18.330
Introduction to numerical analysis
Class notes

Laurent Demanet

Draft February 7, 2022

Preface

Good additional references are

- Burden and Faires, Introduction to numerical analysis
- Suli and Mayers, Introduction to numerical analysis
- Trefethen, Spectral methods in Matlab

as well as the pdf notes on numerical analysis by John Neu.

If the text points you to some external reference for further study, the latter is generally not part of the material for the class.

Work in progress!

Chapter 1

Series and sequences

Throughout these notes we'll keep running into Taylor series and Fourier series. It's important to understand what is meant by convergence of series before getting to numerical analysis proper. These notes are self-contained, but two good extra references for this chapter are Tao, Analysis I; and Dahlquist and Bjorck, Numerical methods.

A sequence is a possibly infinite collection of numbers lined up in some order:

$$a_1, a_2, a_3, \ldots$$

A series is a possibly infinite sum:

$$a_1 + a_2 + a_3 + \ldots$$

We'll consider real numbers for the time being, but the generalization to complex numbers is a nice exercise which mostly consists in replacing each occurrence of an absolute value by a modulus.

The questions we address in this chapter are:

- What is the meaning of an infinite sum?
- Is this meaning ever ambiguous?
- How can we show convergence vs. divergence?
- When can we use the usual rules for finite sums in the infinite case?

1.1 Convergence vs. divergence

We view infinite sums as limits of partial sums. Since partial sums are sequences, let us first review convergence of sequences.

Definition 1. *A sequence $(a_j)_{j=0}^{\infty}$ is said to be ϵ-close to a number b if there exists a number $N \geq 0$ (it can be very large), such that for all $n \geq N$,*

$$|a_j - b| \leq \epsilon.$$

A sequence $(a_j)_{j=0}^{\infty}$ is said to converge to b if it is ϵ-close to b for all $\epsilon > 0$ (however small). We then write $a_j \to b$, or $\lim_{j\to\infty} a_j = b$.

If a sequence does not converge we say that it diverges. Unbounded sequences, i.e., sequences that contain arbitrarily large numbers, always diverge. (So we never say "converges to infinity", although it's fine to say "diverges to infinity".)

Examples:

- $e^{-n} \to 0$ as $n \to \infty$, and convergence is very fast.
- $n/(n+2) \to 1$ as $n \to \infty$, and convergence is rather slow.
- $(-1)^n$ is bounded, but does not converge.
- $log(n) \to \infty$ as $n \to \infty$, so the sequence diverges. For a proof that $\log(n)$ takes on arbitrarily large values, fix any large integer m. Does there exist an n such that $\log(n) \geq m$? Yes, it suffices to take $n \geq e^m$.

Definition 2. *Consider a sequence $(a_j)_{j=0}^{\infty}$. We define the N-th partial sum S_N as*

$$S_N = a_0 + a_1 + \ldots + a_N = \sum_{j=0}^{N} a_j.$$

We say that the series $\sum_j a_j$ converges if the sequence of partial sums S_N converges to some number b as $N \to \infty$. We then write $\sum_{j=0}^{\infty} a_j = b$.

Again, if a series does not converge we say that it diverges.

Example 1. *Consider* $\sum_{j=0}^{\infty} 2^{-j}$*, i.e.,*

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots.$$

This series converges to the limit 2. To prove this, consider the partial sum

$$S_N = \sum_{j=0}^{N} 2^{-j}.$$

Let us show by induction that $S_N = 2 - 2^{-N}$*. The base case* $N = 0$ *is true since* $2^{-0} = 2 - 2^{-0}$*. For the induction case, assume* $S_N = 2 - 2^{-N}$*. We then write*

$$S_{N+1} = S_N + 2^{-(N+1)} = (2 - 2^{-N}) + 2^{-(N+1)} = 2 - 2^{-(N+1)},$$

the desired conclusion.

Example 2. *The previous example was the* $x = 1/2$ *special case of the so-called geometric series*

$$1 + x + x^2 + x^3 + \dots$$

WIth a similar argument, we obtain the limit as

$$\sum_{j=0}^{\infty} x^j = \frac{1}{1-x},$$

provided the condition $|x| < 1$ *holds. This expression can also be seen as the Taylor expansion of* $1/(1-x)$*, centered at zero, and with radius of convergence 1.*

Example 3. *Consider the so-called harmonic series*

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots.$$

This series diverges. To see this, let us show that the N *partial sum is comparable to* $\log(N)$*. We use the integral test*

$$S_N = \sum_{j=1}^{N} \frac{1}{j} \geq \int_1^{N+1} \frac{1}{x}\, dx.$$

(Insert picture here)

The latter integral is $\log(N+1)$*, which diverges as a sequence. The partial sums, which are larger, must therefore also diverge.*

Example 4. *Consider*

$$\sum_{j=1}^{\infty} \frac{1}{n^q},$$

for some $q > 0$. As a function of q, this is the Riemann zeta function $\zeta(q)$. (A fascinating object for number theorists.)

We've seen above that $q = 1$ leads to divergence. A similar integral test would show that the series converges when $q > 1$, while it diverges when $q \leq 1$.

We now switch to a finer understanding of convergence: certain series are absolutely convergent, while others are conditionally convergent. This will affect what type of algebraic manipulations can be done on them.

Definition 3. *A series $\sum_{j=0}^{\infty} a_j$ is said to be absolutely convergent if $\sum_{j=0}^{\infty} |a_j|$ converges. If a series is not absolutely convergent, but nevertheless converges, we say that it is conditionally convergent.*

The subtlety with conditional convergence is that alternating plus and minus signs may lead to convergence because of cancelations when summing consecutive terms.

Example 5. *Consider*

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \ldots.$$

This series is not absolutely convergent, because it reduces to the harmonic series when absolute values of the terms are taken. It is nevertheless convergent, hence conditionally convergent, as the following argument shows. Assume N is even, and let

$$S_N = \sum_{j=1}^{N} \frac{(-1)^j}{j}.$$

Terms can be grouped 2-by-2 to obtain, for $j \geq 1$,

$$\frac{1}{j} - \frac{1}{j+1} = \frac{1}{j(j+1)}.$$

A fortiori, $\frac{1}{j(j+1)} \leq \frac{1}{j^2}$, *so*

$$S_N \leq \sum_{j=1,3,5,\ldots}^{N-1} \frac{1}{j^2},$$

which we know converges. If on the other hand N *is odd, then* $S_N = S_{N-1} + \frac{1}{N+1}$. *Both terms* S_N *and* $1/(N+1)$ *are converging sequences, so their sum converges as well. This proves convergence.*

Note that the series converges to

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \ldots = \log(2).$$

This is the special case $x = 1$ *in the Taylor expansion of* $\log(1+x)$ *about* $x = 0$.

In passing, without proof, here is a useful test to check convergence of alternating series.

Theorem 1. *(Alternating series test) Consider the series*

$$\sum_{j=0}^{\infty} (-1)^j a_j,$$

where $a_j > 0$. *If* (a_j) *converges to zero (as a sequence), then the series is convergent.*

The main problem with conditionally convergent series is that if the terms are rearranged, then the series may converge to a different limit. The "safe zone" for handling infinite sums as if they were finite is when convergence is absolute.

Theorem 2. *Let* $f : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ *be a bijection, i.e.,* f *is a rearrangement of the nonnegative integers. Consider a series* $\sum_{j=0}^{\infty} a_j$. *If this series is absolutely convergent, then*

$$\sum_{j=0}^{\infty} a_j = \sum_{j=0}^{\infty} a_{f(j)}.$$

Here is what usually happens when the assumption of absolute convergence is not satisfied.

Example 6. *Consider again*

$$\frac{1}{3}-\frac{1}{4}+\frac{1}{5}-\frac{1}{6}+\dots$$

which as we have seen equals $\log(2)-(1-1/2)=\log(2)-1/2=.193147\dots$. *We can rearrange the terms of the series by assigning two negative terms for each positive term:*

$$\frac{1}{3}-\frac{1}{4}-\frac{1}{6}+\frac{1}{5}-\frac{1}{8}-\frac{1}{10}+\frac{1}{7}+\dots$$

This series is also convergent, but happens to converge to $(\log(2)-1)/2 = -.153426\dots$.

Other operations that can be safely performed on absolutely convergent series are passing absolute values inside the sum, and exchanging sums. Again, complications arise if the series is only conditionally convergent. (See Tao, Analysis I, for counter-examples.)

Theorem 3. *The following operations are legitimate for absolutely convergent series.*

- *Passing absolute values inside sums:*

$$|\sum_{j=0}^{\infty} a_j| \le \sum_{j=0}^{\infty} |a_j|.$$

- *Swapping sums:*

$$\sum_{j=0}^{\infty}\sum_{k=0}^{\infty} a_{j,k} = \sum_{k=0}^{\infty}\sum_{j=0}^{\infty} a_{j,k}$$

Note in passing that the same is true for integrals of unbounded integrands or integrals over unbounded domains: they need to be absolutely convergent (integrability of the absolute value of the function) for the integral swap to be legitimate. This is the content of Fubini's theorem. Again, there are striking counter-examples when the integrals are not absolutely convergent and the swap doesn't work (See Tao, Analysis I).

1.2 The big-O notation

Here are a few useful pieces of notation for comparing growth or decay of sequences, used extensively by numerical analysts. They are called the big-O, little-o, and big-Theta notations. Big-O is used much more often than the other two. They occur when comparing decay rates of truncation errors, and runtimes of algorithms.

Definition 4. *Consider two nonzero sequences f_n and g_n for $n = 0, 1, 2, \ldots$.*
We write $f_n = O(g_n)$ when there exists $C > 0$ such that $|f_n| \leq C|g_n|$.
We write $f_n = o(g_n)$ when $f_n/g_n \to 0$ as $n \to \infty$.
We write $f_n = \Theta(g_n)$ when $f_n = O(g_n)$ and $g_n = O(f_n)$.

Examples:

- $f_n = n^2$ and $g_n = n^3$: we have $n^2 = O(n^3)$ and $n^2 = o(n^3)$ but $n^2 \neq \Theta(n^3)$.
- $f_n = \frac{n}{n+2}$ and $g_n = \frac{n}{n-3}$: we have $f_n = O(g_n)$ and $f_n = \Theta(g_n)$, but $f_n \neq o(g_n)$.
- Exponentials always dominate polynomials: $n^a = o(e^{bn})$ whenever $a > 0$ and $b > 0$.
- Conversely, $e^{-bn} = o(n^{-a})$.

Out loud, we can read the expression $f_n = O(g_n)$ as "f_n is on the order of g_n".

The same notations can be used to compare sequences indexed by a parameter that goes to zero, such as (typically) the grid spacing h. The definition above is simply adapted by letting $h \to 0$ rather than $n \to \infty$.

Examples:

- $f(h) = h^2$ and $g(h) = h^3$: this time we have $g(h) = O(f(h))$ and $g(h) = o(f(h))$ when the limit of interest is $h \to 0$.
- Powers of h don't converge to zero nearly as fast as this exponential: $e^{a/h} = o(h^b)$ whatever $a > 0$ and $b > 0$.

Similarly, we may wish to compare functions f and g of a continuous variable x as either $x \to \infty$ or $x \to 0$; the definition is again modified in the obvious way. Whenever a $O(\cdot)$ or $o(\cdot)$ is written, some underlying limit is understood.

Chapter 2

Integrals as sums and derivatives as differences

We now switch to the simplest methods for integrating or differentiating a function from its function samples. A careful study of Taylor expansions reveals how accurate the constructions are.

2.1 Numerical integration

Consider a function f – we'll specify which assumptions we need to make about it in a minute. Let us reformulate the integral

$$\int_0^1 f(x)\, dx$$

by breaking up the interval $[a, b]$ into subintervals $[x_{j-1}, x_j]$, with $x_j = jh$, $h = 1/N$, and

$$0 = x_0 < x_1 < \ldots < x_N = 1.$$

Together, the (x_j) are called a Cartesian grid. Still without approximation, we can write

$$\int_0^1 f(x)\, dx = \sum_{j=1}^{N} \int_{x_{j-1}}^{x_j} f(x)\, dx.$$

A *quadrature* is obtained when integrals are approximated by quantities that depend only on the samples $f(x_j)$ of f on the grid x_j. For instance, we can approximate the integral over $[x_{j-1}, x_j]$ by the signed area of the rectangle of height $f(x_{j-1})$ and width h:

$$\int_{x_{j-1}}^{x_j} f(x)\,dx \simeq hf(x_{j-1}).$$

Putting the terms back together, we obtain the *rectangle method*:

$$\int_0^1 f(x)\,dx \simeq h\sum_{j=1}^{N} f(x_{j-1}). \tag{2.1}$$

(Insert picture here)

To understand the accuracy of this approximation, we need a detour through Taylor expansions. Given a very smooth function $f(x)$, it sometimes makes sense to write an expansion around the point $x = a$, as

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \frac{1}{3!}f'''(a)(x-a)^3 + \dots$$

As a compact series, this would be written as

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(a)(x-a)^n,$$

where $n! = n\cdot(n-1)\cdot\ldots 2\cdot 1$ is the factorial of n. The word “sometimes” is important: the formula above requires the function to be infinitely differentiable, and even still, we would need to make sure that the series converges (by means of the analysis of the previous chapter). The radius of convergence of the Taylor expansion is the largest R such that, for all x such that $|x-a| < R$, the series converges.

It is also possible to truncate a Taylor series after N terms, as

$$f(x) = \sum_{n=0}^{N-1} \frac{1}{n!} f^{(n)}(a)(x-a)^n + \frac{1}{N!} f^{(N)}(y)(x-a)^N, \tag{2.2}$$

where y is some number between x and a. The formula does not specify what the number y is; only that there exists one such that the remainder takes the form of the last term. In spite of the fact that y is unspecified, this is a much more useful form of Taylor expansion than the infinite series:

- We often only care about the size of the remainder, and write inequalities such as

$$|\frac{1}{N!}f^{(N)}(y)(x-a)^N| \leq \frac{1}{N!}\left(\max_{z\in[x,a]}|f^{(N)}(z)|\right)|x-a|^N,$$

  or, for short,

$$\frac{1}{N!}f^{(N)}(y)(x-a)^N = O(|x-a|^N),$$

  where all the "constants" are hidden in the O. So, the remainder is on the order of $|x-a|^N$. The limit implicit in the O notation is here $(x-a)\to 0$.

- The finite-N formula is valid whenever the function is N times differentiable, not infinitely differentiable! In fact, the Taylor series itself may diverge, but equation (2.2) is still valid.

In order to study the problem of approximating $\int_{x_{j-1}}^{x_j} f(x)\,dx$, let us expand f in a (short, truncated) Taylor series near $x = x_{j-1}$:

$$f(x) = f(x_{j-1}) + f'(y(x))(x-x_{j-1}),$$

where $y(x)\in[x_{j-1},x]$. We've written $y(x)$ to highlight that it depends on x. This works as long as f has one derivative. Integrating on both sides, and recognizing that $f(x_{j-1})$ is constant with respect to x, we get

$$\int_{x_{j-1}}^{x_j} f(x)\,dx = hf(x_{j-1}) + \int_{x_{j-1}}^{x_j} f'(y(x))(x-x_{j-1})\,dx.$$

We don't know much about $y(x)$, but we can certainly write the inequality (recall that $x_j - x_{j-1} = h$)

$$|\int_{x_{j-1}}^{x_j} f'(y(x))(x-x_{j-1})\,dx| \leq \max_{y\in[x_{j-1},x_j]}|f'(y)|\,h\int_{x_{j-1}}^{x_j} 1\,dx = h^2\max_{y\in[x_{j-1},x_j]}|f'(y)|.$$

So long as f is differentiable, we have

$$\int_{x_{j-1}}^{x_j} f(x)\,dx = hf(x_{j-1}) + O(h^2),$$

where the derivative of f hides in the O sign. The integral over $[0, 1]$ has N such terms, so when they are added up the error compounds to $N \cdot O(h^2) = O(h)$ (because $h = 1/N$). Thus we have just proved that

$$\int_0^1 f(x)\, dx = h \sum_{j=1}^{N} f(x_{j-1}) + O(h).$$

The error is on the order of the grid spacing h itself, so we say the method is first-order accurate (because the exponent of h is one.)

Choosing the (signed) height of each rectangle from the left endpoint x_{j-1} does not sound like the most accurate thing to do. Evaluating the function instead at $x_{j-1/2} = (j - 1/2)h$ is a better idea, called the midpoint method. It is possible to show that

$$\int_0^1 f(x)\, dx = h \sum_{j=1}^{N} f(x_{j-1/2}) + O(h^2),$$

provided the function f is twice differentiable (because f'' hides in the O sign). The accuracy is improved, since h^2 gets much smaller than h as $h \to 0$. We say the midpoint method is second-order accurate.

Another solution to getting order-2 accuracy is to consider trapezoids instead of rectangles for the interval $[x_{j-1}, x_j]$. The area of the trapezoid spanned by the 4 points $(x_{j-1}, 0)$; $(x_{j-1}, f(x_{j-1}))$; $(x_j, f(x_j))$; $(x_j, 0)$ is $h(f(x_{j-1}) + f(x_j))/2$. This gives rise to the so-called trapezoidal method, or trapezoidal rule.

(Insert picture here)

Let us compare this quantity to the integral of f over the same interval. Consider the truncated Taylor expansions

$$f(x) = f(x_{j-1}) + f'(x_{j-1})(x - x_{j-1}) + O(h^2),$$

$$f(x_j) = f(x_{j-1}) + f'(x_{j-1})h + O(h^2),$$

where the second derivative of f appears in the O sign. Integrating the first relation gives

$$\int_{x_{j-1}}^{x_j} f(x)\, dx = hf(x_{j-1}) + \frac{h^2}{2} f'(x_{j-1}) + O(h^3).$$

The area of the trapezoid, on the other hand, is (using the second Taylor relation)

$$\frac{h}{2}(f(x_{j-1}) + f(x_j)) = hf(x_{j-1}) + \frac{h^2}{2}f'(x_{j-1}) + O(h^3).$$

Those two equations agree, except for the terms $O(h^3)$ which usually differ in the two expressions. Hence

$$\int_{x_{j-1}}^{x_j} f(x)\,dx = \frac{h}{2}(f(x_{j-1}) + f(x_j)) + O(h^3).$$

We can now sum over j (notice that all the terms appear twice, except for the endpoints) to obtain

$$\int_0^1 f(x)\,dx = \frac{h}{2}f(0) + h\sum_{j=1}^{N-1} f(x_j) + \frac{h}{2}f(1) + O(h^2).$$

The trapezoidal rule is second-order accurate. All it took is a modification of the end terms to obtain $O(h^2)$ accuracy in place of $O(h)$.

Example: $f(x) = x^2$ in $[0,1]$. We have $\int_0^1 x^2\,dx = 1/3$.

- For the rectangle rule with $N = 4$ and $h = 1/4$, consider the gridpoints $x = 0, 1/4, 1/2, 3/4$, and 1. Each rectangle has height $f(x_{j-1})$ where x_{j-1} is the left endpoint. We have

  $$\int_0^1 x^2\,dx \simeq \frac{1}{4}\left[0 + \frac{1}{4^2} + \frac{1}{2^2} + \frac{3^2}{4^2}\right] = \frac{14}{64} = .2188...$$

  This is quite far ($O(h)$, as we know) from 1/3.

- For the trapezoidal rule, consider the same grid. We now also consider the grid point at $x = 1$, but the contribution of both $x = 0$ and $x = 1$ is halved.

  $$\int_0^1 x^2\,dx = \frac{1}{4}\left[\frac{1}{2}\cdot 0 + \frac{1}{4^2} + \frac{1}{2^2} + \frac{3^2}{4^2} + \frac{1}{2}\cdot 1\right] = \frac{22}{64} = .3438...$$

  This is much closer ($O(h^2)$ as a matter of fact) from 1/3.

We'll return to the topic of numerical integration later, after we cover polynomial interpolation.

2.2 Numerical differentiation

The simplest idea for computing an approximation to the derivative $u'(x_j)$ of a function u from the samples $u_j = u(x_j)$ is to form finite difference ratios. On an equispaced grid $x_j = jh$, the natural candidates are:

- the one-sided, forward and backward differences

$$\Delta_+ u_j = \frac{u_{j+1} - u_j}{h}, \qquad \Delta_- = \frac{u_j - u_{j-1}}{h},$$

- and the two-sided centered difference

$$\Delta_0 u_j = \frac{u_{j+1} - u_{j-1}}{2h}.$$

Let us justify the accuracy of these difference quotients at approximating derivatives, as $h \to 0$. The analysis will depend on the smoothness of the underlying function u.

Assume without loss of generality that $x_{j-1} = -h$, $x_j = 0$, and $x_{j+1} = h$. For the forward difference, we can use a Taylor expansion about zero to get

$$u(h) = u(0) + hu'(0) + \frac{h^2}{2} u''(y), \qquad y \in [0, h].$$

When substituted in the formula for the forward difference, we get

$$\frac{u(h) - u(0)}{h} = u'(0) + \frac{h}{2} u''(y).$$

The error is a $O(h)$ as soon as the function has two bounded derivatives. We say that the forward difference is a method of order one. The analysis for the backward difference is very similar.

For the centered difference, we now use two Taylor expansions about zero, for $u(h)$ and $u(-h)$, that we write up to order 3:

$$u(\pm h) = u(0) \pm hu'(0) + \frac{h^2}{2} u''(0) \pm \frac{h^3}{6} u'''(y_\pm),$$

with $y_\pm$ either in $[0, h]$ or $[-h, 0]$ depending on the choice of sign. Subtract $u(-h)$ from $u(h)$ to get (the h^2 terms cancel out)

$$\frac{u(h) - u(-h)}{2h} = u'(0) + \frac{h^2}{12} (u'''(y_+) + u'''(y_-)).$$

The error is now $O(h^2)$ provided u is three times differentiable, hence the method is of order 2.

