^{\infty} \hat{q}_j^\sigma \, (-1)^j \, \frac{1}{(\mathbf{c}_\sigma - \mathbf{c}_\tau)^j},$$

$$\hat{u}_i^\tau = -\hat{q}_0^\sigma \frac{1}{i(\mathbf{c}_\sigma - \mathbf{c}_\tau)^i} + \sum_{j=1}^{\infty} \hat{q}_j^\sigma (-1)^j \binom{i+j-1}{j-1} \frac{1}{(\mathbf{c}_\sigma - \mathbf{c}_\tau)^{i+j}}.$$

1.0.4. The incoming from incoming translation operator. Suppose that τ is the child of a box σ. Suppose further that the incoming expansion $\hat{\mathbf{u}}^\sigma$ represents a potential in σ caused by sources that are all well-separated from σ. Then these sources are also well-separated from τ, and the potential in τ can be represented via an incoming expansion $\hat{\mathbf{u}}^\tau$ so that

$$u(\mathbf{x}) = \sum_{i=0}^{\infty} \hat{u}_i^\tau (\mathbf{x} - \mathbf{c}_\tau)^i,$$

where

$$\hat{u}_i^\tau = \sum_{j=i}^{\infty} \hat{u}_j^\sigma \binom{j}{i} (\mathbf{c}_\tau - \mathbf{c}_\sigma)^{j-i}.$$

1.0.5. The targets from incoming translation operator. Suppose that τ is a box whose incoming potential is represented via the incoming representation $\hat{\mathbf{u}}^\tau$. Then the potential at the actual target points are constructed via

$$u(\mathbf{x}_i^\tau) = \sum_{j=0}^{\infty} \hat{u}_j^\tau (\mathbf{x}_i^\tau - \mathbf{c}_\tau)^j.$$