The simplest choice for the second derivative is the centered second difference

$$\Delta^2 u_j = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2}.$$

It turns out that $\Delta^2 = \Delta_+\Delta_- = \Delta_-\Delta_+$, i.e., the three-point formula for the second difference can be obtained by calculating the forward difference of the backward difference, or vice-versa. Δ^2 is second-order accurate: the error is $O(h^2)$. To see this, write the Taylor expansions around $x = 0$ to fourth order:

$$u(\pm h) = u(0) \pm hu'(0) + \frac{h^2}{2}u''(0) \pm \frac{h^3}{6}u'''(0) + \frac{h^4}{24}u''''(y_\pm).$$

with $y_\pm$ either in $[0, h]$ or $[-h, 0]$ depending on the choice of sign. The odd terms all cancel out when calculating the second difference, and what is left is

$$\frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} = u''(0) + \frac{h^2}{24}(u''''(y_+) + u''''(y_-)).$$

So the method is $O(h^2)$, but only when u is four times differentiable.

Example: $f(x) = x^2$ again. We get $f'(x) = 2x$ and $f''(x) = 2$.

$$\Delta_+ f(x) = \frac{(x+h)^2 - x^2}{h} = \frac{2xh + h^2}{h} = 2x + h.$$

$$\Delta_- f(x) = \frac{x^2 - (x-h)^2}{h} = \frac{2xh - h^2}{h} = 2x - h.$$

$$\Delta_0 f(x) = \frac{(x+h)^2 - (x-h)^2}{2h} = \frac{4xh}{2h} = 2x.$$

$$\Delta^2 f(x) = \frac{(x+h)^2 - 2x^2 + (x-h)^2}{h^2} = \frac{2h^2}{h^2} = 2.$$

The error is manifestly h for the forward difference, and $-h$ for the backward difference (both are $O(h)$.) Coincidentally, the error is zero for Δ_0 and Δ^2: centered differences differentiate parabolas exactly. This is an exception: finite differences are never exact in general. Of course, $0 = O(h^2)$ so there's no contradiction.

When u has less differentiability than is required by the remainder in the Taylor expansions, it may happen that the difference schemes may have lower order than advertised. This happens in numerical simulations of some differential equations where the solution is discontinuous, e.g., in fluid dynamics (interfaces between phases).

Sometimes, points on the left or on the right of x_j are not available because we are at one of the edges of the grid. In that case one should use a one-sided difference formula (such as the backward or forward difference.) You can find tables of high-order one-sided formulas for the first, second, and higher derivatives online.

The more systematic way of deriving finite difference formulas is by differentiating a polynomial interpolant. We return to the topics of integration and differentiation in the next chapter.

Chapter 3

Interpolation

Interpolation is the problem of fitting a smooth curve through a given set of points, generally as the graph of a function. It is useful at least in data analysis (interpolation is a form of regression), industrial design, signal processing (digital-to-analog conversion) and in numerical analysis. It is one of those important recurring concepts in applied mathematics. In this chapter, we will immediately put interpolation to use to formulate high-order quadrature and differentiation rules.

3.1 Polynomial interpolation

Given $N+1$ points $x_j \in \mathbb{R}$, $0 \le j \le N$, and sample values $y_j = f(x_j)$ of a function at these points, the polynomial interpolation problem consists in finding a polynomial $p_N(x)$ of degree N which reproduces those values:

$$y_j = p_N(x_j), \qquad j = 0, \ldots, N.$$

In other words the graph of the polynomial should pass through the points (x_j, y_j). A degree-N polynomial can be written as $p_N(x) = \sum_{n=0}^{N} a_n x^n$ for some coefficients $a_0, \ldots, a_N$. For interpolation, the number of degrees of freedom ($N+1$ coefficients) in the polynomial matches the number of points where the function should be fit. If the degree of the polynomial is strictly less than N, we cannot in general pass it through the points (x_j, y_j). We can still try to pass a polynomial (e.g., a line) in the "best approximate manner", but this is a problem in approximation rather than interpolation; we will return to it later in the chapter on least-squares.

Let us first see how the interpolation problem can be solved numerically in a direct way. Use the expression of p_N into the interpolating equations $y_j = p_N(x_j)$:

$$\sum_{n=0}^{N} a_n x_j^n = y_j, \qquad j = 0, \ldots, N.$$

In these $N+1$ equations indexed by j, the unknowns are the coefficients $a_0, \ldots, a_N$. We are in presence of a linear system

$$V\mathbf{a} = \mathbf{y}, \qquad \Leftrightarrow \qquad \sum_{n=0}^{N} V_{jn} a_n = y_j,$$

with V the so-called Vandermonde matrix, $V_{jn} = x_j^n$, i.e.,

$$V = \begin{pmatrix} 1 & x_0 & \cdots & x_0^N \\ 1 & x_1 & \cdots & x_1^N \\ \vdots & \vdots & & \vdots \\ 1 & x_N & \cdots & x_N^N \end{pmatrix}.$$

We can then use a numerical software like Matlab to construct the vector of abscissas x_j, the right-hand-side of values y_j, the V matrix, and numerically solve the system with an instruction like $\mathbf{a} = V \setminus \mathbf{y}$ (in Matlab). This gives us the coefficients of the desired polynomial. The polynomial can now be plotted in between the grid points x_j (on a finer grid), in order to display the interpolant.

Historically, mathematicians such as Lagrange and Newton did not have access to computers to display interpolants, so they found explicit (and elegant) formulas for the coefficients of the interpolation polynomial. It not only simplified computations for them, but also allowed them to understand the error of polynomial interpolation, i.e., the difference $f(x) - p_N(x)$. Let us spend a bit of time retracing their steps. (They were concerned with applications such as fitting curves to celestial trajectories.)

We'll define the interpolation error from the uniform (L^∞) norm of the difference $f - p_N$:

$$\|f - p_N\|_\infty := \max_x |f(x) - p_N(x)|,$$

where the maximum is taken over the interval $[x_0, x_N]$.

Call P_N the space of real-valued degree-N polynomials:

$$P_N = \{\sum_{n=0}^{N} a_n x^n \,:\, a_n \in \mathbb{R}\}.$$

Lagrange's solution to the problem of polynomial interpolation is based on the following construction.

Lemma 1. *(Lagrange elementary polynomials) Let $\{x_j, j = 0, \ldots, N\}$ be a collection of disjoint numbers. For each $k = 0, \ldots, N$, there exists a unique degree-N polynomial $L_k(x)$ such that*

$$L_k(x_j) = \delta_{jk} = \begin{cases} 1 & \text{if } j = k; \\ 0 & \text{if } j \neq k. \end{cases}$$

Proof. Fix k. L_k has roots at x_j for $j \neq k$, so L_k must be of the form[1]

$$L_k(x) = C \prod_{j \neq k} (x - x_j).$$

Evaluating this expression at $x = x_k$, we get

$$1 = C \prod_{j \neq k} (x_k - x_j) \qquad \Rightarrow \qquad C = \frac{1}{\prod_{j \neq k}(x_k - x_j)}.$$

Hence the only possible expression for L_k is

$$L_k(x) = \frac{\prod_{j \neq k}(x - x_j)}{\prod_{j \neq k}(x_k - x_j)}.$$

□

These elementary polynomials form a basis (in the sense of linear algebra) for expanding any polynomial interpolant p_N.

[1] That's because, if we fix j, we can divide $L_k(x)$ by $(x - x_j)$, $j \neq k$. We obtain $L_k(x) = (x - x_j)q(x) + r(x)$, where $r(x)$ is a remainder of lower order than $x - x_j$, i.e., a constant. Since $L_k(x_j) = 0$ we must have $r(x) = 0$. Hence $(x - x_j)$ must be a factor of $L_k(x)$. The same is true of any $(x - x_j)$ for $j \neq k$. There are N such factors, which exhausts the degree N. The only remaining degree of freedom in L_k is the multiplicative constant.

Theorem 4. *(Lagrange interpolation theorem)*

Let $\{x_j, j = 0, \dots, N\}$ be a collection of disjoint real numbers. Let $\{y_j, j = 0, \dots, N\}$ be a collection of real numbers. Then there exists a unique $p_N \in P_N$ such that

$$p_N(x_j) = y_j, \qquad j = 0, \dots, N.$$

Its expression is

$$p_N(x) = \sum_{k=0}^{N} y_k L_k(x), \tag{3.1}$$

where $L_k(x)$ are the Lagrange elementary polynomials.

Proof. The justification that (3.1) interpolates is obvious:

$$p_N(x_j) = \sum_{k=0}^{N} y_k L_k(x_j) = \sum_{k=0}^{N} y_k \delta_{jk} = y_j.$$

It remains to see that p_N is the unique interpolating polynomial. For this purpose, assume that both p_N and q_N take on the value y_j at x_j. Then $r_N = p_N - q_N$ is a polynomial of degree N that has a root at each of the $N+1$ points $x_0, \dots, x_N$. The fundamental theorem of algebra, however, says that a nonzero polynomial of degree N can only have N (complex) roots. Therefore, the only way for r_N to have $N + 1$ roots is that it is the zero polynomial. So $p_N = q_N$.

□

By definition,

$$p_N(x) = \sum_{k=0}^{N} f(x_k) L_k(x)$$

is called the *Lagrange interpolation polynomial* of f at x_j.

Example 7. *Linear interpolation through (x_1, y_1) and (x_2, y_2):*

$$L_1(x) = \frac{x - x_2}{x_1 - x_2}, \qquad L_2(x) = \frac{x - x_1}{x_2 - x_1},$$

$$\begin{aligned} p_1(x) &= y_1 L_1(x) + y_2 L_2(x) \\ &= \frac{y_2 - y_1}{x_2 - x_1} x + \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1} \\ &= y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1). \end{aligned}$$

Example 8. *(Example (6.1) in Suli-Mayers) Consider $f(x) = e^x$, and interpolate it by a parabola ($N = 2$) from three samples at $x_0 = -1, x_1 = 0, x_2 = 1$. We build*

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{1}{2}x(x - 1)$$

Similarly,

$$L_1(x) = 1 - x^2, \qquad L_2(x) = \frac{1}{2}x(x + 1).$$

So the quadratic interpolant is

$$\begin{aligned} p_2(x) &= e^{-1}L_0(x) + e^0 L_1(x) + e^1 L_2(x), \\ &= 1 + \sinh(1)\, x + (\cosh(1) - 1)\, x^2, \\ &\simeq 1 + 1.1752\, x + 0.5431\, x^2. \end{aligned}$$

Another polynomial that approximates e^x reasonably well on $[-1, 1]$ is the Taylor expansion about $x = 0$:

$$t_2(x) = 1 + x + \frac{x^2}{2}.$$

Manifestly, p_2 is not very different from t_2.

(Insert picture here)

Let us now move on to the main result concerning the interpolation error of smooth functions.

Theorem 5. *Let $f \in C^{N+1}[a, b]$ for some $N > 0$, and let $\{x_j : j = 0, \ldots, N\}$ be a collection of disjoint reals in $[a, b]$. Consider p_N the Lagrange interpolation polynomial of f at x_j. Then for every $x \in [a, b]$ there exists $\xi(x) \in [a, b]$ such that*

$$f(x) - p_N(x) = \frac{f^{(N+1)}(\xi(x))}{(N + 1)!}\pi_{N+1}(x),$$

where

$$\pi_{N+1}(x) = \prod_{j=1}^{N+1}(x - x_j).$$

An estimate on the interpolation error follows directly from this theorem. Set

$$M_{N+1} = \max_{x \in [a,b]} |f^{(N+1)}(x)|$$

(which is well defined since $f^{(N+1)}$ is continuous by assumption, hence reaches its lower and upper bounds.) Then

$$|f(x) - p_N(x)| \le \frac{M_{N+1}}{(N+1)!} |\pi_{N+1}(x)|$$

In particular, we see that the interpolation error is zero when $x = x_j$ for some j, as it should be.

Let us now prove the theorem.

Proof. (Can be found in Suli-Mayers, Chapter 6) □

In conclusion, the interpolation error:

- depends on the smoothness of f via the high-order derivative $f^{(N+1)}$;
- has a factor $1/(N+1)!$ that decays fast as the order $N \to \infty$;
- and is directly proportional to the value of $\pi_{N+1}(x)$, indicating that the interpolant may be better in some places than others.

The natural follow-up question is that of convergence: can we always expect convergence of the polynomial interpolant as $N \to \infty$? In other words, does the factor $1/(N+1)!$ always win over the other two factors?

Unfortunately, the answer is no in general. There are examples of very smooth (analytic) functions for which polynomial interpolation diverges, particularly so near the boundaries of the interplation interval. This behavior is called the *Runge phenomenon*, and is usually illustrated by means of the following example.

Example 9. *(Runge phenomenon) Let $f(x)$ for $x \in [-5, 5]$. Interpolate it at equispaced points $x_j = 10j/N$, where $j = -N/2, \ldots, N/2$ and N is even. It is easy to check numerically that the interpolant diverges near the edges of $[-5, 5]$, as $N \to \infty$. See Trefethen's textbook on page 44 for an illustration of the Runge phenomenon.*

(Figure here)

If we had done the same numerical experiment for $x \in [-1, 1]$, the interpolant would have converged. This shows that the size of the interval matters. Intuitively, there is divergence when the size of the interval is larger than the "features", or characteristic length scale, of the function (here the width of the bump near the origin.)

The analytical reason for the divergence in the example above is due in no small part to the very large values taken on by $\pi_{N+1}(x)$ far away from the origin — in contrast to the relatively small values it takes on near the origin. This is a problem intrinsic to equispaced grids. We will be more quantitative about this issue in the section on Chebyshev interpolants, where a remedy involving non-equispaced grid points will be explained.

As a conclusion, polynomial interpolants can be good for small N, and on small intervals, but may fail to converge (quite dramatically) when the interpolation interval is large.

3.2 Polynomial rules for integration

In this section, we return to the problem of approximating $\int_a^b u(x)dx$ by a weighted sum of samples $u(x_j)$, also called a quadrature. The plan is to form interpolants of the data, integrate those interpolants, and deduce corresponding quadrature formulas. We can formulate rules of arbitrarily high order this way, although in practice we almost never go beyond order 4 with polynomial rules.

3.2.1 Polynomial rules

Without loss of generality, consider the local interpolants of $u(x)$ formed near the origin, with $x_0 = 0, x_1 = h$ and $x_{-1} = -h$. The rectangle rule does not belong in this section: it is not formed from an interpolant.

- The trapezoidal rule, where we approximate $u(x)$ by a line joining $(0, u(x_0))$ and $(h, u(x_1))$ in $[0, h]$. We need 2 derivatives to control the error:

$$u(x) = p_1(x) + \frac{u''(\xi(x))}{2} x(x-h),$$

$$p_1(x) = u(0)L_0(x) + u(h)L_1(x),$$

$$L_0(x) = \frac{h-x}{h}, \qquad L_1(x) = \frac{x}{h},$$

$$\int_0^h L_0(x)dx = \int_0^h L_1(x)dx = h/2, \qquad \text{(areas of triangles)}$$

$$\int_0^h \left|\frac{u''(\xi(x))}{2} x(x-h)\right| dx \leq C \max_{\xi} |u''(\xi)| h^3.$$

The result is

$$\int_0^h u(x)\,dx = h\left(\frac{u(0)+u(h)}{2}\right) + O(h^3).$$

As we have seen, the terms combine as

$$\int_0^1 u(x)\,dx = \frac{h}{2}u(x_0) + h\sum_{j=1}^{N-1} u(x_{j-1}) + \frac{h}{2}u(x_N) + O(h^2).$$

- Simpson's rule, where we approximate $u(x)$ by a parabola through $(-h, u(x_{-1}))$, $(0, u(x_0))$, and $(h, u(x_1))$ in $[-h, h]$. We need three derivatives to control the error:

$$u(x) = p_2(x) + \frac{u'''(\xi(x))}{6}(x+h)x(x-h),$$

$$p_2(x) = u(-h)L_{-1}(x) + u(0)L_0(x) + u(h)L_1(x),$$

$$L_{-1}(x) = \frac{x(x-h)}{2h^2}, \qquad L_0(x) = \frac{(x+h)(x-h)}{-h^2}, \qquad L_1(x) = \frac{(x+h)x}{2h^2},$$

$$\int_{-h}^h L_{-1}(x)dx = \int_{-h}^h L_1(x)dx = h/3, \qquad \int_{-h}^h L_0(x)dx = 4h/3,$$

$$\int_{-h}^h |\frac{u'''(\xi(x))}{6}(x+h)x(x-h)|\,dx \le C\max_\xi |u'''(\xi)|h^4.$$

The result is

$$\int_{-h}^h u(x)\,dx = h\left(\frac{u(-h)+4u(0)+u(h)}{3}\right) + O(h^4).$$

The composite Simpson's rule is obtained by using this approximation on $[0, 2h] \cup [2h, 4h] \cup \ldots [1-2h, 1]$, adding the terms, and recognizing that the samples at $2nh$ (except 0 and 1) are represented twice.

$$\int_0^1 u(x)dx = \frac{h}{3}\left(u(0) + 4u(h) + 2u(2h) + 4u(3h) + \ldots\right.$$
$$\left. + 2u(1-2h) + 4u(1-h) + u(1)\right) + O(h^3).$$

It turns out that the error is in fact $O(h^5)$ on $[-h, h]$, and $O(h^4)$ on $[0, 1]$, as a homework question invites to prove. For this, we need u to be four times differentiable (the constant in front of h^5 involves u''''.)

The higher-order variants of polynomial rules, called Newton-Cotes rules, are not very interesting because the Runge phenomenon kicks in again. Also, the weights (like 1,4,2,4,2, etc.) become negative, which leads to unacceptable error magnification if the samples of u are not known exactly.

Piecewise spline interpolation is not a good choice for numerical integration either, because of the two leftover degrees of freedom, whose arbitrary choice affects the accuracy of quadrature in an unacceptable manner. (We'll study splines in a later section.)

We'll return to (useful!) integration rules of arbitrarily high order in the scope of spectral methods.

3.3 Polynomial rules for differentiation

A systematic way of deriving finite difference formulas of higher order is to view them as derivatives of a polynomial interpolant passing through a small number of points neighboring x_j. For instance (again we use $-h, 0, h$ as reference points without loss of generality):

- The forward difference at 0 is obtained from the line joining $(0, u(0))$ and $(h, u(h))$:
$$p_1(x) = u(0)L_0(x) + u(h)L_1(x),$$
$$L_0(x) = \frac{h-x}{h}, \qquad L_1(x) = \frac{x}{h},$$
$$p_1'(0) = \frac{u(h) - u(0)}{h}.$$
We already know that $u'(0) - p_1'(0) = O(h)$.

- The centered difference at 0 is obtained from the line joining $(-h, u(-h))$ and $(h, u(h))$ (a simple exercise), but it is also obtained from differentiating the parabola passing through the points $(-h, u(-h))$, $(0, u(0))$, and $(h, u(h))$. Indeed,
$$p_2(x) = u(-h)L_{-1}(x) + u(0)L_0(x) + u(h)L_1(x),$$
$$L_{-1}(x) = \frac{x(x-h)}{2h^2}, \qquad L_0(x) = \frac{(x+h)(x-h)}{-h^2}, \qquad L_1(x) = \frac{(x+h)x}{2h^2},$$
$$p_2'(x) = u(-h)\frac{2x-h}{2h^2} + u(0)\frac{2x}{-h^2} + u(h)\frac{2x+h}{2h^2},$$

$$p_2'(0) = \frac{u(h) - u(-h)}{2h}.$$

We already know that $u'(0) - p_2'(0) = O(h^2)$.

- Other examples can be considered, such as the centered second difference (-1 2 -1), the one-sided first difference (-3 4 -1), etc.

Differentiating one-sided interpolation polynomials is a good way to obtain one-sided difference formulas, which comes in handy at the boundaries of the computational domain. The following result establishes that the order of the finite difference formula matches the order of the polynomial being differentiated.

Theorem 6. *Let $f \in C^{N+1}[a,b]$, $\{x_j, j = 0, \ldots, N\}$ some disjoint points, and p_N the corresponding interpolation polynomial. Then*

$$f'(x) - p_N'(x) = \frac{f^{(N+1)}(\xi)}{N!}\,\pi_N^*(x),$$

for some $\xi \in [a,b]$, and where $\pi_N^*(x) = (x - \eta_1)\ldots(x - \eta_N)$ for some $\eta_j \in [x_{j-1}, x_j]$.

The proof of this result is an application of Rolle's theorem that we leave out. (It is not simply a matter of differentiating the error formula for polynomial interpolation, because we have no guarantee on $d\xi/dx$.)

A consequence of this theorem is that the error in computing the derivative is a $O(h^N)$ (which comes from a bound on the product $\pi_N^*(x)$.)

It is interesting to notice, at least empirically, that the Runge's phenomenon is absent when the derivative is evaluated at the center of the interval over which the interpolant is built.

3.4 Piecewise polynomial interpolation

The idea of piecewise polynomial interpolation, also called spline interpolation, is to subdivide the interval $[a,b]$ into a large number of subintervals $[x_{j-1}, x_j]$, and to use low-degree polynomials over each subintervals. This helps avoiding the Runge phenomenon. The price to pay is that the interpolant is no longer a C^∞ function – instead, we lose differentiability at the junctions between subintervals, where the polynomials are made to match.

If we use polynomials of order n, then the interpolant is piecewise C^∞, and overall C^k, with $k \leq n-1$. We could not expect to have $k = n$, because it would imply that the polynomials are identical over each subinterval. We'll see two examples: linear splines when $n = 1$, and cubic splines when $n = 3$. (We have seen in the homework why it is a bad idea to choose $n = 2$.)

3.4.1 Linear splines

We wish to interpolate a continuous function $f(x)$ of $x \in [a, b]$, from the knowledge of $f(x_j)$ at some points x_j, $j = 0, \ldots, N$, not necessarily equispaced. Assume that $x_0 = a$ and $x_N = b$. The piecewise linear interpolant is build by tracing a straight line between the points $(x_{j-1}, f(x_{j-1}))$ and $(x_j, f(x_j)))$; for $j = 1, \ldots, N$ the formula is simply

$$s_L(x) = \frac{x_j - x}{x_j - x_{j-1}} f(x_{j-1}) + \frac{x - x_{j-1}}{x_j - x_{j-1}} f(x_j), \qquad x \in [x_{j-1}, x_j].$$

In this case we see that $s_L(x)$ is a continuous function of x, but that its derivative is not continuous at the junction points, or nodes x_j.

If the function is at least twice differentiable, then piecewise linear interpolation has second-order accuracy, i.e., an error $O(h^2)$.

Theorem 7. *Let $f \in C^2[a, b]$, and let $h = \max_{j=1,\ldots,N}(x_j - x_{j-1})$ be the grid diameter. Then*

$$\|f - s_L\|_{L^\infty[a,b]} \leq \frac{h^2}{8} \|f''\|_{L^\infty[a,b]}.$$

Proof. Let $x \in [x_{j-1}, x_j]$ for some $j = 1, \ldots, N$. We can apply the basic result of accuracy of polynomial interpolation with $n = 1$: there exists $\xi \in [x_{j-1}, x_j]$ such that

$$f(x) - s_L(x) = \frac{1}{2} f''(\xi)\, (x - x_{j-1})(x - x_j), \qquad x \in [x_{j-1}, x_j].$$

Let $h_j = x_j - x_{j-1}$. It is easy to check that the product $(x - x_{j-1})(x - x_j)$ takes its maximum value at the midpoint $\frac{x_{j-1}+x_j}{2}$, and that the value is $h_j^2/4$. We then have

$$|f(x) - s_L(x)| \leq \frac{h_j^2}{8} \max_{\xi \in [x_{j-1}, x_j]} |f''(\xi)|, \qquad x \in [x_{j-1}, x_j].$$

The conclusion follows by taking a maximum over j. $\square$

We can express any piecewise linear interpolant as a superposition of "tent" basis functions $\phi_k(x)$:

$$s_L(x) = \sum_k c_k \phi_k(x),$$

where $\phi_k(x)$ is the piecewise linear function equal to 1 at $x = x_k$, and equal to zero at all the other grid points x_j, $j \neq k$. Or in short, $\phi_k(x_j) = \delta_{jk}$. On an equispaced grid $x_j = jh$, an explicit formula is

$$\phi_k(x) = \frac{1}{h} S_{(1)}(x - x_k),$$

where

$$S_{(1)}(x) = x_+ - 2(x-h)_+ + (x-2h)_+,$$

and where x_+ denotes the positive part of x (i.e., x if $x \geq 0$, and zero if $x < 0$.)

Observe that we can simply take $c_k = f(x_k)$ above. The situation will be more complicated when we pass to higher-order polynomials.

3.4.2 Cubic splines

Let us now consider the case $n = 3$ of an interpolant which is a third-order polynomial, i.e., a cubic, on each subinterval. The most we can ask is that the value of the interpolant, its derivative, and its second derivative be continuous at the junction points x_j.

Definition 5. *(Interpolating cubic spline) Let $f \in C[a,b]$, and $\{x_j, j = 0, \ldots, N\} \subset [a,b]$. An interpolating cubic spline is a function $s(x)$ such that*

1. *$s(x_j) = f(x_j)$;*

2. *$s(x)$ is a polynomial of degree 3 over each segment $[x_{j-1}, x_j]$;*

3. *$s(x)$ is globally C^2, i.e., at each junction point x_j, we have the relations*

$$s(x_j^-) = s(x_j^+), \qquad s'(x_j^-) = s'(x_j^+), \qquad s''(x_j^-) = s''(x_j^+),$$

   where the notations $s(x_j^-)$ and $s(x_j^+)$ refer to the adequate limits on the left and on the right.

Let us count the degrees of freedom. A cubic polynomial has 4 coefficients. There are $N+1$ points, hence N subintervals, for a total of $4N$ numbers to be specified.

The interpolating conditions $s(x_j) = f(x_j)$ specify two degrees of freedom per polynomial: one value at the left endpoint x_{j-1}, and one value at the right endpoint x_j. That's $2N$ conditions. Continuity of $s(x)$ follows automatically. The continuity conditions on s', s'' are imposed only at the interior grid points x_j for $j = 1, \ldots, N-1$, so this gives rise to $2(N-1)$ additional conditions, for a total of $4N-2$ equations.

There is a mismatch between the number of unknowns ($4N$) and the number of conditions ($4N-2$). Two more degrees of freedom are required to completely specify a cubic spline interpolant, hence the precaution to write "an" interpolant in the definition above, and not "the" interpolant.

The most widespread choices for fixing these two degrees of freedom are:

- Natural splines:
  $$s''(x_0) = s''(x_N) = 0.$$
  If $s(x)$ measures the displacement of a beam, a condition of vanishing second derivative corresponds to a free end.

- Clamped spline:
  $$s'(x_0) = p_0, \qquad s'(x_N) = p_N,$$
  where p_0 and p_N are specified values that depend on the particular application. If $s(x)$ measures the displacement of a beam, a condition of vanishing derivative corresponds to a horizontal clamped end.

- Periodic spline: assuming that $s(x_0) = s(x_N)$, then we also impose
  $$s'(x_0) = s'(x_N), \qquad s''(x_0) = s''(x_N).$$

Let us now explain the algorithm most often used for determining a cubic spline interpolant, i.e., the $4N$ coefficients of the N cubic polynomials, from the knowledge of $f(x_j)$. Let us consider the natural spline.

It is advantageous to write a system of equations for the second derivatives at the grid points, that we denote $\sigma_j = s''(x_j)$. Because $s(x)$ is piecewise cubic, we know that $s''(x)$ is piecewise linear. Let $h_j = x_j - x_{j-1}$. We can write
$$s''(x) = \frac{x_j - x}{h_j}\sigma_{j-1} + \frac{x - x_{j-1}}{h_j}\sigma_j, \qquad x \in [x_{j-1}, x_j].$$

Hence

$$s(x) = \frac{(x_j - x)^3}{6h_j}\sigma_{j-1} + \frac{(x - x_{j-1})^3}{6h_j}\sigma_j + \alpha_j(x - x_{j-1}) + \beta_j(x_j - x).$$

We could have written $ax + b$ for the effect of the integration constants in the equation above, but writing it in terms of α_j and β_j makes the algebra that follows simpler. The two interpolation conditions for $[x_{j-1}, x_j]$ can be written

$$s(x_{j-1}) = f(x_{j-1}) \qquad \Rightarrow \qquad f(x_{j-1}) = \frac{\sigma_{j-1}h_j^2}{6} + h_j\beta_j,$$

$$s(x_j) = f(x_j) \qquad \Rightarrow \qquad f(x_j) = \frac{\sigma_j h_j^2}{6} + h_j\alpha_j.$$

One then isolates α_j, β_j in this equation; substitutes those values in the equation for $s(x)$; and evaluates the relation $s'(x_j^-) = s'(x_j^+)$. Given that $\sigma_0 = \sigma_N = 0$, we end up with a system of $N - 1$ equations in the $N - 1$ unknowns $\sigma_1, \ldots, \sigma_N$. Skipping the algebra, the end result is

$$h_j\sigma_{j-1} + 2(h_{j+1}{+}h_j)\sigma_j + h_{j+1}\sigma_{j+1} = 6\left(\frac{f(x_{j+1}) - f(x_j)}{h_{j+1}} - \frac{f(x_j) - f(x_{j-1})}{h_j}\right).$$

We are in presence of a tridiagonal system for σ_j. It can be solved efficiently with Gaussian elimination, yielding a LU decomposition with bidiagonal factors. Unlike in the case of linear splines, there is no way around the fact that a linear system needs to be solved.

Notice that the tridiagonal matrix of the system above is diagonally dominant (each diagonal element is strictly greater than the sum of the other elements on the same row, in absolute value), hence it is always invertible.

One can check that the interpolation for cubic splines is $O(h^4)$ well away from the endpoints. This requires an analysis that is too involved for the present set of notes.

Finally, like in the linear case, let us consider the question of expanding a cubic spline interpolant as a superposition of basis functions

$$s(x) = \sum_k c_k\phi_k(x).$$

There are many ways of choosing the functions $\phi_k(x)$, so let us specify that they should have as small a support as possible, that they should have the same smoothness C^2 as $s(x)$ itself, and that they should be translates of each other when the grid x_j is equispaced with spacing h.

The only solution to this problem is the cubic B-spline. Around any interior grid point x_k, is supported in the interval $[x_{k-2}, x_{k+2}]$, and is given by the formula

$$\phi_k(x) = \frac{1}{4h^3} S_{(3)}(x - x_{k-2}),$$

where

$$S_{(3)}(x) = x_+^3 - 4(x-h)_+^3 + 6(x-2h)_+^3 - 4(x-3h)_+^3 + (x-4h)_+^3,$$

and where x_+ is the positive part of x. One can check that $\phi_k(x)$ takes the value 1 at x_k, $1/4$ at $x_{k\pm1}$, zero outside of $[x_{k-2}, x_{k+2}]$, and is C^2 at each junction. It is a bell-shaped curve. it is an interesting exercise to check that it can be obtained as the convolution of two "tent" basis functions of linear interpolation:

$$S_{(3)}(x) = cS_{(1)}(x) * S_{(1)}(x),$$

where c is some constant, and the symbol $*$ denotes convolution:

$$f * g(x) = \int_{\mathbb{R}} f(y) g(x-y)\, dy.$$

Now with cubic B-splines, we cannot put $c_k = f(x_k)$ anymore, since $\phi_k(x_j) \neq \delta_{jk}$. The particular values of c_k are the result of solving a linear system, as mentioned above. Again, there is no way around solving a linear system. In particular, if $f(x_k)$ changes at one point, or if we add a datum point (x_k, f_k), then the update requires re-computing the whole interpolant. Changing one point has ripple effects throughout the whole interval $[a, b]$. This behavior is ultimately due to the more significant overlap between neighboring ϕ_k in the cubic case than in the linear case.

Chapter 4

Nonlinear equations

4.1 Root finding

Consider the problem of solving any nonlinear relation $g(x) = h(x)$ in the real variable x. We rephrase this problem as one of finding the zero (root) of a function, here $f(x) = g(x) - h(x)$. The minimal assumption we need on f, g, h is that they're continuous.

We have at our disposal any number of evaluations of f and its derivative f'.

1. **Method 1: bisection**. The bisection methods starts from two points a_0 and b_0 such that

$$f(a_0) > 0, \qquad \text{and} \qquad f(b_0) < 0.$$

Because f is continous, there must exist a root in the interval $[a_0, b_0]$. At stage k, assume that we have obtained an interval $[a_k, b_k]$ such that the same sign properties hold: $f(a_k) > 0$ and $f(b_k) < 0$. The bisection method consists in subdividing the interval $[a_k, b_k]$ in two and discard the half in which there may not be a root. Let $m_k = (a_k + b_k)/2$.

- If $f(m_k) < 0$, then it is the interval $[a_k, m_k]$ which is of interest. We put $a_{k+1} = a_k$ and $b_{k+1} = m_k$.
- If $f(m_k) > 0$, then it is the interval $[m_k, b_k]$ which is of interest. We put $a_{k+1} = m_k$ and $b_{k+1} = b_k$.
- If $f(m_k) = 0$, then m_k is a root and we stop the algorithm.

In practice, this iteration is stopped once $f(m_k)$ gets small enough. Let x^* be the unknown root. The error obeys

$$|x^* - m_k| \le |b_k - a_k| = 2^{-k}|b_0 - a_0|.$$

Every step of the bisection discovers a new correct digit in the binary expansion of x^*.

The advantage of the bisection method is that it is guaranteed to converge to a root, by construction. On the other hand, convergence is rather show compared to the next 2 methods we now present. If there are several roots, the bisection method will converge toward one of them (we may not have no control over which root the method chooses.)

2. **Method 2: Newton-Raphson**. This method is very important: it is the basis of most optimization solvers in science and engineering. Let us first present the Newton-Raphson method for solving a single scalar equation $f(x) = 0$.

   Newton's method fits a tangent line to the point $(x_n, f(x_n))$ on the graph of f, and defines x_{n+1} at the intersection of this tangent line with the x axis. We have

$$0 = f(x_n) + (x_{n+1} - x_n)f'(x_n),$$

   from which we isolate

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

   For instance, we can find the decimal expansion of $\sqrt{2}$ by finding the positive root of $f(x) = x^2 - 2$. The iteration reads

$$x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n} = \frac{x_n}{2} + \frac{1}{x_n}.$$

   Starting with $x_0 = 1$, we get $x_1 = \frac{3}{2} = 1.5$, $x_2 = \frac{17}{12} = 1.4167...$, $x_3 = \frac{577}{408} = 1.4142157...$ The true value of $\sqrt{2}$ is 1.4142135...

   Convergence is very fast, when it occurs. Assume that f'' is continuous, and that $f'(x) \neq 0$ in some neighborhood of the root x^* (large enough so that all our iterates stay in this neighborhood.) Put $\epsilon_n = x_n - x^*$.

Then we can perform a Taylor expansion of f around x_n, and evaluate it at $x = x^*$:

$$0 = f(x^*) = f(x_n) + (x^* - x_n)f'(x_n) + \frac{1}{2}(x^* - x_n)^2 f''(\xi),$$

for some $\xi \in \text{int}(x_n, x^*)$ (the notation int refers to the interval generated by x_n and x^*, i.e., either $[x_n, x^*]$ or $[x^*, x_n]$.) We also have the equation defining x_{n+1}:

$$0 = f(x_n) + (x_{n+1} - x_n)f'(x_n).$$

Subtracting those 2 equations, we get

$$0 = -\epsilon_{n+1} f'(x_n) + \frac{1}{2}\epsilon_n^2 f''(\xi),$$

$$\epsilon_{n+1} = \frac{1}{2}\frac{f''(\xi)}{f'(x_n)}\epsilon_n^2.$$

Our assumptions ensure that the ratio $\frac{f''(\xi)}{f'(x_n)}$ exists and converges to some limit $(f''(x^*)/f'(x^*))$ as $n \to \infty$. Hence the sequence is bounded uniformly in n, and we can write

$$|\epsilon_{n+1}| \le C\epsilon_n^2,$$

where $C > 0$ is some number (which depends on f but not on n.) It follows that

$$|\epsilon_n| \le \frac{1}{C}(C\epsilon_0)^{2^k}.$$

We say the method "converges quadratically" because the exponent of ϵ_n is 2. The number of correct digits is *squared* at each iteration. In contrast, the bisection method only converges linearly. We also sometimes refer to "linear convergence" as first-order convergence, although the meaning of the expression is completely different from what is was in the previous chapters.

Convergence is ensured as soon as the starting point x_0 is close enough to the (unknown) root x^*, in the sense that $|C\epsilon_0| < 1$, so that $(C\epsilon_0)^{2^k} \to 0$ as $k \to \infty$. If the condition $|C\epsilon_0| < 1$ is not satisfied, Newton's method may very well diverge. For instance, we expect problems when the derivative is very small: following the tangent can take us to a region very far away from the root. An example of a function $f(x)$ for which Newton's method diverges is $\operatorname{atan}(x)$, when x_0 is chosen to be too far from the origin.

On the plus side, Newton's method is fast. On the minus side, Newton's method only converges to a root only when you're already quite close to it.

3. **Method 3: the secant method**.

If we do not know the derivative, we cannot set up Newton's method, but we can approximate it by replacing the derivative by (let $f_n = f(x_n)$)

$$f[x_{n-1}, x_n] = \frac{f_n - f_{n-1}}{x_n - x_{n-1}}.$$

Hence we define x_{n+1} by

$$x_{n+1} = x_n - \frac{f_n}{f[x_{n-1}, x_n]}.$$

The geometrical idea is to replace the tangent line at x_n by the secant line supported by x_{n-1} and x_n. The secant method requires two points x_0 and x_1 as starting guesses.

Notice that at each step, only one evaluation of f is necessary, because $f(x_{n-1})$ is already known from the previous iteration. If we were to form a finite difference approximation of the derivative with a very small grid step h, we may be more accurate but that requires two evaluations of f rather than one.

Let us check the convergence properties of the secant method. The line joining the two points (x_{n-1}, f_{n-1}) and (x_n, f_n) is the degree-1 interpolant in the interval $[x_{n-1}, x_n]$:

$$p(x) = f_n + f[x_{n-1}, x_n](x - x_n).$$

Outside of this interval, it is an extrapolant. Regardless of whether $x \in [x_{n-1}, x_n]$ or not, the difference between p and f is known from a theorem we saw in the previous chapter:

$$f(x) - p(x) = \frac{1}{2} f''(\xi)(x - x_n)(x - x_{n-1}),$$

where ξ is in the smallest interval containing x, x_{n-1}, and x_n. Evaluating this relation at the root $x = x^*$, we get

$$0 = f_n + f[x_{n-1}, x_n](x^* - x_n) + \frac{1}{2} f''(\xi)(x^* - x_n)(x^* - x_{n-1}).$$

On the other hand the definition of x_{n+1} gives

$$0 = f_n + f[x_{n-1}, x_n](x_{n+1} - x_n).$$

Subtracting these two equations we get

$$\epsilon_{n+1} = \frac{1}{2} \frac{f''(\xi)}{f[x_{n-1}, x_n]} \epsilon_n \epsilon_{n-1}.$$

Again, thanks to the same assumptions on f as in Newton's method, the ratio $\frac{f''(\xi)}{f[x_{n-1},x_n]}$ has a finite limit as $n \to \infty$, hence is bounded by some number $C > 0$. We get

$$|\epsilon_{n+1}| \leq C|\epsilon_n||\epsilon_{n-1}|.$$

The decay of ϵ_n is somewhere between first (linear) and second (quadratic) order. To obtain a more precise rate of decay, we guess that the inequality above should be reducible to the form $|\epsilon_n| \leq C|\epsilon_{n-1}|^p$ for some p. Using this equation and $|\epsilon_{n+1}| \leq C|\epsilon_n|^p$ above, we get

$$|\epsilon_{n-1}|^{p^2} \leq C|\epsilon_{n-1}|^p|\epsilon_{n-1}|.$$

The exponents match on the left and the right provided $p^2 = p + 1$, which has for positive solution

$$p = \frac{1 + \sqrt{5}}{2}. \qquad \text{(a number sometimes called the golden ratio).}$$

We check that $p = 1.618...$, a number between 1 and 2, Hence the secant method is faster than bisection, but slower than Newton's method. The secant method inherits the problem of Newton's method: it only converges when the starting guesses x_0 and x_1 are sufficiently close to the root.

We can also set up Newton's method in several dimensions. A system of nonlinear equations is of the form

$$f_i(x_1, \ldots, x_n) = 0, \qquad i = 1, \ldots, n.$$

We take the same number of equations and unknowns, so that we may be in a situation where there is one solution (rather than a continuum of solutions or no solution at all.) Whether the system has zero, one or several solutions is still a question that needs to be addressed separately. The shorthand notation for the system is $\mathbf{f}(\mathbf{x}) = 0$

By analogy with the 1D case we perform a Taylor expansion about $\mathbf{x}_n$:

$$0 = \mathbf{f}(\mathbf{x}^*) = \mathbf{f}(\mathbf{x}_n) + \nabla \mathbf{f}(\mathbf{x}_n)(\mathbf{x}^* - \mathbf{x_n}) + O(\|\mathbf{x}^* - \mathbf{x}_n\|^2).$$

With indices this equation is written as

$$0 = f_i(\mathbf{x}^*) = f_i(\mathbf{x}_n) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j}(\mathbf{x}_n)(x_{j,n} - x_j^*) + O(\sum_j (x_{j,n} - x_j^*)^2).$$

(Watch the subscript n which indicates the n-th iterate while the subscript j indicates the j-th component.) The next iterate $\mathbf{x}_{n+1}$ is defined by neglecting the quadratic error and isolating $\mathbf{x}^*$. A linear system of equations needs to be solved: the Jacobian matrix $\nabla \mathbf{f}(\mathbf{x}_n)$ is inverted and we get

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \left[\nabla \mathbf{f}(\mathbf{x}_n)\right]^{-1} \mathbf{f}(\mathbf{x}_n).$$

The geometrical interpretation of this equation is that we can fit the tangent plane to each of the surfaces $y = f_i(x_1, \ldots, x_n)$ in $\mathbb{R}^{n+1}$, find the line at the intersection of all these planes, and check where this line intersects the (hyper)plane $y = 0$.

Newton's method is still quadratically convergent in multiple dimensions, and special care must still be taken to make sure that we start close enough to a root.

Example 10.

$$x_1^2 + x_2^2 = 1, \qquad x_2 = \sin(x_1).$$

Write this as a root-finding problem: $f_1 = f_2 = 0$ *with*

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 1, \qquad f_2(x_1, x_2) = x_2 - sin(x_1).$$

The Jacobian matrix is

$$J = \nabla \mathbf{f}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 & 2x_2 \\ -\cos(x_1) & 1 \end{pmatrix}.$$

Use the formula for the inverse of a 2-by-2 matrix:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix},$$

to obtain

$$J^{-1} = \frac{1}{2x_1 + 2x_2\cos(x_1)} \begin{pmatrix} 1 & -2x_2 \\ \cos(x_1) & 2x_1 \end{pmatrix}.$$

The Newton iteration is therefore

$$\begin{pmatrix} x_{1,n+1} \\ x_{2,n+1} \end{pmatrix} = \begin{pmatrix} x_{1,n} \\ x_{2,n} \end{pmatrix} - J^{-1} \begin{pmatrix} x_{1,n}^2 + x_{2,n}^2 - 1 \\ x_{2,n} - \sin x_{1,n} \end{pmatrix}.$$

4.2 Optimization problems

Another important recurring problem in science and engineering is that of finding a minimum or a maximum of a function $F(x)$. A point x^* is a local minimum when $F(y) \geq F(x^*)$ for all y in a neighborhood of x^*. It is a global minimum when $F(y) \geq F(x^*)$ for all y. We write

$$\min_x F(x)$$

for the minimum value $F(x^*)$. We then call x^* the argument of the minimum. Maximizing $F(x)$ instead is the same as minimizing $-F(x)$, so it suffices to talk about minimization.

When $F(x)$ is smooth, and x is allowed to run over all real numbers (not restricted to an interval or other set), then it suffices to solve $F'(x) = 0$ (and check that $F''(x) > 0$) in order to find a local minimum. Hence it suffices to apply Newton's method or any other root-finding method to the function $f(x) = F'(x)$. We obtain

$$x_{n+1} = x_n - \frac{F'(x_n)}{F''(x_n)}.$$

In multiple dimensions, we minimize a scalar function $F(x_1, \ldots, x_n)$. The optimality condition, obeyed at the minimum $x_1^*, \ldots x_n^*$, is that all the partial derivatives of F vanish, i.e.,

$$\nabla F(x_1^*, \ldots x_n^*) = 0.$$

Newton's method, also called Newton descent, follows from considering these equations as a nonlinear system $f_i(x_1, \ldots, x_n) = 0$ with $f_i = \frac{\partial F}{\partial x_i}$. We get

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [\nabla\nabla F(\mathbf{x}_n)]^{-1} \nabla F(\mathbf{x}_n).$$

The matrix $\nabla\nabla F$ of second partial derivatives of F is called the Hessian. In index notation,

$$(\nabla\nabla F)_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j}.$$

Compare Newton's method with simple gradient descent:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha \nabla F(\mathbf{x}_n),$$

for some sufficiently small scalar α. Gradient descent is slower but typically converges from a larger set of initial guesses than Newton's method.

Example 11. *Consider*

$$F(x_1, x_2) = x_1^2 + (\log x_2)^2.$$

This function has a unique minimum for $x_1 \in \mathbb{R}$ and $x_2 > 0$. We compute

$$\nabla F(x_1, x_2) = \begin{pmatrix} 2x_1 \\ \frac{2\log x_2}{x_2} \end{pmatrix}$$

and

$$\nabla\nabla F(x_1, x_2) = \begin{pmatrix} 2 & 0 \\ 0 & \frac{2-2\log x_2}{x_2^2} \end{pmatrix}.$$

Newton's iteration is therefore

$$\begin{pmatrix} x_{1,n+1} \\ x_{2,n+1} \end{pmatrix} = \begin{pmatrix} x_{1,n} \\ x_{2,n} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{x_{2,n}^2}{2-2\log x_{2,n}} \end{pmatrix} \begin{pmatrix} 2x_{1,n} \\ 2\frac{\log x_{2,n}}{x_{2,n}} \end{pmatrix}.$$

Notice that x_1 goes in one step to zero, because a quadratic function is exactly minimized by Newton's method.

Chapter 5

Methods for ordinary differential equations

5.1 Initial-value problems

Initial-value problems (IVP) are those for which the solution is entirely known at some time, say $t = 0$, and the question is to solve the ODE

$$y'(t) = f(t, y(t)), \qquad y(0) = y_0,$$

for other times, say $t > 0$. We will consider a scalar y, but considering systems of ODE is a straightforward extension for what we do in this chapter. We'll treat both theoretical questions of existence and uniqueness, as well as practical questions concerning numerical solvers.

We speak of t as being time, because that's usually the physical context in which IVP arise, but it could very well be a space variable.

Does a solution exist, is it unique, and does it tend to infinity (blow up) in finite time? These questions are not merely pedantic. As we now show with two examples, things can go wrong very quickly if we posit the wrong ODE.

Example 12. *Consider*

$$y' = \sqrt{y}, \qquad y(0) = 0.$$

By separation of variables, we find

$$\int \frac{dy}{\sqrt{y}} = \int dt \qquad \Rightarrow \qquad y(t) = \frac{(t+C)^2}{4}.$$

Imposing the initial condition yields $C = 0$, hence $y(t) = t^2/4$. However, $y(t) = 0$ is clearly another solution, so we have non-uniqueness. In fact, there is an infinite number of solutions, corresponding to $y(t) = 0$ for $0 \leq t \leq t^*$ for some t^*, which then takes off along the parabola the parabola $y(t) = (t - t^*)^2/4$ for times $t \geq t^*$.

Example 13. *Consider*

$$y' = y^2, \qquad y(0) = 1.$$

By separation of variables, we find

$$\int \frac{dy}{y^2} = \int dt \qquad \Rightarrow \qquad y(t) = \frac{-1}{t + C}.$$

The initial condition gives $C = -1$, hence $y(t) = 1/(1 - t)$. It blows up at time $t = 1$, because $y(t)$ has a vertical asymptote. We say that the solution exists locally in any interval to the left of $t = 1$, but we don't have global existence.

Blowup and non-uniqueness are generally, although not always[1], unrealistic in applications. A theory of existence and uniqueness, including global existence (non blowup), is desired to guide us in formulating valid models for physical phenomena.

The basic result is the following.

Theorem 8. *(Picard) For given T, C, and y_0, consider the box B in (t, y) space, given by*

$$B = [0, T] \times [y_0 - C, y_0 + C].$$

Assume that

- *$f(t, y)$ is continuous over B;*
- *$|f(t, y)| \leq K$ when $(t, y) \in B$; (boundedness)*
- *$|f(t, u) - f(t, v)| \leq L|u - v|$ when $(t, u), (t, v) \in B$. (Lipschitz continuity).*

[1]In nonlinear optics for instance, laser pulses may "collapse". This situation is somewhat realistically modeled by an ODE that blows up, although not for times arbitrarily close to the singularity.

Assume furthermore that $C \geq \frac{K}{L}(e^{LT} - 1)$*. Then there exists a unique* $y \in C^1[0, T]$*, such that*

$$y'(t) = f(t, y(t)), \qquad y(0) = y_0,$$

and such that $|y(t) - y_0| \leq C$*. In short, the solution exists, is unique, and stays in the box for times* $0 \leq t \leq T$*.*

Proof. The technique is called Picard's iteration. See p.311 in Suli-Mayers. □

The ODE $y' = \sqrt{y}$ does not satisfy the assumptions of the theorem above because the square root is not a Lipschitz function. The ODE $y' = y^2$ does not satisfy the assumptions of the theorem above because y^2 is not bounded by a constant K.

5.2 Numerical methods for Initial-Value Problems

Here is an overview of some of the most popular numerical methods for solving ODEs. Let $t_n = nh$, and denote by y_n the approximation of $y(t_n)$.

1. Forward Euler (a.k.a. explicit Euler).

   $$y_{n+1} = y_n + hf(t_n, y_n).$$

   This formula comes from approximating the derivative y' at $t = t_n$ by a forward difference. It allows to march in time from the knowledge of y_n, to get y_{n+1}.

2. Backward Euler (a.k.a. implicit Euler).

   $$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}).$$

   This time we use a backward difference for approximating the derivative at $t = t_{n+1}$. The unknown y_{n+1} appears implicitly in this equation, hence the name implicit. It still needs to be solved for as a function of y_n, using (for instance) Newton's method. The strategy is still to march in time, but at every step there is a nonlinear equation to solve.

3. Trapezoidal (a.k.a. midpoint) rule (implicit).

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right].$$

In this equation $\frac{y_{n+1}-y_n}{h}$ has the interpretation of a centered difference about the midpoint $t_{n+\frac{1}{2}} = \frac{t_n+t_{n+1}}{2}$, but since $f(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})$ is not accessible ($y_{n+\frac{1}{2}}$ is not part of what we wish to solve for), we replace it by the average $\frac{1}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right]$. This gives a more balanced estimate of the slope $\frac{y_{n+1}-y_n}{h}$. It is an implicit method: y_{n+1} needs to be solved for.

4. Improved Euler, Runge-Kutta 2 (explicit).

$$\tilde{y}_{n+1} = y_n + hf(t_n, y_n),$$

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right].$$

This is the simplest of "predictor-corrector" methods. It is like the midpoint rule, except that we use a guess $\tilde{y}_{n+1}$ for the unknown value of y_{n+1} in the right-hand side, and this guess comes from the explicit Euler method. Now y_{n+1} only appears in the left-hand side, so this is an explicit method.

5. Runge-Kutta 4 (explicit).

$$y_{n+1} = y_n + h[k_1 + 2k_2 + 2k_3 + k_4],$$

where the slopes $k_1, \ldots, k_4$ are given in succession by

$$k_1 = f(t_n, y_n), \qquad k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1),$$

$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2), \qquad k_4 = f(t_n + h, y_n + hk_3).$$

6. There are also methods that involve not just the past value y_n, but a larger chunk of history y_{n-1}, y_{n-2},etc. These methods are called multistep. They are in general less flexible than the one-step methods described so far, in that they require a constant step h as we march in time.

Two features of a numerical method are important when choosing a numerical method:

- Is it *convergent*, i.e., does the computed solution at some fixed time $t = T$ tend to the true solution as $h \to 0$, and at which rate?
- Is it *stable*, i.e., does the solution computed with a fixed time step h become unstable ("blows up") as t increases?

5.2.1 Convergence

To understand convergence better in the case of one-step methods, let us write the numerical scheme as

$$y_{n+1} = \Psi(t_n, y_n, h),$$

and introduce the *local error*

$$e_{n+1}(h) = \Psi(t_n, y(t_n), h) - y(t_{n+1}),$$

as well as the *global error*

$$E_n(h) = y_n - y(t_n).$$

The expression of the local error can be explained as "trying the exact solution in the numerical scheme" — although the exact solution is unknown. It is a lot easier to approach the convergence question via local errors than global errors. It does not immediately make sense, for instance, to "try an approximate solution in the exact ODE".

Convergence is the study of the global error. *Consistency* is the study of the local error. A numerical method is called consistent if the local error decays sufficiently fast as $h \to 0$ that there is hope that the global error would be small as well. The particular rate at which the local error decays is related to the notion of order of an ODE solver.

Definition 6. *(Consistency) Ψ is consistent if, for any $n \geq 0$,*

$$\lim_{h \to 0} \frac{e_n(h)}{h} = 0$$

Definition 7. *(Order) Ψ is of order p if $e_n(h) = O(h^{p+1})$.*

The basic convergence theorem for one-step solvers, that we will not prove, is that if the local error is $O(h^{p+1})$, then the global error is $O(h^p)$. This convergence result is only true as stated for one-step methods; for multi-step methods we would also need an assumption of stability (discussed below). Intuitively, the local errors compound over the $O(1/h)$ time steps necessary to reach a given fixed time t, hence the loss of one power of h. Of course the local errors don't exactly add up; but they do up to a multiplicative constant. It is the behavior of the global error that dictates the notion of order of the numerical scheme.

It is a good exercise to show, using elementary Taylor expansions, that the explicit and implicit Euler methods are of order 1, and that the midpoint rule and improved Euler methods are of order 2. It turns out that Runge-Kutta 4 is of order 4, but it is not much fun to prove that.

5.2.2 Stability

Consistency and convergence do not tell the whole story. They are helpful in the limit $h \to 0$, but don't say much about the behavior of a solver in the interesting regime when h is small, but not so small as to make the method computationally inefficient.

The single most important consideration in the regime of moderately small h is stability. The rigorous criterion for stability is that if we run the numerical scheme with different initial conditions y_0 and $\tilde{y}_0$, the discrepancy in computed solutions should be under control, in the sense that $|y_n - \tilde{y}_n| \leq C|y_0 - \tilde{y}_0|$, with $n = O(1/h)$, and C independent of h. However, this criterion is too complex to handle as such. The important ideas already appear if we study the representative setting of linear stability. The linearization of the ODE $y' = f(t, y)$ about a point y_0 is obtained from writing the Taylor expansion

$$\frac{d}{dt}(y(t) - y_0) = f(t, y(t)) = f(t, y_0) + \frac{\partial f}{\partial y}(t, y_0)(y(t) - y_0) + o(|y(t) - y_0|).$$

The culprit for explosive (exponential) growth or decay is the linear term $\frac{\partial f}{\partial y}(t, y_0)(y(t) - y_0)$. Indeed, if the other two terms are neglected, we can write the solution, locally, as an exponential. For practical purposes, it is therefore sufficient to check stability for the linear equation $y' = \lambda y$, keeping in mind that λ is a number representative of the derivative $\frac{\partial f}{\partial y}(t, y_0)$ of f at y_0 in the y variable.

Definition 8. *(Linear stability[2]) Suppose $y' = \lambda y$ for some $\lambda \in \mathbb{C}$. Then the numerical method Ψ is linearly stable if $y_n \to 0$ as $n \to \infty$.*

Of course linear stability depends on the value of λ. Stability for the original equation $y' = \lambda y$ is guaranteed if $\mathrm{Re}(\lambda) < 0$ (because the solution is $y(0)e^{\lambda t}$), and the question is that of showing whether a numerical method Ψ is stable under the same condition or not.

If a numerical method is stable in the above sense for a certain range of values of λ, then it is possible to show that it will be stable for the ODE $y' = f(t, y)$ as long as $\frac{\partial f}{\partial y}$ is in that range of λ (and f is smooth enough). We won't prove this theorem here.

Let us consider a few examples.

Example 14. *For the forward Euler method applied to $y' = \lambda y$, we get*

$$y_{n+1} = y_n + h\lambda y_n = (1 + h\lambda)y_n.$$

The iterates y_n tend to zero provided $|1 + h\lambda| < 1$, where the $|\cdot|$ denote the complex modulus. Write $h\lambda = x + iy$, so that the inequality becomes $(1 + x)^2 + y^2 < 1$. This is the equation of a disc in the complex $(h\lambda)$-plane, with center at -1, and radius 1. If $h\lambda$ sits inside this disk, the method is stable, and otherwise it isn't. We say that the forward Euler method is conditionally stable: typically, we require both $Re(\lambda) < 0$ and a small step size h in order to guarantee stability.

Example 15. *The backward Euler method applied to $y' = \lambda y$ gives*

$$y_{n+1} = y_n + h\lambda y_{n+1},$$

or in other words

$$y_{n+1} = \frac{y_n}{1 - h\lambda}.$$

The iterates y_n tend to zero provided $|1 - h\lambda| > 1$. In terms of $h\lambda = x + iy$, this becomes $(x - 1)^2 + y^2 > 1$. This condition is satisfied whenever $h\lambda$ is outside the disc in the complex $(h\lambda)$-plane with center at $+1$, and radius 1. In that case the method is stable, and otherwise it isn't. We say that the backward Euler method is unconditionally stable: the stability zone for the ODE ($Re(\lambda) < 0$) is always included in the stability zone of the numerical

[2]Sometimes called A-stability in some texts.

method, regardless of h. In fact the zone of stability for the backward Euler method is larger than the left complex half-plane, so there exist choices of (large) time steps for which the numerical method is stable although the ODE isn't.

Example 16. *The linearized stability analysis of the midpoint method gives*

$$y_{n+1} = y_n + h\lambda(\frac{y_n}{2} + \frac{y_{n+1}}{2}),$$

hence

$$y_{n+1} = \left(\frac{1 + h\lambda/2}{1 - h\lambda/2}\right) y_n.$$

The method is stable provided

$$|\frac{1 + h\lambda/2}{1 - h\lambda/2}| < 1.$$

In terms of $h\lambda = x + iy$, we can simplify to get

$$(1 + \frac{x}{2})^2 + (\frac{y}{2})^2 < (1 - \frac{x}{2})^2 + (\frac{y}{2})^2,$$

which is true if and only if $x < 0$. So the stability region is Re$(h\lambda) < 0$, the same as that of the ODE. As a result, the method is unconditionally stable.

It is a general rule that explicit methods have conditional stability (stability only happens when the time step is small enough, if it does at all), whereas implicit methods are unconditionally stable.

The stability regions for the Runge-Kutta methods are plotted on page 351 of Suli-Mayers; or google for it.

Stability can also be studied for systems of ODEs $y'(t) = f(t, y(t))$ where both y and f are vectors. The interesting object is now the Jacobian matrix

$$A = \nabla_y f(t, y_0).$$

(it is a matrix because the gradient of a vector function is a "vector of vectors", i.e., a matrix.) The linearized problem is now

$$y'(t) = Ay(t).$$

It turns out[3] that stability will happen when the eigenvalues λ of A *all* obey $\mathrm{Re}(\lambda) < 0$. So the story is the same as in the scalar case, except that λ is now *any* eigenvalue of the Jacobian matrix. So we need to make sure that $h\lambda$ is in the stability zone of the ODE solver in the complex plane, for every eigenvalue λ of the Jacobian matrix.

Problems for which λ has a very large, negative real part are called *stiff*. Physically they are very stable, but they pose numerical problems for explicit methods since the region of stability does not extend very far along the negative real axis. Implicit methods are hands down the best for stiff problems.

It should also be mentioned that stability can be studied for multi-step methods of the form $y_{n+1} = ay_n + by_{n-1}$, say, in the linearized case. Such recurrence equations can be solved by letting $y_n = \rho^n$, and determining the number(s) ρ for which y_n is a solution. In the 3-term case, there are two such numbers (roots of a quadratic). All the ρ_j thus obtained should satisfy $|\rho_j| < 1$.

5.2.3 Miscellaneous

Another interesting class of methods, and a way to naturally design high-order methods, is *deferred correction*. Assume that time is split into a uniform grid $t_j = jh$, and that some low-order method has given us the samples $y_1, \ldots, y_{n+1}$ for some (small) m. Denote by $\pi_n(t)$ the n-th order interpolation polynomial passing through the (t_j, y_j). Consider the error ("defect")

$$\delta(t) = y(t) - \pi_n(t).$$

It obeys the equation

$$\delta'(t) = f(t, y(t)) - \pi_n'(t), \qquad \delta(0) = 0.$$

We do not have access to $y(t)$ in the argument of f, but we can replace it by our best guess $\pi_n(t) + \delta(t)$. This way we can compute an approximate defect

$$\tilde{\delta}'(t) = f(t, \pi_n(t) + \tilde{\delta}(t)) - \pi_n'(t), \qquad \tilde{\delta}(0) = 0.$$

[3]When A is diagonalizable as $V\Lambda V^{-1}$, the system becomes $y'(t) = V\Lambda V^{-1}y(t)$, so $(V^{-1}y)'(t) = \Lambda V^{-1}y(t)$. Let $z(t) = V^{-1}y(t)$ be the coefficients of the vector $y(t)$ in the eigen-basis of A. Then the equations are decoupled as $z_j'(t) = \lambda_j z_j(t)$, resulting in $z_j(t) = z_j(0)e^{\lambda_j t}$ and $y(t) = V\mathrm{diag}(e^{\lambda_j t})V^{-1}y(0)$. This reduces the question of stability to the scalar case, with the λ now taking on the role of the eigenvalues of A.

using the same (or another) low-order method. Once $\tilde{\delta}(t)$ is computed, add it back to the interpolant to get

$$\tilde{y}(t) = \pi_n(t) + \tilde{\delta}(t).$$

This procedure can be repeated a few times to get the desired accuracy. (This version of deferred correction is actually quite recent and due to Dutt, Greengard, and Rokhlin, 2000).

5.3 Boundary-value problems

Boundary-value problems (BVP) are ODE where some feature of the solution is specified at two ends of an interval. The number of initial or boundary conditions matches the order of the highest derivative in the equation, hence such ODE are generally second-order scalar equations. The simplest examples are

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u(0) = a, u(1) = b \qquad \text{(Dirichlet)}$$

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u'(0) = a, u'(1) = b \qquad \text{(Neumann)}$$

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u(0) = u(1) \qquad \text{(periodic)}$$

We don't really mean to study the equation $-u'' = f$ for its own sake (you can find the general solution by integrating f twice and fixing the two constants from the boundary conditions), but its study serves two purposes:

- It is the simplest ODE that models problems in continuous elasticity: here u is the displacement of a vertical elastic bar, or rod, that sags under its own weight. The right-hand-side f is the gravitational force as a function of x, and u' is the "elongation", or strain of the bar. A condition where u is specified means that the bar is fixed at that end, while the condition $u' = 0$ would mean that that end is free. The condition $u(0) = u(1)$ means that it's an elastic band. By solving the ODE we are finding the displacement u generated by the force f.

- It is the simplest boundary-value problem to treat numerically, and contains many of the important features of such problems. It needs to be understood before moving on to any other example.

Alternative problems of this kind are

$$-u''(x) + \alpha(x)u(x) = f(x), \qquad u(0) = a, u(1) = b,$$

for instance, the solution of which does not generally obey an explicit formula.

A first intuitive method for solving BVP is the *shooting method*. Consider again $u''(x) + \alpha(x)u(x) = f(x), \quad u(0) = a, u(1) = b$. We cannot in general march in time from 0 to 1 (the variable x is more often a spatial variable), but we can guess what the derivative should have been for the solution to end up near b at $x = 1$. Introduce a parameter s, and write

$$-u''(x; s) + \alpha(x)u(x; s) = f(x), \qquad u(0; s) = a, u'(0; s) = s.$$

Of course in general we won't reach b at $x = 1$, but we can refine our estimate of s so that we get closer to it. The problem becomes that of solving for s in $u(1; s) = b$, where $u(1; s)$ is defined implicitly from the ODE. This equation that can be viewed as a root-finding or fixed-point problem and solved using any of the methods we've seen previously. The secant method only requires evaluations of $u(1; s)$, which involves solving the ODE. To set up Newton's method, we need the derivative $\frac{\partial u}{\partial s}(1; s)$ of the solution as a function of its parameter s. Exercise: find the value of this derivative by differentiating the ODE in the s parameter, and obtain an ODE for $\frac{\partial u}{\partial s}(1; s)$ itself.

The shooting method is easy and intuitive (and there isn't much more to say about it), but it has the disadvantage of being restricted to ODE. In contrast, we'll now investigate finite difference methods, which can also be applied to partial differential equations (PDE).

Let's start by considering the problem $-u'' = f$, $u(0) = u(1) = 0$. Consider the grid $x_j = jh$, $j = 0, \ldots, N$, $h = 1/N$. This grid has $N + 1$ points. The usual 3-point stencil for the centered second difference gives rise to

$$-\frac{U_{j+1} - 2U_j + U_{j-1}}{h^2} = f(x_j). \tag{5.1}$$

In this context, capital U denotes the numerical solution, while lowercase u denotes the exact solution. For the boundary conditions, we simply set $U_0 = U_N = 0$. The rest of the U_j for $j = 1, \ldots N - 1$ are unknowns and can be solved for from the linear system $KU = F$ generated by (5.1). The resulting matrix K (of size $N-1$ by $N-1$) is

$$K = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

The zero elements are not shown. The right-hand side is here $F_j = f(x_j)$. In Matlab, one then solves U as $K \setminus F$. Had the boundary conditions been $U_0 = a$ and $U_N = b$ instead, it is a good exercise to check that this does not change K, but that the right-hand side gets modified as

$$F = \begin{pmatrix} f(x_1) + \frac{a}{h^2} \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) + \frac{b}{h^2} \end{pmatrix}.$$

Of course, the matrix K should be invertible for this strategy to make sense. We will see below that this is the case. It is also important that K^{-1} be bounded for convergence of the numerical scheme.

Definition 9. *The local truncation error (LTE) of a numerical scheme $KU = F$, is the error made when evaluating the numerical scheme with the exact solution $u(x_j)$ in place of the numerical solution U_j. It is the quantity τ in*

$$Ku = F + \tau.$$

The local truncation is directly obtained from the truncation error of the finite-difference scheme, for instance

$$-\frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{h^2} = f(x_j) + O(h^2),$$

so the LTE is $O(h^2)$.

Definition 10. *The (actual) error of a numerical scheme $KU = F$, is the vector of differences $e_j = u(x_j) - U_j$.*

In order to obtain the error e from the LTE, one writes

$$Ku = F + \tau, \qquad KU = F,$$

and subtract those two equations. This gives $K(u - U) = (F - F) + \tau$, or $Ke = \tau$. If K is invertible, this gives

$$e = K^{-1}\tau.$$

The next few sections introduce the tools needed to control how large e can get from the knowledge that it is τ "magnified" by K^{-1}.

5.3.1 Matrix norms and eigenvalues

This section is mostly a linear algebra refresher.

Definition 11. *The spectral norm, or 2-norm, of (any rectangular, real) matrix A is*

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2},$$

where the vector 2-norm is $\|x\|_2 = \sqrt{\sum_i x_i^2}$. In other words, the matrix 2-norm is the maximum stretch factor for the length of a vector after applying the matrix to it.

Notice that, by definition,

$$\|Ax\|_2 \leq \|A\|_2 \|x\|_2,$$

for any vector x, so the matrix 2-norm is a very useful tool to write all sorts of inequalities involving matrices.

We can now characterize the 2-norm of a *symmetric* matrix as a function of its eigenvalues. The eigenvalue decomposition of a matrix A is, in matrix form, the equation $AV = V\Lambda$, where V contains the eigenvectors as columns, and Λ contains the eigenvalues of the diagonal (and zero elsewhere.) For those (non-defective) matrices for which there is a full count of eigenvectors, we also have

$$A = V\Lambda V^{-1}.$$

Symmetric matrices have a full set of orthogonal eigenvectors, so we can further write $V^T V = I$, $VV^T = I$ (i.e. V is unitary, a.k.a. a rotation), hence

$$A = V\Lambda V^T.$$

In terms of vectors, this becomes

$$A = \sum_i v_i \lambda_i v_i^T.$$

Note that λ_i are automatically real when $A = A^T$.

Theorem 9. *Let $A = A^T$. Then*

$$\|A\|_2 = \max_i |\lambda_i(A)|.$$

Proof. First observe that the vector 2-norm is invariant under rotations: if $V^T V = I$ and $VV^T = I$, then $\|Vx\|_2 = \|x\|_2$. This follows from the definition:

$$\|Vx\|_2^2 = (Vx)^T Vx = x^T V^T V x = x^T x = \|x\|_2^2.$$

Now fix x and consider now the ratio

$$\begin{aligned}
\frac{\|Ax\|_2^2}{\|x\|_2^2} &= \frac{\|V\Lambda V^T x\|_2^2}{\|x\|_2^2} \\
&= \frac{\|\Lambda V^T x\|_2^2}{\|x\|_2^2} \qquad \text{(unitary invariance)} \\
&= \frac{\|\Lambda y\|_2^2}{\|Vy\|_2^2} \qquad \text{(change variables)} \\
&= \frac{\|\Lambda y\|_2^2}{\|y\|_2^2} \qquad \text{(unitary invariance again)} \\
&= \frac{\sum_i \lambda_i^2 y_i^2}{\sum_i y_i^2}.
\end{aligned}$$

This quantity is maximized when y is concentrated to have nonzero component where λ_i is the largest (in absolute value): $y_j = 1$ when $|\lambda_j| = \max_n |\lambda_n|$, and zero otherwise. In that case,

$$\frac{\|Ax\|_2^2}{\|x\|_2^2} = \max_n \lambda_n^2,$$

the desired conclusion. □

Note: if the matrix is not symmetric, its 2-norm is on general not its largest eigenvalue (in absolute value). Instead, the 2-norm is the largest singular value (not material for this class, but very important concept.)

One very useful property of eigenvalues is that if $A = V\Lambda V^T$ is invertible, then

$$A^{-1} = V\Lambda^{-1}V^T$$

(which can be checked directly), and more generally

$$f(A) = Vf(\Lambda)V^T,$$

where the function f is applied componentwise to the λ_i (which can be checked when f has a Taylor expansion, and otherwise serves as the definition of $f(A)$). The eigenvectors of the function of a matrix are unchanged, and the eigenvalues are the function of the original eigenvalues.

If a matrix A is not invertible, things usually go wrong when trying to solve the linear system $Ax = b$.

Definition 12. *The nullspace of (any rectangular, real) matrix A is the space Null(A) of all vectors v such that $Av = 0$. In other words, it is the eigenspace corresponding to the eigenvalue zero.*

Null(A) always contains the zero vector. The following conditions are equivalent to characterize singular matrices:

- A is singular (non-invertible);
- Null(A) contains some nonzero vector;
- 0 is an eigenvalue of A;
- $\det(A) = 0$;
- The rows/columns are linearly dependent;
- (Zero is a pivot in the row echelon reduction of A.)

We now present a version of the inversion theorem for *symmetric* matrices. If the matrix is not symmetric, the statement looks quite different.

Theorem 10. *Let $A = A^T$. Consider the system $Ax = b$.*

- *If A is invertible, then the solution is unique and $x = A^{-1}b$.*
- *If Null(A) = span($v_1, \dots v_m$) $\neq 0$, then*

  - *If b has a component along any of the v_j (i.e., $v_j^T b \neq 0$ for some j), then the system has no solution.*
  - *If all $v_j^T b = 0$, $j = 1, \ldots, m$, then there exists an infinite number of solution to the system. If x_0 is a solution, then so is $x_0 + \sum_{j=1}^m c_j v_j$ for arbitrary coefficients c_j.*

In terms of eigenvectors v_j, if the matrix is invertible, the solution of $Ax = b$ is

$$x = \sum_{j=1}^N v_j \frac{1}{\lambda_j} v_j^T b.$$

If A is not invertible, but $v_j^T b = 0$ for all the eigenvectors v_j, $j = 1, \ldots, m$ corresponding to the zero eigenvalue (as in the theorem above), then we still have

$$x = \sum_{j=m+1}^N v_j \frac{1}{\lambda_j} v_j^T b + \sum_{j=1}^m c_j v_j,$$

where the first sum only runs from $m+1$ to N, and the coefficients c_j are arbitrary. (Apply the matrix A to this equation to see that they disappear.)

If $v_j^T b \neq 0$, then the operation $\frac{1}{\lambda_j} v_j^T b$ would result in an infinity when $\lambda_j = 0$ — a crude reason to see why the system has no solution in that case.

5.3.2 Properties of the matrix K

We are now equipped to study the matrix K and its inverse. Recall that we are interested in controlling $e = K^{-1}\tau$ to get the error from the LTE. From the results in the previous section, we know that

$$\|e\|_2 = \|K^{-1}\tau\|_2 \leq \|K^{-1}\|_2 \|\tau\|_2.$$

Since K is a symmetric matrix,

$$\|K^{-1}\|_2 = \max_j |\lambda_j(K^{-1})| = \frac{1}{\min_j |\lambda_j(K)|}.$$

So it remains for us to understand the eigenvalues of K, and specifically to show that the minimum eigenvalue (in absolute value) does not get too small. What we mean by this is that there should exist a number $c > 0$ such that

$$c < \min_j |\lambda_j(K)|,$$

independently of the grid spacing h (recall that the size of K and its entries depend on h.) Only in that case will we be able to conclude that $\|e\|$ is of the same order of magnitude as $\|\tau\|$.

Note in passing that if $\tau = O(h^2)$ then $\|\tau\|_2 = \sqrt{\sum_i \tau_i^2}$ will be $O(\sqrt{\frac{1}{h}h^4}) = O(h^{3/2})$, which is not very appealing. Instead, it is common to modify the vector 2-norm as

$$\|\tau\|_{2,h} = \sqrt{h \sum_i \tau_i^2},$$

so as to restore $\|\tau\|_{2,h} = O(h^2)$ (note how h times the sum resembles an integral quadrature.) In that case, we also expect $\|\tau\|_{2,h} = O(h^2)$. The reasoning with matrix 2-norms does not change one bit form this different choice of normalization.

Let us now study the eigenvalues and eigenvectors of K. The best way to guess them is to notice that $KU = F$ is a discretization of $-u'' = f$ with Dirichlet boundary conditions. The eigenvalue problem

$$-v'' = \lambda v, \qquad v(0) = v(1) = 0,$$

has a solution in terms of sines: for each $n \geq 1$, we have the pair

$$v_n(x) = \sin(n\pi x), \qquad \lambda = n^2\pi^2.$$

This analogy is very fruitful: the eigenvectors of K are precisely the v_n sampled at $x_j = jh$,

$$v_j^{(n)} = \sin(n\pi jh), \qquad j = 1, \ldots, N-1, \qquad n = 1, \ldots, N-1.$$

(here n is the label index, and j is the component index.) It is straightforward and a little tedious to check from trigonometric formulas that $v^{(n)}$ defined by this formula are indeed eigenvectors, with eigenvalues

$$\lambda_n = \frac{4}{h^2}\sin^2(\frac{\pi n h}{2}), \qquad n = 1, \ldots, N-1.$$

A Taylor expansion for small h shows that the minimum eigenvalue is $\lambda_1 = \pi^2 + O(h^2)$, and this $O(h^2)$ is positive, so that $\lambda_1 \geq \pi^2$.

Note in passing that since K is symmetric, the eigenvectors are automatically orthogonal, hence there is a way to normalize them so that $V^T V = I$

and $VV^T = I$. Applying the matrix V^T is called the discrete sine transform (DST), and applying the matrix V is called the inverse discrete sine transform (IDST).

The formula for the eigenvalues has two important consequences:

- The matrix K is invertible, now that it is clear that all its eigenvalues are positive. So setting up the discrete problem as $KU = F$ makes sense.

- Returning to the error bound, we can now conclude that $\|e\|_{2,h} \leq \frac{1}{\pi^2}\|\tau\|_{2,h} = O(h^2)$, establishing once and for all that the finite-difference method is second-order accurate for $-u'' = f$ with Dirichlet boundary conditions.

The reasoning of first establishing consistency (small LTE) and showing a stability result transferring at the level of the actual error is very common in numerical analysis. *Consistency + Stability = Convergence.*

5.3.3 Other boundary conditions and other equations

The boundary conditions (BC) play a crucial role in the description of the problem: if the BC changes, the solution changes completely, and so can the LTE and the error. For instance, let's return to the Neumann problem

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u'(0) = a, u'(1) = b \qquad \text{(Neumann)}$$

The discretization of the interior equations is the same as previously, but at least one extra equation needs to be added for the BC. For instance, at zero, we may write a forward difference

$$\frac{U_1 - U_0}{h} = a.$$

and similarly, a backward difference at $x = 1$. Each BC adds one row and one column to the original matrix K, and results in a different system of equations. The choice above leads to a first-order LTE, only $O(h)$, even though the LTE for the interior equations is $O(h^2)$. This is enough to spoil the error itself: we don't have $\|e\|_{2,h} = O(h^2)$ in general, as a result of the low-accuracy boundary conditions.

A more accurate way to discretize a Neumann boundary condition is to introduce a "ghost" node U_{-1}, and write

$$\frac{U_1 - U_{-1}}{2h} = a.$$

This new unknown is linked to the others by writing the equation one more time at $j = 0$,

$$\frac{= U_1 + 2U_0 - U_{-1}}{h^2} = f(x_0).$$

(Previously, we had only evaluated the equation at $j = 1, \ldots, N-1$.) This results in *two* additional rows and columns per BC. The same treatment should be applied at $x = 1$.

Note that these additional "boundary" rows can be scaled so that the resulting matrix looks symmetric, or at least more balanced. For instance, by rescaling the $\frac{U_1 - U_{-1}}{2h} = a$ by $2/h$ (including the right-hand side) we get the resulting matrix

$$T = \begin{pmatrix} -1 & 0 & 1 & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}$$

The eigenvalues depend on this choice of normalization!

Notice that, unlike the Dirichlet problem, the Neumann problem has a nullspace. The vector identically equal to 1 is in the nullspace of either of the 2 discretizations we have presented. As a result, there exists a solution only if the admissibility condition $1^T f = \sum_i f(x_i) = 0$ is satisfied (see a theorem in the previous section). This is also a feature of the non-discretized BVP: it has a solution if and only if $\int_0^1 f(x)\,dx = 0$.

The periodic problem is also very interesting:

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u(0) = u(1) \qquad \text{(periodic)}$$

The boundary condition $U_0 = U_N$ modifies K by adding two elements in the bottom-left and top-right:

$$C = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{pmatrix}$$

This new matrix C is circulant and singular. We can check that the vector 1 is in its nullspace, so the BVP has a solution if and only if $1^T f = \sum_i f(x_i) = 0$.

The eigenvectors of C are the Fourier components

$$v_j^{(n)} = w^{jn} = e^{2\pi ijn/N}, \qquad w = e^{2\pi i/N}.$$

Since C is symmetric, these $v^{(n)}$ are orthogonal (and orthonormal when divided by $\sqrt{N}$). To deal with vectors and matrices that have complex entries, don't forget that the transposes come with a complex conjugate, so the dot product is

$$x^* y = \overline{x}^T y = \sum_i \overline{x}_i y_i.$$

The norm is $\|x\|_2 = \sqrt{\sum_i |x_i|^2}$. The orthogonality relations for the matrix of eigenvectors are now $V^*V = I$ and $VV^* = I$, where $*$ is transpose conjugate.

Chapter 6

Fourier analysis

(Historical intro: the heat equation on a square plate or interval.)

Fourier's analysis was tremendously successful in the 19th century for formulating series expansions for solutions of some very simple ODE and PDE. This class shows that in the 20th century, Fourier analysis has established itself as a central tool for numerical computations as well, for vastly more general ODE and PDE when explicit formulas are not available.

6.1 The Fourier transform

We will take the Fourier transform of integrable functions of one variable $x \in \mathbb{R}$.

Definition 13. *(Integrability) A function f is called integrable, or absolutely integrable, when*

$$\int_{-\infty}^{\infty} |f(x)|\, dx < \infty,$$

in the sense of Lebesgue integration. One also writes $f \in L^1(\mathbb{R})$ for the space of integrable functions.

We denote the physical variable as x, but it is sometimes denoted by t in contexts in which its role is time, and one wants to emphasize that. The frequency, or wavenumber variable is denoted k. Popular alternatives choices for the frequency variable are ω (engineers) or ξ (mathematicians), or p (physicists).

Definition 14. *The Fourier transform (FT) of an integrable function $f(x)$ is defined as*

$$\hat{f}(k) = \int_{-\infty}^{\infty} e^{-ikx} f(x)\, dx. \tag{6.1}$$

When $\hat{f}(k)$ is also integrable, $f(x)$ can be recovered from $\hat{f}(k)$ by means of the inverse Fourier transform (IFT)

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} \hat{f}(k)\, dk. \tag{6.2}$$

Intuitively, $\hat{f}(k)$ is the amplitude density of f at frequency k. The formula for recovering f is a decomposition of f into constituent waves.

The justification of the inverse FT formula belongs in a real analysis class (where it is linked to the notion of approximate identity.) We will justify the form of (6.2) heuristically when we see Fourier series in the next section.

The precaution of assuming integrability is so that the integrals can be understood in the usual Lebesgue sense. In that context, taking integrals over infinite intervals is perfectly fine. If (6.1) and (6.2) are understood as limits of integrals over finite intervals, it does not matter how the bounds are chosen to tend to $\pm\infty$.

One may in fact understand the formulas for the FT and IFT for much larger function classes than the integrable functions, namely distributions, but this is also beyond the scope of the class. We will generally not overly worry about these issues. It is good to know where to draw the line: the basic case is that of integrable functions, and anything beyond that requires care and adequate generalizations.

Do not be surprised to see alternative formulas for the Fourier transfom in other classes or other contexts. Wikipedia lists them.

Here are some important properties of Fourier transforms:

- (Differentiation)

$$\widehat{f'}(k) = ik\hat{f}(k).$$

  Justification: integration by parts in the integral for the FT.

- (Translation) If $g(x) = f(x+a)$, then

$$\hat{g}(k) = e^{ika}\hat{f}(k).$$

  Justification: change of variables in the integral for the FT.

Let's see some examples of FT.

Example 17. *Let*

$$f(x) = \frac{1}{2a}\chi_{[-a,a]}(x) = \begin{cases} \frac{1}{2a} & \textit{if } x \in [-a,a]; \\ 0 & \textit{otherwise.} \end{cases}$$

Then

$$\hat{f}(k) = \frac{1}{2a}\int_{-a}^{a} e^{-ikx} dx = \frac{\sin(ka)}{ka}.$$

This function is a scaled version of the sinc function,

$$sinc(k) = \frac{\sin k}{k}.$$

It is easy to check by L'Hospital's rule that

$$sinc(0) = 1.$$

At $k \to \infty$, $sinc(k)$ decays like $1/k$, but does so by alternating between positive and negative values. It is a good exercise to check that sinc is not absolutely integrable. It turns out that the Fourier transform can still be defined for it, so lack of integrability is not a major worry.

Example 18. *Consider the Gaussian function*

$$f(x) = e^{-x^2/2}.$$

By completing the square and adequately modifying the contour of integration in the complex plane (not part of the material for this class), it can be shown that

$$\hat{f}(k) = \sqrt{2\pi}\, e^{-k^2/2}.$$

Example 19. *The Dirac delta $\delta(x)$ has a FT equal to 1 (why?).*

Another basic property of Fourier transforms is the convolution theorem.

Theorem 11. *(The convolution theorem.) Denote convolution as $f \star g(x) = \int_{-\infty}^{\infty} f(y)g(x-y)\, dy$. Then*

$$\widehat{f \star g}(k) = \hat{f}(k)\, \hat{g}(k).$$

Proof. Let $h = f \star g$. We can use Fubini below provided every function is integrable.

$$\begin{aligned}\hat{h}(k) &= \int e^{-ikx} \int f(y) g(x-y)\, dy dx \\ &= \int \int e^{-iky} f(y) e^{-ik(x-y))} g(x-y)\, dy dx \\ &= \left(\int e^{-iky} f(y)\, dy \right) \left(\int e^{-ikx'} g(x')\, dx' \right) \\ &= \hat{f}(k)\, \hat{g}(k).\end{aligned}$$

□

The Fourier transform is an important tool in the study of linear differential equations because it turns differential problems into algebraic problems. For instance, consider a polynomial $P(x) = \sum a_n x^n$, and the ODE

$$P\left(\frac{d}{dx}\right) u(x) = f(x), \qquad x \in \mathbb{R},$$

which means $\sum a_n \frac{d^n u}{dx^n} = f$. (Such ODE are not terribly relevant in real life because they are posed over the whole real line.) Upon Fourier transformation, the equation becomes

$$P(ik)\hat{u}(k) = \hat{f}(k),$$

which is simply solved as

$$\hat{u}(k) = \frac{\hat{f}(k)}{P(ik)},$$

and then

$$u(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} \frac{\hat{f}(k)}{P(ik)}\, dk.$$

Beware the zeros of P when applying this formula! They always carry important physical interpretation. For instance, they could be resonances of a mechanical system.

The formula $\hat{u}(k) = \frac{\hat{f}(k)}{P(ik)}$ also lends itself to an application of the convolution theorem. Let $K(x)$ be the inverse Fourier transform of $1/P(ik)$. Then we have

$$u(x) = \int K(x-y) f(y)\, dy.$$

The function K is called *Green's function* for the original ODE.

6.2 Sampling and restriction

We aim to use Fourier transforms as a concept to help understand the accuracy of representing and manipulating functions on a grid, using a finite number of degrees of freedom. We also aim at using a properly discretized Fourier transform as a numerical tool itself.

For this purpose, $x \in \mathbb{R}$ and $k \in \mathbb{R}$ must be replaced by x and k on finite grids. Full discretization consists of *sampling* and *restriction.*

Let us start by sampling $x \in h\mathbb{Z}$, i.e., considering $x_j = jh$ for $j \in \mathbb{Z}$. The important consequence of sampling is that some complex exponential waves e^{ikx} for different k will appear to be the same on the grid x_j. We call *aliases* such functions that identify on the grid.

Definition 15. *(Aliases) The functions e^{ik_1x} and e^{ik_2x} are aliases on the grid $x_j = jh$ if*

$$e^{ik_1x_j} = e^{ik_2x_j}, \qquad \forall j \in \mathbb{Z}.$$

Aliases happen as soon as

$$k_1jh = k_2jh + 2\pi \times \text{integer}(j).$$

Letting $j = 1$, and calling the integer n, we have

$$k_1 - k_2 = \frac{2\pi}{h}n,$$

for some $n \in Z$. Two wave numbers k_1, k_2 are indistinguishable on the grid if they differ by an integer multiple of $2\pi/h$.

For this reason, we restrict without loss of generality the wavenumber to the interval

$$k \in [-\pi/h, \pi/h].$$

We also call this interval the fundamental cell in frequency (in reference to a similar concept in crytallography.)

Real-life examples of aliases are rotating wheels looking like they go backwards in a movie, Moiré patterns on jackets on TV, and stroboscopy.

The proper notion of Fourier transform on a grid is the following.

Definition 16. *Let $x_j = hj, \qquad f_j = f(x_j)$. Semidiscrete Fourier transform (SFT):*

$$\hat{f}(k) = h \sum_{j=-\infty}^{\infty} e^{-ikx_j} f_j, \qquad k \in [-\pi/h, \pi/h]. \tag{6.3}$$

Inverse semidiscrete Fourier transform (ISFT):

$$f_j = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{ikx} \hat{f}(k)\, dk. \tag{6.4}$$

As we saw, sampling in x corresponds to a restriction in k. If one still wanted to peek outside $[-\pi/h, \pi/h]$ for the SFT, then the SFT would simply repeat by periodicity:

$$\hat{f}(k + \frac{2n\pi}{h}) = \hat{f}(k).$$

(why?). That's why we restrict it to the fundamental cell.

We can now define the proper notion of Fourier analysis for functions that are restricted to x in some interval, namely $[-\pi, \pi]$ for convention. Unsurprisingly, the frequency is sampled as a result. the following formulas are dual to those for the SFT.

Definition 17. *Fourier series (FS):*

$$\hat{f}_k = \int_{-\pi}^{\pi} e^{-ikx} f(x)\, dx. \tag{6.5}$$

Inverse Fourier series (IFS)

$$f(x) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} e^{ikx} \hat{f}_k, \qquad x \in [-\pi, \pi]. \tag{6.6}$$

If one uses the Fourier series inversion formula (6.6) for x outside of its intended interval $[-\pi, \pi]$, then the function simply repeats by periodicity:

$$f(x + 2n\pi) = f(x).$$

(again, why?)

The two formulas (6.5) and (6.6) can be justified quite intuitively. The expression $\int f(x)\overline{g(x)}\, dx$ is an inner product on functions. It is easy to see that the complex exponentials

$$\sqrt{\frac{h}{2\pi}} e^{-ikx_j} = v_j(k)$$

form an orthonormal set of functions on $[-\pi/h, \pi/h]$, for this inner product. Hence, up to normalization constants, (6.5) is simply calculation of the coefficients in an orthobasis (analysis), and (6.6) is the synthesis of the function back from those coefficients. We'd have to understand more about the peculiarities of infinite-dimensional linear algebra to make this fully rigorous, but this is typically done in a real analysis class.

Here's an example of SFT.

Example 20.

$$f_j = \frac{1}{2a}\chi_{[-a,a]}(x_j)$$

Then

$$\begin{aligned}\hat{f}(k) &= \frac{h}{2a}\sum_{j=-a}^{a} e^{-ikjh} \\ &= \frac{h}{2a}e^{ikah}\sum_{j=0}^{2a} e^{-ikjh} \\ &= \frac{h}{2a}e^{ikah}\,\frac{(e^{-ikh})^{2a+1}-1}{e^{-ikh}-1} \qquad \textit{(geometric series)} \\ &= \frac{h}{2a}\,\frac{\sin(kh(a+1/2))}{\sin(kh/2)}.\end{aligned}$$

This function is called the discrete sinc. It looks like a sinc, but it periodizes smoothly when $k = -\pi/h$ and $k = \pi/h$ are identified.

Our first slogan is therefore:

Sampling in x corresponds to restriction/periodization in k, and restriction/periodization in k corresponds to sampling in x.

6.3 The DFT and its algorithm, the FFT

The discrete Fourier transform is what is left of the Fourier transfom when both space and frequency are sampled and restricted to some interval.

Consider

$$x_j = jh, \qquad j = 1, \dots, N.$$

The point $j = 0$ is identified with $j = N$ by periodicity, so it is not part of the grid. If the endpoints are $x_0 = 0$ and $x_N = 2\pi$, then N and h relate as

$$h = \frac{2\pi}{N} \quad \Rightarrow \quad \frac{\pi}{h} = \frac{N}{2}.$$

For the dual grid in frequency, consider that N points should be equispaced between the bounds $[-\pi/h, \pi/h]$. The resulting grid is

$$k, \qquad k = -\frac{N}{2}+1, \ldots, \frac{N}{2}.$$

We have the following definition.

Definition 18. *Discrete Fourier transform (DFT):*

$$\hat{f}_k = h \sum_{j=1}^{N} e^{-ikjh} f_j, \qquad k = -\frac{N}{2}, \ldots, \frac{N}{2}. \tag{6.7}$$

Inverse discrete Fourier transform (IDFT)

$$f_j = \frac{1}{2\pi} \sum_{k=-N/2+1}^{N/2} e^{ikjh} \hat{f}_k, \qquad j = 1, \ldots, N. \tag{6.8}$$

The DFT can be computed as is, by implementing the formula (6.7) directly on a computer. The complexity of this calculation is a $O(N^2)$, since there are N values of j, and there are N values of k over which the computation must be repeated.

There is, however, a smart algorithm that allows to group the computation of all the f_k in complexity $O(N \log N)$. It is called the fast Fourier transform (FFT). It is traditionally due to Tukey and Cooley (1965), but the algorithm had been discovered a few times before that by people who are not usually credited as much: Danielson and Lanczos in 1942, and well as Gauss in 1805.

The trick of the FFT is to split the samples of f into even samples (j even) and odd samples (j odd). Assume that N is a power of 2. Then

$$\begin{aligned}
\hat{f}_k &= h \sum_{j=1}^{N} e^{-ikjh} f_j \\
&= h \sum_{j=1}^{N/2} e^{-ik(2j)h} f_{2j} + h \sum_{j=1}^{N/2} e^{-ik(2j+1)h} f_{2j+1} \\
&= h \sum_{j=1}^{N/2} e^{-ikj(2h)} f_{2j} + h e^{ikh} \sum_{j=1}^{N/2} e^{-ikj(2h)} f_{2j+1}.
\end{aligned}$$

The first term on the last line is simply the DFT of length $N/2$, on a grid of spacing $2h$, of the even samples of f. The second term is, besides the multiplicative factor, the DFT of length $N/2$, on a grid of spacing $2h$, of the odd samples of f.

Note that those smaller DFTs would normally be only calculated for $k = -\frac{N}{4}+1,\ldots,\frac{N}{4}$, but we need them for $k = -\frac{N}{2}+1,\ldots,\frac{N}{2}$. This is not a big problem: we know that the DFT extends by periodicity outside the standard bounds for k, so all there is to do is copy $\hat{f}_k$ by periodicity outside of $k = -\frac{N}{4}+1,\ldots,\frac{N}{4}$.

Already, we can see the advantage in this reduction: solving one problem of size N is essentially reduced to solving two problems of size $N/2$. Even better, the splitting into even and odd samples can be repeated recursively until the DFT are of size 1. When $N = 1$, the DFT is simply multiplication by a scalar.

At each stage, there are $O(N)$ operations to do to put together the summands in the equation of the last line above. Since there are $O(\log N)$ levels, the overall complexity is $O(N \log N)$.

There are variants of the FFT when N is not a power of 2.

6.4 Smoothness and truncation

In this section, we study the accuracy of truncation of Fourier transforms to finite intervals. This is an important question not only because real-life numerical Fourier transforms are restricted in k, but also because, as we know, restriction in k serves as a proxy for sampling in x. It will be apparent in Chapter 2, section 2.1 that every claim that we make concerning truncation of Fourier transforms will have an implication in terms of accuracy of sampling a function on a grid, i.e., how much information is lost in the process of sampling a function $f(x)$ at points $x_j = jh$.

We will manipulate functions in the spaces L^1, L^2, and L^∞. We have already encountered L^1.

Definition 19. *Let $1 \le p < \infty$. A function f of $x \in \mathbb{R}$ is said to belong to the space $L^p(\mathbb{R})$ when*

$$\int_{-\infty}^{\infty} |f(x)|^p \, dx < \infty.$$

Then the norm of f in $L^p(\mathbb{R})$ is $\|f\|_p = \left(\int_{-\infty}^{\infty} |f(x)|^p \, dx\right)^{1/p}$.

A function f of $x \in \mathbb{R}$ is said to belong to L^∞ when

$$\operatorname{ess\,sup} |f(x)| < \infty.$$

Then the norm of f in $L^\infty(\mathbb{R})$ is $\operatorname{ess\,sup} |f(x)|$.

In the definition above, "ess sup" refers to the essential supremum, i.e., the infimum over all dense sets $X \subset \mathbb{R}$ of the supremum of f over X. A set X is dense when $\mathbb{R}\backslash X$ has measure zero. The notions of supremum and infimum correspond to maximum and minimum respectively, when they are not necessarily attained. All these concepts are covered in a real analysis class. For us, it suffices to heuristically understand the L^∞ norm as the maximum value of the modulus of the function, except possibly for isolated points of discontinuity which don't count in calculating the maximum.

It is an interesting exercise to relate the L^∞ norm to the sequence of L^p norms as $p \to \infty$.

We will need the very important Parseval and Plancherel identitites. They express "conservation of energy" from the physical domain to the frequency domain.

Theorem 12. *(Parseval's identity). Let $f, g \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. Then*

$$\int_{-\infty}^{\infty} f(x)\overline{g(x)}\, dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(k)\overline{\hat{g}(k)}\, dk.$$

Proof. Let h be the convolution $f \star \tilde{g}$, where $\tilde{g}(x) = \overline{g}(-x)$. It is easy to see that the Fourier transform of $\tilde{g}$ is $\overline{\hat{g}(k)}$ (why?). By the convolution theorem (Section 1.1), we have

$$\hat{h}(k) = \hat{f}(k)\, \overline{\hat{g}(k)}.$$

If we integrate this relation over $k \in \mathbb{R}$, and divide by 2π, we get the IFT at $x = 0$:

$$h(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(k)\overline{\hat{g}(k)}\, dk.$$

On the other hand,

$$h(0) = \int_{-\infty}^{\infty} f(x)\overline{g(-(0-x))}\, dx = \int_{-\infty}^{\infty} f(x)\overline{g(x)}\, dx.$$

□

Theorem 13. *(Plancherel's identity). Let $f \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. Then*

$$\int_{-\infty}^{\infty} |f(x)|^2\,dx = \frac{1}{2\pi}\int_{-\infty}^{\infty} |\hat{f}(k)|^2\,dk.$$

Proof. Apply Parseval's identity with $g = f$. □

(With the help of these formulas, it is in fact possible to extend their validity and the validity of the FT to $f, g \in L^2(\mathbb{R})$, and not simply $f, g \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. This is a classical density argument covered in many good analysis texts.)

We need one more concept before we get to the study of truncation of Fourier transforms. It is the notion of total variation. We assume that the reader is familiar with the spaces $C^k(\mathbb{R})$ of bounded functions which are k times continuously differentiable.

Definition 20. *(Total variation) Let $f \in C^1(\mathbb{R})$. The total variation of f is the quantity*

$$\|f\|_{TV} = \int_{-\infty}^{\infty} |f'(x)|\,dx. \tag{6.9}$$

For functions that are not C^1, the notion of total variation is given by either expression

$$\|f\|_{TV} = \lim_{h\to 0}\int_{-\infty}^{\infty} \frac{|f(x)-f(x-h)|}{|h|}\,dx = \sup_{\{x_p\} \text{ finite subset of } \mathbb{R}} \sum_p |f(x_{p+1})-f(x_p)|, \tag{6.10}$$

These more general expressions reduce to $\int_{-\infty}^{\infty} |f'(x)|\,dx$ when $f \in C^1(\mathbb{R})$. When a function has finite total variation, we say it is in the space of functions of bounded variation, or $BV(\mathbb{R})$.

The total variation of a piecewise constant function is simply the sum of the absolute value of the jumps it undergoes. This property translates to a useful intuition about the total variation of more general functions if we view them as limits of piecewise constant functions.

The important meta-property of the Fourier transform is that

decay for large $|k|$ corresponds to smoothness in x.

There are various degrees to which a function can be smooth or rates at which it can decay, so therefore there are several ways that this assertion can be made precise. Let us go over a few of them. Each assertion either expresses a decay (in k) to smoothness (in x) implication, or the converse implication.

- *Let $\hat{f} \in L^1(\mathbb{R})$ (decay), then $f \in L^\infty(\mathbb{R})$ and f is continuous (smoothness).* That's because $|e^{ikx}| = 1$, so

$$|f(x)| \leq \frac{1}{2\pi} \int |e^{ikx} \hat{f}(k)| \, dk = \frac{1}{2\pi} \int |\hat{f}(k)| \, dk,$$

  which proves boundedness. As for continuity, consider a sequence $y_n \to 0$ and the formula

$$f(x - y_n) = \frac{1}{2\pi} \int e^{ik(x-y_n)} \hat{f}(k) \, dk.$$

  The integrand converges pointwise to $e^{ikx}\hat{f}(k)$, and is uniformly bounded in modulus by the integrable function $|\hat{f}(k)|$. Hence Lebesgue's dominated convergence theorem applies and yields $f(x - y_n) \to f(x)$, i.e., continuity in x.

- *Let $\hat{f}(k)(1 + |k|^p) \in L^1(\mathbb{R})$ (decay). Then $f \in C^p$ (smoothness).* We saw the case $p = 0$ above; the justification is analogous in the general case. We write

$$|f^{(n)}(x)| \leq \frac{1}{2\pi} \int |e^{ikx}(ik)^n \hat{f}(k)| \, dk \leq \int |k|^n |\hat{f}(k)| \, dk,$$

  which needs to be bounded for all $0 \leq n \leq p$. This is obviously the case if $\hat{f}(k)(1 + |k|^p) \in L^1(\mathbb{R})$. Continuity of $f^{(p)}$ is proved like before.

- *Let $f \in BV(\mathbb{R})$ (smoothness). Then $\hat{f}(k) \leq \|f\|_{TV} \, |k|^{-1}$ (decay).* If $f \in C^1 \cap BV(\mathbb{R})$, then this is justified very simply from (6.9), and

$$ik\hat{f}(k) = \int e^{-ikx} f'(x) \, dx.$$

  Take a modulus on both sides, and get the desired relation

$$|k||\hat{f}(k)| \leq \int |f'(x)| dx = \|f\|_{TV} < \infty.$$

When $f \in BV(\mathbb{R})$, but $f \notin C^1$, either of the more general formulas (6.10) must be used instead. It is a great practice exercise to articulate a modified proof using the $\lim_{h\to 0}$ formula, and properly pass to the limit.

- *Let f such that $f^{(k)} \in L^2(\mathbb{R})$ for $0 \le k < p$, and assume that $f^{(p)} \in BV(\mathbb{R})$ (smoothness). Then there exists $C > 0$ such that $|\hat{f}(k)| \le |k|^{-p-1}$ (decay).* This claim is also formulated as point (a) in Theorem 1 on page 30 of Trefethen's "Spectral methods in Matlab". The justification is very simple when $f \in C^{p+1}$: we then get

$$(ik)^{p+1}\hat{f}(k) = \int e^{-ikx} f^{(p+1)}(x)\, dx,$$

so

$$|k|^{p+1}|\hat{f}(k)| \le \int |f^{(p+1)}(x)|\, dx = \|f^{(p)}\|_{TV} < \infty.$$

Again, it is a good exercise to try and extend this result to functions not in C^{p+1}.

- (This is the one we'll use later). (Same proof as above.)

In summary, let f have p derivatives in L^1. Then $|\hat{f}(k)| \le C|k|^{-p}$. This is the form we'll make the most use of in what follows.

Example 21. *The canonical illustrative example for the two statements involving bounded variation is that of the B-splines. Consider*

$$s(x) = \frac{1}{2}\chi_{[-1,1]}(x),$$

the rectangle-shaped indicator of $[-1, 1]$ (times one half). It is a function in $BV(\mathbb{R})$, but it has no derivative in $L^2(\mathbb{R})$. Accordingly, its Fourier transform $\hat{s}(k)$ is predicted to decay at a rate $\sim |k|^{-1}$. This is precisely the case, for we know

$$\hat{s}(k) = \frac{\sin k}{k}.$$

Functions with higher regularity can be obtained by auto-convolution of s; for instance $s_2 = s \star s$ is a triangle-shaped function which has one derivative in

L^2, and such that $s_2' \in BV(\mathbb{R})$. We anticipate that $\hat{s_2}(k)$ would decay like $|k|^{-2}$, and this the case since by the convolution theorem

$$\hat{s_2}(k) = (\hat{s}(k))^2 = \left(\frac{\sin k}{k}\right)^2.$$

Any number of autoconvolutions $s \star s \ldots \star s$ can thus be considered: that's the family of B-splines.

The parallel between smoothness in x and decay in k goes much further. We have shown that p derivatives in x very roughly corresponds to an inverse-polynomial decay $|k|^{-p}$ in k. So C^∞ functions in x have so called super-algebraic decay in k: faster than $C_p|k|^{-p}$ for all $p \geq 0$.

The gradation of very smooth functions goes beyond C^∞ functions to include, in order:

- analytic functions that can be extended to a strip in the complex plane (like $f(x) = 1/(1+x^2)$), corresponding to a Fourier transform decaying exponentially (in this case $\hat{f}(k) = \pi e^{-|k|}$). That's Paley-Wiener theory, the specifics of which is not material for this course.

- analytic functions that can be extended to the whole complex plane with super-exponential growth (like $f(x) = e^{-x^2/2}$), whose Fourier transform decays faster than exponentially (in this case $\hat{f}(k) = \sqrt{\pi/2}e^{-k^2/2}$).

- analytic functions that can be extended to the whole complex plane with exponential growth at infinity (like $f(x) = \frac{\sin x}{x}$), whose Fourier transform is compactly supported (in this case $\hat{f}(k) = 2\pi\chi_{[-1,1]}(k)$). That's Paley-Wiener theory in reverse. Such functions are also called bandlimited.

More on this in Chapter 4 of Trefethen's book.

An important consequence of a Fourier transform having fast decay is that it can be truncated to some large interval $k \in [-N, N]$ at the expense of an error that decays fast as a function of N. The smoother f, the faster $\hat{f}$ decays, and the more accurate the truncation to large N in frequency. On the other hand, there may be convergence issues if $f(x)$ is not smooth.

To make this quantitative, put

$$\hat{f}_N(k) = \chi_{[-N,N]}(k)\hat{f}(k).$$

Recall that $\frac{1}{2\pi}\int_{-N}^{N} e^{ikx}dx = \frac{\sin(Nx)}{\pi x}$. By the convolution theorem, we therefore have

$$f_N(x) = \frac{\sin Nx}{\pi x} \star f(x).$$

In the presence of $f \in L^2(\mathbb{R})$, letting $N \to \infty$ always gives rise to convergence $f_N \to f$ in $L^2(\mathbb{R})$. This is because by Plancherel's identity,

$$\|f - f_N\|_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |\hat{f}_N(k) - \hat{f}(k)|^2 \, dk = \frac{1}{2\pi}\int_{|k|>N} |\hat{f}(k)|^2 \, dk.$$

This quantity tends to zero as $N \to \infty$ since the integral $\int_{\mathbb{R}} |\hat{f}(k)|^2$ over the whole line is bounded.

The story is quite different if we measure convergence in L^∞ instead of L^2, when f has a discontinuity. Generically, L^∞ (called uniform) convergence fails in this setting. This phenomenon is called *Gibb's effect*, and manifests itself through ripples in reconstructions from truncated Fourier transforms. Let us illustrate this on the Heaviside step function

$$u(x) = \begin{cases} 1 & \text{if } x \geq 0; \\ 0 & \text{if } x < 0. \end{cases}$$

(It is not a function in $L^2(\mathbb{R})$ but this is of no consequence to the argument. It could be modified into a function in $L^2(\mathbb{R})$ by some adequate windowing.) Since $u(x)$ is discontinuous, we expect the Fourier transform to decay quite slowly. Consider the truncated FT

$$\hat{u}_N(k) = \chi_{[-N,N]}(k)\hat{u}(k).$$

Back in the x domain, this is

$$\begin{aligned} u_N(x) &= \frac{\sin Nx}{\pi x} \star u(x) \\ &= \int_0^\infty \frac{\sin N(x-y)}{\pi(x-y)}\, dy \\ &= \int_{-\infty}^{Nx} \frac{\sin y}{\pi y}\, dy \\ &\equiv s(Nx). \end{aligned}$$

(Draw picture)

The function $s(Nx)$ is a sigmoid function going from 0 at $-\infty$ to 1 at ∞, going through $s(0) = 1/2$, and taking on min. value $\simeq -.045$ at $x = -\pi/N$ and max. value $\simeq 1.045$ at $x = \pi/N$. The oscillations have near-period π/N. The parameter N is just a dilation index for the sigmoid, it does not change the min. and max. values of the function. Even as $N \to \infty$, there is no way that the sigmoid would converge to the Heaviside step in a uniform manner: we always have

$$\|u - u_N\|_\infty \gtrsim .045.$$

This example of Gibb's effect goes to show that truncating Fourier expansions could be a poor numerical approximation if the function is nonsmooth.

However, if the function is smooth, then everything goes well. Let us study the decay of the approximation error

$$\epsilon_N^2 = \|f - f_N\|_2^2$$

for truncation of the FT to $[-N, N]$. Again, we use Plancherel to get

$$\epsilon_N^2 = \int_{|k|>N} |\hat{f}(k)|^2 \, dk$$

Now assume that $\hat{f}(k) \leq C|k|^{-p}$, a scenario already considered earlier. Then

$$\epsilon_N^2 \leq C \int_{|k|>N} |k|^{-2p} \leq C' \, N^{-2p+1},$$

so, for some constant C'' (dependent on p but independent of N, hence called constant),

$$\epsilon_N \leq C'' \, N^{-p+1/2}.$$

The larger p, the faster the decay as $N \to \infty$.

When a function is C^∞, the above decay rate of the approximation error is valid for all $p > 0$. When the function is analytic and the Fourier transform decays exponentially, the decay rate of the approximation error is even faster, itself exponential (a good exercise). In numerical analysis, either such behavior is called *spectral accuracy.*

6.5 Chebyshev expansions

In the previous sections, we have seen that smooth functions on the real line have fast decaying Fourier transforms.

On a finite interval, a very similar property hold for Fourier series: if a function is smooth in $[-\pi, \pi]$ and connects smoothly by periodicity at $x = -\pi$ and $x = \pi$, then its Fourier series (6.5) decays fast. Periodicity is essential, because the points $x = -\pi$ and $x = \pi$ play no particular role in (6.5). They might as well be replaced by $x = 0$ and $x = 2\pi$ for the integration bounds. So if a function is to qualify as smooth, it has to be equally smooth at $x = 0$ as it is at $x = -\pi$, identified with $x = \pi$ by periodicity.

For instance if a function f is smooth inside $[-\pi, \pi]$, but has $f(-\pi) \neq f(\pi)$, then for the purpose of convergence of Fourier series, f is considered discontinuous. We know what happens in this case: the Gibbs effect takes place, partial inverse Fourier series have unwelcome ripples, and convergence does not occur in L^∞.

If f is smooth and periodic, then it is a good exercise to generalize the convergence results of the previous section, from the Fourier transform to Fourier series.

How shall we handle smooth functions in intervals $[a, b]$, which do not connect smoothly by periodicity? The answer is not unique, but the most standard tool for this in numerical analysis are the *Chebyshev polynomials*.

For simplicity consider $x \in [-1, 1]$, otherwise rescale the problem. Take a C^k nonperiodic $f(x)$ in $[-1, 1]$. The trick is to view it as $g(\theta) = f(\cos\theta)$. Since $\cos[0, \pi] = \cos[\pi, 2\pi] = [-1, 1]$, all the values of $x \in [-1, 1]$ are covered twice by $\theta \in [0, 2\pi]$. Obviously, at any point θ, g inherits the smoothness of f by the chain rule. Furthermore, g is periodic since $\cos\theta$ is periodic. So g is exactly the kind of function which we expect should have a fast converging Fourier series:

$$\hat{g}_k = \int_0^{2\pi} e^{-ik\theta} g(\theta)\, d\theta, \qquad g(\theta) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} e^{ik\theta} \hat{g}_k, \qquad k \in \mathbb{Z}.$$

Since $g(\theta)$ is even in θ, we may drop the $\sin\theta$ terms in the expansion, as well as the negative k:

$$\hat{g}_k = \int_0^{2\pi} \cos(k\theta) g(\theta)\, d\theta, \qquad g(\theta) = \frac{1}{2\pi}\hat{g}_0 + \frac{1}{\pi} \sum_{k=1}^{\infty} \cos(k\theta) \hat{g}_k, \qquad k \in \mathbb{Z}^+.$$

Back to f, we find

$$\hat{g}_k = 2\int_{-1}^{1} \cos(k \arccos x) f(x) \frac{dx}{\sqrt{1-x^2}}, \qquad f(x) = \frac{1}{2\pi}\hat{g}_0 + \frac{1}{\pi} \sum_{k=1}^{\infty} \cos(k \arccos x) \hat{g}_k, \qquad k \in \mathbb{Z}^+.$$

The function $\cos(k \arccos x)$ happens to be a polynomial in x, of order k, called the *Chebyshev polynomial* of order k. Switch to the letter n as is usually done:

$$T_n(x) = \cos(n \arccos x), \qquad x \in [-1, 1].$$

The first few Chebyshev polynomials are

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1.$$

It is a good exercise (involving trigonometric identities) to show that they obey the recurrence relation

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

The orthogonality properties of $T_n(x)$ over $[-1, 1]$ follows from those of $\cos(n\theta)$ over $[0, \pi]$, but watch the special integration weight:

$$\int_{-1}^{1} T_m(x)T_n(x) \, \frac{dx}{\sqrt{1-x^2}} = c_n \delta_{mn},$$

with $c_n = \pi/2$ if $n = 0$ and $c_n = \pi$ otherwise. The Chebyshev expansion of f is therefore

$$\langle f, T_n \rangle = \int_{-1}^{1} T_n(x) f(x) \, \frac{dx}{\sqrt{1-x^2}}, \qquad f(x) = \frac{1}{\pi}\langle f, T_0 \rangle + \frac{2}{\pi} \sum_{n=1}^{\infty} T_n(x) \langle f, T_n \rangle, \qquad k \in \mathbb{Z}^+.$$

Under the hood, this expansion is the FS of a periodic function, so we can apply results pertaining to Fourier series and Fourier transforms to obtain fast decay of Chebyshev expansions. This way, spectral accuracy is restored for nonperiodic functions defined on an interval.

Chapter 7

Spectral Interpolation, Differentiation, Quadrature

7.1 Interpolation

7.1.1 Bandlimited interpolation

While equispaced points generally cause problems for polynomial interpolation, as we just saw, they are the natural choice for discretizing the Fourier transform. For data on $x_j = jh$, $j \in \mathbb{Z}$, recall that the semidiscrete Fourier transform (SFT) and its inverse (ISFT) read

$$\hat{f}(k) = h \sum_{j \in \mathbb{Z}} e^{-ikx_j} f_j, \qquad f_j = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{ikx_j} \hat{f}(k)\, dk.$$

The idea of spectral interpolation, or bandlimited interpolation, is to evaluate the ISFT formula above at some point x not equal to one of the x_j.

Definition 21. *(Fourier/spectral/bandlimited interpolation on $\mathbb{R}$) Let $x_j = jh$, $j \in \mathbb{Z}$. Consider $f : \mathbb{R} \mapsto \mathbb{R}$, its restriction $f_j = f(x_j)$, and the SFT $\hat{f}(k)$ of the samples f_j. Then the spectral interpolant is*

$$p(x) = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{ikx} \hat{f}(k)\, dk.$$

We can view the formula for $p(x)$ as the inverse Fourier transform of the compactly supported function equal to $\hat{f}(k)$ for $k \in [-\pi/h, \pi/h]$, and zero otherwise. When the Fourier transform of a function is compactly supported, we say that that function is *bandlimited*, hence the name of the interpolation scheme.

Example 22. *Let*

$$f_j = \delta_{0j} = \begin{cases} 1 & \text{if } j = 0; \\ 0 & \text{if } j \neq 0. \end{cases}$$

Then the SFT is $\hat{f}(k) = h$ for $k \in [-\pi/h, \pi/h]$ as we saw previously. Extend it to $k \in \mathbb{R}$ by zero outside of $[-\pi/h, \pi/h]$. Then

$$p(x) = \frac{\sin(\pi x/h)}{\pi x/h} = sinc(\pi x/h).$$

This function is also called the Dirichlet kernel. It vanishes at $x_j = jh$ for $j \neq 0$, integer.

Example 23. *In full generality, consider now the sequence*

$$f_j = \sum_{k\in\mathbb{Z}} \delta_{jk} f_k.$$

By linearity of the integral,

$$p(x) = \sum_{k\in\mathbb{Z}} f_k sinc(\pi(x - x_k)/h).$$

The interpolant is a superposition of sinc functions, with the samples f_j as weights. Here sinc is the analogue of the Lagrange elementary polynomials of a previous section, and is called the interpolation kernel. *For this reason, bandlimited interpolation sometimes goes by the name Fourier-sinc interpolation.*

(Figure here for the interpolation of a discrete step.)

In the example above we interpolate a discontinuous function, and the result is visually not very good. It suffers from the same Gibbs effect that we encountered earlier. The smoother the underlying $f(x)$ which f_j are the samples of, however, the more accurate bandlimited interpolation.

In order to study the approximation error of bandlimited interpolation, we need to return to the link between SFT and FT. The relationship between $p(x)$ and f_j is *sampling*, whereas the relationship between the FT

$\hat{f}(k)\chi_{[-\pi/h,\pi/h](k)}$ and the SFT $\hat{f}(k)$ is *periodization.* We have already alluded to this correspondence earlier, and it is time to formulate it more precisely.

(Figure here; sampling and periodization)

Theorem 14. *(Poisson summation formula, FT version) Let $u : \mathbb{R} \mapsto \mathbb{R}$, sufficiently smooth and decaying sufficiently fast at infinity. (We are deliberately imprecise!) Let $v_j = u(x_j)$ for $x_j = jh$, $j \in \mathbb{Z}$, and*

$$\hat{u}(k) = \int_{\mathbb{R}} e^{-ikx} u(x)\, dx, \qquad (FT), \qquad k \in \mathbb{R},$$

$$\hat{v}(k) = h \sum_{j \in \mathbb{Z}} e^{-ikx_j} u(x_j). \qquad (SFT), \qquad k \in [-\pi/h, \pi/h].$$

Then

$$\hat{v}(k) = \sum_{m \in \mathbb{Z}} \hat{u}(k + m\frac{2\pi}{h}), \qquad k \in [-\pi/h, \pi/h] \tag{7.1}$$

In some texts the Poisson summation formula is written as the special case $k = 0$:

$$h \sum_{j \in \mathbb{Z}} u(jh) = \sum_{m \in \mathbb{Z}} \hat{u}(\frac{2\pi}{h} m).$$

Exercise: use what we have already seen concerning translations and Fourier transforms to show that the above equation implies (hence is equivalent to) equation (7.1).

Proof. Consider the right-hand side in (7.1), and call it

$$\hat{\phi}(k) = \sum_{m \in \mathbb{Z}} \hat{u}(k + m\frac{2\pi}{h}), \qquad k \in [-\pi/h, \pi/h].$$

It suffices to show that $\hat{\phi}(k) = \hat{v}(k)$, or equivalently in terms of their ISFT, that $\phi_j = v_j$ for $j \in \mathbb{Z}$. The ISFT is written

$$\phi_j = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} \left[\sum_{m \in \mathbb{Z}} \hat{u}(k + m\frac{2\pi}{h}) \right] e^{ikjh}\, dk.$$

The function u is smooth, hence integrable, and the sum over m converges fast. So we can interchange sum and integral:

$$\phi_j = \frac{1}{2\pi} \sum_{m \in \mathbb{Z}} \int_{-\pi/h}^{\pi/h} \hat{u}(k + m\frac{2\pi}{h}) e^{ikjh} \, dk.$$

Now put $k' = k + m\frac{2\pi}{h}$, and change variable:

$$\phi_j = \frac{1}{2\pi} \sum_{m \in \mathbb{Z}} \int_{-\frac{\pi}{h} - m\frac{2\pi}{h}}^{\frac{\pi}{h} - m\frac{2\pi}{h}} \hat{u}(k') e^{ik'jh} e^{-i\frac{2\pi}{h}jh} \, dk'.$$

The extra exponential factor $e^{-i\frac{2\pi}{h}jh}$ is equal to 1 because $j \in \mathbb{Z}$. We are in presence of an integral over $\mathbb{R}$ chopped up into pieces corresponding to sub-intervals of length $2\pi/h$. Piecing them back together, we get

$$\phi_j = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{u}(k') e^{ik'jh} dk',$$

which is exactly the inverse FT of $\hat{u}$ evaluated at $x_j = jh$, i.e., $\phi_j = u(x_j) = v_j$. □

The Poisson summation formula shows that sampling a function at rate h corresponds to periodizing its spectrum (Fourier transform.) with a period $2\pi/h$. So the error made in sampling a function (and subsequently doing bandlimited interpolation) is linked to the possible overlap of the Fourier transform upon periodization.

- **Scenario 1**. Assume $\text{supp}(\hat{u}) \subset [-\pi/h, \pi/h]$. Then no error is made in sampling and interpolating u at rate h, because nothing happens upon $2\pi/h$-periodization and windowing into $[-\pi/h, \pi/h]$:

  $$\hat{p}(k) = \hat{u}(k) \qquad \Rightarrow \qquad p(x) = u(x).$$

  (Draw picture)

- **Scenario 2**. Now assume that $\hat{u}(k)$ is not included in $[-\pi/h, \pi/h]$. In general the periodization of $\hat{u}(k)$ will result in some overlap inside $[-\pi/h, \pi/h]$. We call this *aliasing*. In that case, some information is lost and interpolation will not be exact.

Scenario 1 is known as the *Shannon sampling theorem*: a function bandlimited in $[-\pi/h, \pi/h]$ in k space is perfectly interpolated by bandlimited interpolation, on a grid of spacing h or greater. In signal processing h is also called the sampling rate, because x has the interpretation of time. When h is the largest possible rate such that no aliasing occurs, it can be referred to as the Nyquist rate.

More generally, the accuracy of interpolation is linked to the smoothness of $u(x)$. If the tails $\hat{u}(k)$ are small and h is large, we can expect that the error due to periodization and overlap won't be too big. The following result is a consequence of the Poisson summation formula.

Theorem 15. *(Error of bandlimited interpolation) Let u have $p \geq 1$ derivatives in $L^1(\mathbb{R})$. Let $v_j = u(x_j)$ at $x_j = jh$, $j \in \mathbb{Z}$. Denote by $p(x)$ the bandlimited interpolant formed from v_j. Then, as $h \to 0$,*

$$|\hat{u}(k) - \hat{p}(k)| = O(h^p) \qquad |k| \leq \frac{\pi}{h},$$

and

$$\|u - p\|_2 = O(h^{p-1/2}).$$

Proof. Denote by $\hat{u}(k)$ the FT of $u(x)$, and by $\hat{v}(k)$ the SFT of v_j, so that $\hat{p}(k) = \hat{v}(k)$ on $[-\pi/h, \pi/h]$. By the Poisson summation formula (7.1),

$$\hat{v}(k) - \hat{u}(k) = \sum_{m \neq 0} \hat{u}(k + m\frac{2\pi}{h}), \qquad k \in [-\pi/h, \pi/h].$$

As we saw earlier, the smoothness condition on u imply that

$$|\hat{u}(k)| \leq C\,|k|^{-p}.$$

Since

$$k + m\frac{2\pi}{h} \in [-\frac{\pi}{h} + m\frac{2\pi}{h}, \frac{\pi}{h} + m\frac{2\pi}{h}],$$

we have $|k + m\frac{2\pi}{h}| \geq |m\frac{\pi}{h}|$, hence

$$|\hat{u}(k + m\frac{2\pi}{h})| \leq C'\,|m\frac{\pi}{h}|^{-p},$$

for some different constant C'. Summing over $m \neq 0$,

$$|\hat{v}(k) - \hat{u}(k)| \leq C' \sum_{m \neq 0} |m|^{-p} (\frac{\pi}{h})^{-p} \leq C''\,(\frac{\pi}{h})^{-p} \leq C'''\,h^p.$$

One can switch back to the x domain by means of the Plancherel formula

$$\|u - p\|_{L^2}^2 = \frac{1}{2\pi} \|\hat{u}(k) - \hat{v}(k)\|_{L^2(\mathbb{R})}^2.$$

The right-hand side contains the integral of $|\hat{u}(k) - \hat{v}(k)|^2$ over $\mathbb{R}$. Break this integral into two pieces:

- Over $[-\pi/h, \pi/h]$, we have seen that $|\hat{u}(k) - \hat{v}(k)|^2 = O(h^{2p})$. The integral is over an interval of length $O(1/h)$, hence the L^2 norm squared is $O(h^{2p-1})$. Taking a square root to get the L^2 norm, we get $O(h^{p-1/2})$.
- For $|k| \geq \pi/h$, we have $\hat{p}(k) = 0$, so it suffices to bound $\int_{|k|\geq \pi/h} |\hat{u}(k)|^2 \, dk$. Since $|\hat{u}(k)| \leq C \, |k|^{-p}$, this integral is bounded by $O((\pi/h)^{2p-1})$. Taking a square root, we again obtain a $O(h^{p-1/2})$.

$\square$

We have seen how to interpolate a function defined on $\mathbb{R}$, but as a closing remark let us notice that a similar notion exists for functions defined on intervals, notably $x \in [-\pi, \pi]$ or $[0, 2\pi]$. In that case, wavenumbers are discrete, the FT is replaced by the FS, and the SFT is replaced by the DFT. Evaluating the DFT for x not on the grid would give an interpolant:

$$\frac{1}{2\pi} \sum_{k=-N/2+1}^{N/2} e^{ikx} \hat{f}_k.$$

Contrast with the formula (6.8) for the IDFT. This is almost what we want, but not quite, because the highest wavenumber $k = N/2$ is treated asymmetrically. It gives rise to an unnatural complex term, even if $\hat{f}_k$ is real and even. To fix this, it is customary to set $\hat{f}_{-N/2} = \hat{f}_{N/2}$, to extend the sum from $-N/2$ to $N/2$, but to halve the terms corresponding to $k = -N/2$ and $N/2$. We denote this operation of halving the first and last term of a sum by a double prime after the sum symbol:

$$\sum{}''$$

Itis easy to check that this operation does not change the interpolating property. The definition of bandlimited interpolant becomes the following in the case of intervals.

Definition 22. *(Spectral interpolation on $[0, 2\pi]$) Let $x_j = jh$, $j = 0, \ldots, N-1$ with $h = 1/N$. Consider $f : [0, 2\pi] \mapsto \mathbb{R}$, its restriction $f_j = f(x_j)$, and the DFT $\hat{f}_k$ of the samples f_j. Then the spectral interpolant is*

$$p(x) = \frac{1}{2\pi} {\sum}''^{N/2}_{k=-N/2} e^{ikx} \hat{f}_k.$$

Because $p(x)$ is a superposition of "monomials" of the form $e^{ikx} = (e^{ix})^k$ for k integer, we call it a *trigonometric polynomial.*

The theory for interpolation by trigonometric polynomials (inside $[0, 2\pi]$) is very similar to that for general bandlimited interpolants. The only important modification is that $[0, 2\pi]$ is a periodized interval, so a function qualifies as smooth only if it connects smoothly across $x = 0$ indentified with $x = 2\pi$ by periodicity. The Poisson summation formula is still the central tool, and has a counterpart for Fourier series.

Theorem 16. *(Poisson summation formula, FS version) Let $u : [0, 2\pi] \mapsto \mathbb{R}$, sufficiently smooth. Let $v_j = u(\theta_j)$ for $\theta_j = jh$, $j = 1, \ldots, N$, $h = 2\pi/N$, and*

$$\hat{u}_k = \int_0^{2\pi} e^{-ik\theta} u(\theta)\, d\theta, \qquad (FS), \qquad k \in \mathbb{Z},$$

$$\hat{v}_k = h \sum_{j=1}^{N} e^{-ik\theta_j} u(\theta_j). \qquad (DFT), \qquad k = -\frac{N}{2}, \ldots, \frac{N}{2} - 1.$$

Then

$$\hat{v}_k = \sum_{m \in \mathbb{Z}} \hat{u}_{k+mN}, \qquad k = -\frac{N}{2}, \ldots, \frac{N}{2} - 1. \tag{7.2}$$

(Recall that $N = \frac{2\pi}{h}$ so this formula is completely analogous to (7.1).)

For us, the important consequence is that if u has p derivatives in L^1, over the periodized interval $[0, 2\pi]$, then the bandlimited interpolation error is a $O(h^p)$ in the pointwise sense in k space, and $O(h^{p-1/2})$ in the L^2 sense.

For this result to hold it is important that u has p derivatives at the origin as well (identified by periodicity with 2π), i.e., the function is equally smooth as it straddles the point where the interval wraps around by periodicity.Otherwise, if $u(\theta)$ has discontinuities such as. $u(2\pi^-) \neq u(0^+)$, interpolation will suffer from the Gibbs effect.

7.1.2 Chebyshev interpolation

Consider now smooth functions inside $[-1, 1]$ (for illustration), but not necessarily periodic. So the periodization $\sum_j f(x + 2j)$ may be discontinuous. Polynomial interpolation on equispaced points may fail because of the Runge phenomenon, and bandlimited interpolation will fail due to Gibbs's effect.

A good strategy for interpolation is the same trick as the one we used for truncation in the last chapter: pass to the variable θ such that

$$x = \cos\theta.$$

Then $x \in [-1, 1]$ corresponds to $\theta \in [0, \pi]$. We define $g(\theta) = f(\cos\theta)$ with g 2π-periodic and even.

We can now consider the bandlimited interpolant of $g(\theta)$ on an equispaced grid covering $[0, 2\pi]$, like for instance

$$\theta_j = \frac{\pi j}{N}, \quad \text{where } j = 1, \ldots, 2N.$$

Using the definition we saw at the end of the last section (with the "double prime"), we get

$$q(\theta) = \frac{1}{2\pi} {\sum}''^{\,N}_{k=-N} e^{ik\theta} \hat{g}_k, \qquad \hat{g}_k = \frac{\pi}{N} \sum_{j=0}^{2N-1} e^{-ik\theta_j} g(\theta_j).$$

By even symmetry in θ and k (why?), we can write

$$q(\theta) = \sum_{k=0}^{N} \cos(k\theta) c_k,$$

with

$$c_0 = \frac{\hat{g}_0}{2\pi}, \qquad c_k = \frac{\hat{g}_k}{\pi} \text{ for } k \neq 0.$$

Note that the c_k are determined from the samples $g(\theta_j)$.

Back to x, we get the sample points $x_j = \cos(\theta_j)$. They are called *Chebyshev points*. They are not equispaced anymore, and because they are the projection on the x-axis of equispaced points on the unit circle, they cluster near the edges of $[-1, 1]$. There are $N + 1$ of them, from $j = 0$ to N, because of the symmetry in θ. It turns out that the x_j are the *extremal points* of the

Chebyshev polynomial $T_N(x)$, i.e., the points in $[-1, 1]$ where T_N takes its maximum and minimum values.

In terms of the variable x, the interpolant can be expressed as

$$p(x) = q(\text{acos } x) = \sum_{n=0}^{N} T_n(x) c_n, \qquad T_n(x) = \cos(n \,\text{acos } x),$$

with the same c_n as above. Since $T_n(x)$ are polynomials of degree $0 \leq n \leq N$, and $p(x)$ interpolates $f(x)$ at the $N+1$ (non-equispaced) points x_j, we are in presence of the Lagrange interpolation polynomial for f at x_j!

In practice, the formula in terms of $T_n(x)$ may not even be needed. From interpolation points $\tilde{x}_j$, it suffices to map them to $\tilde{\theta}_j = \text{acos } \tilde{x}_j$, then get the interpolated values as $q(\text{acos } \tilde{x}_j)$. For instance, when $\tilde{x}_j$ are Chebyshev points on a grid twice finer than x_j, namely $\tilde{x}_j = \text{acos } \frac{\pi j}{2N}$ for $j = 1, \ldots, 4N$, it is a good exercise to write a fast FFT-based algorithm for Chebyshev interpolation.

The interesting conclusion is that, although this is a polynomial interpolant, the error analysis is inherited straight from Fourier analysis. If f is smooth, then g is smooth and periodic, and we saw that the bandlimited interpolant converges very fast. The Chebyshev interpolant of f is equal to the bandlimited interplant of g so it converges at the exact same rate (in L^∞ in k or n space) — for instance $O(N^{-p-1})$ when f has p derivatives (in BV).

In particular, we completely bypassed the standard analysis of error of polynomial interpolation, and proved universal convergence for smooth f. The factor

$$\pi_{N+1}(x) = \prod_{j=0}^{N} (x - x_j)$$

that was posing problems in the error estimate then does not pose a problem anymore, because of the very special choice of Chebyshev points $\cos(\pi j/N)$ for the interpolation. Intuitively, clustering the grid points near the edges of the interval $[-1, 1]$ helps giving $\pi_{N+1}(x)$ more uniform values throughout $[-1, 1]$, hence reduces the gross errors near the edges.

Let us now explain the differences in the behavior of the monic polynomial $\prod_{j=0}^{N}(x - x_j)$ for equispaced vs. Chebyshev points, and argue that Chebyshev points are near-ideal for interpolation of smooth functions in intervals. The discussion below is mostly taken from Trefethen, p.43. (See also the last problem on homework 2 for an example of different analysis.)

Let $p(z) = \prod_{j=0}^N (z - x_j)$, where we have extended the definition of the monic polynomial to $z \in \mathbb{C}$. We compute

$$\log |p(z)| = \sum_{j=0}^N \log |z - x_j|,$$

Put

$$\phi_N(z) = (N+1)^{-1} \sum_{j=0}^N \log |z - x_j|.$$

The function ϕ_N is like an electrostatic potential, due to charges at $z = x_j$, each with potential $(N+1)^{-1} \log |z - x_j|$. Going back to $p(z)$ from ϕ_N is easy:

$$p(z) = e^{(N+1)\phi_N(z)}.$$

Already from this formula, we can see that small variations in ϕ_N will lead to exponentially larger variations in $p(z)$, particularly for large N.

Let us now take a limit $N \to \infty$, and try to understand what happens without being too rigorous. What matters most about the Chebyshev points is their *density*: the Chebyshev points are the projection onto the real-axis of a sequence of equispaced points on a circle. If the density of points on the circle is a constant $1/(2\pi)$, then the density of points on $[-1, 1]$ generated by vertical projection is

$$\rho_{Cheb}(x) = \frac{1}{\pi\sqrt{1 - x^2}}. \qquad \text{(normalized to integrate to 1 on } [-1,1])$$

(This is a density in the sense that

$$N \int_a^b \rho_{Cheb}(x)\, dx$$

approximately gives the number of points in $[a, b]$.) Contrast with a uniform distribution of points, with density

$$\rho_{equi}(x) = \frac{1}{2}.$$

Then the potential corresponding to any given $\rho(x)$ is simply

$$\phi(z) = \int_{-1}^1 \rho(x) \log |z - x|\, dx.$$

The integral can be solved explicitly for both densities introduced above:

- For ρ_{equi}, we get

  $$\phi_{equi}(z) = -1 + \frac{1}{2} Re((z+1)\log(z+1) - (z-1)\log(z-1)).$$

  It obeys $\phi_{equi}(0) = -1$, $\phi_{equi}(\pm 1) = -1 + \log 2$.

- For ρ_{Cheb}, we get

  $$\phi_{Cheb}(z) = \log \frac{|z - \sqrt{z^2 - 1}|}{2}.$$

  This function obeys (interesting exercise) $\phi_{Cheb}(x) = -\log 2$ for all $x \in [-1, 1]$ on the real axis.

The level curves of both ϕ_{equiv} and ϕ_{Cheb} in the complex plane are shown on page 47 of Trefethen.

Overlooking the fact that we have passed to a continuum limit for the potentials, we can give a precise estimate on the monic polynomial:

$$|p_{equi}(z)| \simeq e^{(N+1)\phi_{equi}(z)} = \begin{cases} (2/e)^N & \text{near } x = \pm 1; \\ (1/e)^N & \text{near } x = 0. \end{cases}$$

whereas

$$|p_{Cheb}(z)| \simeq e^{(N+1)\phi_{equi}(z)} = 2^{-N}, \qquad z \in [-1, 1].$$

We see that p_{equi} can take on very different values well inside the interval vs. near the edges. On the other hand the values of p_{Cheb} are near-constant in $[-1, 1]$. The density $\rho_{Cheb}(x)$ is the only one that will give rise to this behavior, so there is something special about it. It is the difference in asymptotic behavior of $(2/e)^{-N}$ vs. 2^{-N} that makes the whole difference for interpolation, as $N \to \infty$.

One may argue that neither p_{equi} nor p_{Cheb} blow up as $N \to \infty$, but it is an interesting exercise to show that if we were interpolating in an interval $[-a, a]$ instead of $[-1, 1]$, then the bounds would be multiplied by a^N, by homogeneity.

The Chebyshev points are not the only one that correspond to the density $\rho_{Cheb}(x)$ as $N \to \infty$. For instance, there is also the Chebyshev roots

$$\theta'_j = \frac{\pi}{2N} + \frac{\pi j}{N}, \qquad j = 0, \ldots, 2N - 1,$$

which are the *roots* of $T_N(x)$, instead of being the extremal points. They give rise to very good interpolation properties as well.

Finally, let us mention that the theory can be pushed further, and that the exponential rate of convergence of Chebyshev interpolation for analytic functions can be linked to the maximum value of $\phi(z)$ on the strip in which the extension $f(z)$ of $f(x)$ is analytic. We will not pursue this further.

7.2 Differentiation

7.2.1 Bandlimited differentiation

The idea that a numerical approximation of a derivative can be obtained from differentiating an interpolant can be pushed further. In this section we return to bandlimited interpolants. We've seen that they are extremely accurate when the function is smooth and periodic; so is the resulting differentiation scheme. It is called bandlimited differentiation, or spectral differentiation.

First consider the case of $x \in \mathbb{R}$ and $x_j = jh$, $j \in \mathbb{Z}$. As we've seen, the bandlimited/spectral interpolant of $u(x_j)$ is

$$p(x) = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{ikx} \hat{u}(k)\, dk,$$

where $\hat{v}(k)$ is the SFT of $v_j = u(x_j)$. Differentiating $p(x)$ reduces to a multiplication by ik in the Fourier domain. Evaluating $p'(x_j)$ is then just a matter of letting $x = x_j$ in the resulting formula. The sequence of steps for bandlimited differentiation ($x \in \mathbb{R}$) is the following:

- Obtain the SFT $\hat{v}(k)$ of $v_j = u(x_j)$;
- Multiply $\hat{v}(k)$ by ik;
- Obtain the ISFT of $\hat{w}(k) = ik\hat{v}(k)$, call it w_j.

The numbers w_j obtained above are an approximation of $u'(x_j)$. The following result makes this precise.

Theorem 17. *(Accuracy of bandlimited differentiation, see also Theorem 4 in Trefethen's book) Let u have p derivatives in $L^1(\mathbb{R})$. Let $v_j = u(x_j)$, and $w_j = p'(x_j)$ be the result of bandlimited differentiation. Then*

$$\sup_j |w_j - u'(x_j)| = O(h^{p-2}).$$

and

$$\|u' - p'\|_2 = O(h^{p-3/2}).$$

Proof. The proof hinges on the fact that

$$|\hat{v}(k) - \hat{u}(k)| = O(h^p).$$

One power of h is lost when differentiating u (because ik is on the order of $1/h$ over the fundamental cell $[-\pi/h, \pi/h]$). Half a power of h is lost in going back to the physical domain (j instead of k) via the L^2 norm (why?), and a full power of h is lost when going back to j in the uniform sense (why?). $\square$

The point of the above theorem is that the order of bandlimited differentiation is directly linked to the smoothness of the function, and can be arbitrarily large. This is called spectral accuracy. One can even push the analysis further and show that, when f is real-analytic, then the rate of convergence of w_j towards $u'(x_j)$ is in fact exponential/geometric.

Of course in practice we never deal with a function $u(x)$ defined on the real line. In order to formulate an algorithm, and not simply a sequence of abstract steps, we need to limit the interval over which u is considered. Spectral differentiation in the periodic interval $[0, 2\pi]$ works like before, except DFT are substituted for SFT. For $\theta \in [0, 2\pi]$, we've seen that the spectral interpolant is defined as

$$p(\theta) = \frac{1}{2\pi} {\sum}''^{\,N/2}_{k=-N/2} e^{ik\theta} \hat{f}_k.$$

(The double prime is important here.) Again, a derivative can be implemented by multiplying by ik in the Fourier domain. The sequence of steps is very similar to what it was before, except that we can now label them as "compute", and not just "obtain":

- Compute the DFT $\hat{v}_k$ of $v_j = u(x_j)$;
- Multiply $\hat{v}_k$ by ik;
- Compute the IDFT of $\hat{w}_k = ik\hat{v}_k$, call it w_j.

The result of accuracy are the same as before, with the provision that u needs to be not only smooth, but also smooth when extended by periodicity.

The FFT can be used to yield a fast $O(N \log N)$ algorithm for spectral differentiation.

Note that higher derivatives are obtained in the obvious manner, by multiplying in Fourier by the adequate power of ik.

7.2.2 Chebyshev differentiation

In view of what has been covered so far, the idea of Chebyshev differentiation is natural: it is simply differentiation of the Chebyshev interpolant at the Chebyshev nodes. It proves very useful for those smooth functions on an interval, which do not necessarily extend smoothly by periodicity.

Let us recall that the Chebyshev interpolant is $q(x) = p(\arccos x)$, where $p(\theta)$ is the bandlimited interpolant of $u(\cos\theta_j)$ at the Chebyshev points $x_j = \cos\theta_j$. As such, a differentiation on $q(x)$ is not the same thing as a differentiation on $p(\theta)$. Instead, by the chain rule,

$$q'(x) = \frac{-1}{\sqrt{1-x^2}} p'(\arccos x).$$

The algorithm is as follows. Start from the knowledge of $u(x_j)$ at $x_j = \cos\theta_j$, $\theta_j = jh$, $h = \pi/N$, and $j = 1, \ldots, N$.

- Perform an even extension of $u(x_j)$ to obtain $u(\cos\theta_j)$ for $\theta_j = jh$, $h = \pi/N$, and $j = -N+1, \ldots, N$. Now we have all the equispaced sample of the periodic function $u(\cos\theta)$ for θ_j covering $[0, 2\pi]$, not just $[0, \pi]$.
- Take the DFT of those samples, call it $\hat{v}_k$,
- Multiply by ik,
- Take the IDFT of the result $ik\hat{v}_k$,
- Multiply by $-1/\sqrt{1-x_j^2}$ to honor the chain rule. At the endpoints $x_j = -1$ or 1, take a special limit to obtain the proper values of $p'(-1)$ and $p(1)$. See Trefethen's book for the correct values.

This is a fast algorithm since we can use the FFT for the DFT and IDFT.

Since we are only a change of variables away from Fourier analysis in a periodic domain, the accuracy of Chebyshev differentiation is directly inherited from that of bandlimited differentiation. We also have spectral accuracy.

Note that higher derivatives can be treated similarly, by applying the chain rule repeatedly.

In practice, Chebyshev methods are particularly useful for boundary-value problems (we'll come back to this), when all samples of a function are to be determined at once, and when we have the freedom of choosing the sample points.

7.3 Integration

7.3.1 Spectral integration

To go beyond methods of finite order, the idea of spectral integration is to integrate a bandlimited interpolant. This strategy yields very high accuracy when the function is smooth and periodic.

Consider a function $u(\theta)$, $\theta \in [0, 2\pi]$. Its samples are $v_j = u(\theta_j)$ with $\theta_j = jh$, $h = 2\pi/N$, and $j = 1, \ldots, N$. Form the DFT:

$$\hat{v}_k = h \sum_{j=1}^{N} e^{-ik\theta_j} v_j,$$

and the bandlimited interpolant,

$$p(\theta) = \frac{1}{2\pi} {\sum}''^{N/2}_{k=-N/2} e^{ik\theta} \hat{v}_k.$$

Integrating $p(\theta)$ gives the remarkably simple following result.

$$\int_0^{2\pi} p(\theta)\, d\theta = \hat{v}_0 = h \sum_{j=1}^{N} u(\theta_j).$$

We are back to the trapezoidal rule! (The endpoints are identified, $\theta_0 = \theta_N$.) While we have already encountered this quadrature rule earlier, it is now derived from Fourier analysis. So the plot thickens concerning its accuracy properties.

Specifically, we have to compare $\hat{v}_0$ to $\hat{u}_0 = \int_0^{2\pi} u(\theta)\, d\theta$, where $\hat{u}_k$ are the Fourier series coefficients of $u(\theta)$. The relationship between $\hat{v}_0$ and $\hat{u}_0$ is the Poisson summation formula (7.2):

$$\hat{v}_0 = \sum_{m \in \mathbb{Z}} \hat{u}_{mN}, \qquad N = \frac{2\pi}{h}.$$

The most important term in this sum is $\hat{u}_0$ for $m = 0$, and our task is again to control the other ones, for $m \neq 0$. The resulting accuracy estimate is the following.

Theorem 18. *Assume u has p derivatives in $L^1[0, 2\pi]$, where $[0, 2\pi]$ is considered a periodic interval. (So it matters that the function connects smoothly by periodicity.) Then*

$$\int_0^{2\pi} u(\theta)\, d\theta - h \sum_{j=1}^{N} u(\theta_j) = O(h^p).$$

Proof. By the Poisson summation formula, in the notations of the preceding few paragraphs,

$$\hat{u}_0 - \hat{v}_0 = \sum_{m \neq 0} \hat{u}_{mN}.$$

We have already seen that the smoothness properties of u are such that

$$|\hat{u}_k| \leq C|k|^{-p}.$$

So we have

$$|\hat{u}_0 - \hat{v}_0| \leq C \sum_{m \neq 0} (mN)^{-p} \leq C'\, N^{-p},$$

which is a $O(h^p)$. $\square$

As a conclusion, the trapezoidal rule is spectrally accurate (error $O(h^{p+1})$ for all $p \geq 0$ when $u \in C^\infty$), *provided* the function to be integrated in smooth *and* periodic. If the function is C^∞ in an interval but is for instance discontinuous upon periodization, then we revert to the usual $O(h^2)$ rate. So the true reason for the trapezoidal rule generally being $O(h^2)$ and not $O(h^\infty)$ is only the presence of the boundaries!

An important example of such periodic smooth function, is a regular C^∞function multiplied by a C^∞ window that goes smoothly to zero at the endpoints of $[a, b]$, like for instance a member of a partition of unity. (This plays an important role in some electromagnetism solvers, for computing the radar cross-section of scatterers.)

7.3.2 Chebyshev integration

Like for Chebyshev differentiation, we can warp θ into x by the formula $\theta = \arccos x$, and treat smooth functions that are not necessarily periodic. Integrating a Chebyshev interpolant gives rise to Chebyshev integration, also called Clenshaw-Curtis quadrature.

Assume that $f(x)$ is given for $x \in [-1, 1]$, otherwise rescale the x variable. The Chebyshev interpolant is built from the knowledge of f at the Chebyshev nodes $x_j = \cos\theta_j$, and takes the form

$$p(x) = \sum_{n \geq 0} a_n T_n(x).$$

We have seen that a_n are obtained by even extension of $f(x_j)$, followed by an FFT where the sine terms are dropped. As a result,

$$\int_{-1}^{1} p(x)\, dx = \sum_{n \geq 0} a_n \int_{-1}^{1} T_n(x)\, dx.$$

We compute

$$\int_{-1}^{1} T_n(x)\, dx = \int_{0}^{\pi} \cos(n\theta) \sin\theta d\theta.$$

This integral can be evaluated easily by relating cos and sin to complex exponentials (or see Trefethen's book for an elegant shortcut via complex functions), and the result is

$$\int_{-1}^{1} T_n(x)\, dx = \begin{cases} 0 & \text{if } n \text{ is odd;} \\ \frac{2}{1-n^2} & \text{if } n \text{ is even.} \end{cases}$$

The algorithm for Chebyshev differentiation is simply: (1) find the coefficients a_n for $0 \leq n \leq N$, and (2) form the weighted sum

$$\sum_{n \text{ even}, n \leq N} a_n \frac{2}{1-n^2}.$$

The accuracy estimate of Chebyshev integration is the same as that of bandlimited integration, except we do not need periodicity. So the method is spectrally accurate.

An important method related to Chebyshev quadrature is Gaussian quadrature, which is similar but somewhat different. It is also spectrally accurate for smooth functions. Instead of using extrema or zeros of Chebyshev polynomials, it uses zeros of Legendre polynomials. It is usually derived directly from properties of orthogonal polynomials and their zeros, rather than Fourier analysis. This topic is fascinating but would take us a little too far.

Appendix A

Round-off errors

In double-precision floating-point arithmetic (the most widespread choice in scientific computation), real numbers are represented using 8 bytes, or 64 bits (binary digits).

The binary expansion of a number x is of the form

$$x = a \cdot 2^b,$$

where a is called the mantissa (or significand) and b is called the exponent. A normalized number is one for which the mantissa is of the form $1.a_1a_2a_3\ldots$, i.e., the first bit is one.

The convention is to use 52 bits to represent the bits of the mantissa, i.e., we encode the a_j for $j = 1, \ldots, 52$, and to use 11 bits to encode the exponent. This results in the ability to use integers exponents between the values of -1023 and 1023 (with 1024 set aside to represent the exceptions ∞ and NaN). The last bit is used to encode the sign of x.

With this scheme, the range of accessible machine numbers is a collection with spacing dependent on the size of the nearby numbers. For instance,

- Between 1 and 2, the exponent is $b = 0$ and the spacing between machine-representable numbers is 2^{-52}. The smallest numbers in that interval are $1, 1 + 2^{-52}, 1 + 2 \cdot 2^{-52}$, etc.

- Between $1/2$ and 1, the exponent is $b = -1$ and the spacing between machine-representable numbers is 2^{-53}. The largest numbers in that interval are $1, 1 - 2^{-53}, 1 - 2 \cdot 2^{-53}$, etc.

We see that, in the neighborhood of 1, the maximum error that one may incur upon rounding is 2^{-53}. This very special number is called the epsilon machine, and its value is approximately 10^{-16}. If not in the neighborhood of 1, it is the maximum *relative* round-off error incurred in the double precision system.

Example 1: Both $1 = 1.0\ldots00 \cdot 2^0$ and $2^{-52} = 1.0\ldots00 \cdot 2^{-52}$ are exactly representable. Upon taking the sum of these two numbers, 2^{-52} is denormalized to match the exponent of 1, hence is temporarily represented as $0.0\ldots01 \cdot 2^0$. The addition is done without error and results in $1.0\ldots01 \cdot 2^0$.

Example 2: A round-off errors is incurred when trying to sum 1 and 2^{-53}. Even though the latter number is machine-representable, its denormalization to $b = 0$ results in a loss of information, since the bit equal to 1 gets "pushed out" of the 52-bit mantissa. The number 2^{-53} is confounded with zero in denormalized form, hence $1 + 2^{-53}$ is computed to be 1.

Having 64 bits of precision looks like it is more than enough for representing any given real number in the computer, and that is often the case, but we should be mindful that the propagation of round-off errors plagues some difficult computations. Often, a loss of precision occurs when subtracting nearby numbers; if x_1 and x_2 agree to 6 binary places, say, then the computation of their difference will incur a loss of 6 bits in the mantissa (why?). This may not sound like a big deal, but if this kind of operation is repeated 9 times or more in a sequence, then all the bits of the mantissa end up being lost.