

# Mat1062: Introductory Numerical Methods for PDE

Mary Pugh

March 26, 2009

## 1 Ownership

These notes are the joint property of Rob Almgren and Mary Pugh.

## 2 Spectral Methods

We have now argued that we may look at a particular function either in the original form  $u(x)$ , or in the Fourier representation  $\hat{u}_k$ . What can we do with the latter that we couldn't do with the former?

For PDEs, we often need to differentiate. If  $u$  is periodic and  $v(x) = u'(x)$ , then

$$\begin{aligned}\hat{v}_k &= \frac{1}{L} \int_0^L e^{-ik2\pi x/L} v(x) dx = \frac{1}{L} \int_0^L e^{-ik2\pi x/L} u'(x) dx \\ &= -\frac{1}{L} \int_0^L \left( e^{-ik2\pi x/L} \right)' u(x) dx + \frac{1}{L} u(x) e^{-ik2\pi x/L} \Big|_{x=0}^{x=L} \\ &= \frac{ik2\pi}{L} \frac{1}{L} \int_0^L e^{-ik2\pi x/L} u(x) dx = \frac{ik2\pi}{L} \hat{u}_k\end{aligned}$$

Equivalently, we can differentiate the inversion formula (we can take the derivative inside the series if the  $\hat{u}_k$  decay fast enough) to write

$$u'(x) = \frac{d}{dx} \sum_{k=-\infty}^{\infty} \hat{u}_k e^{ik2\pi x/L} = \sum_{k=-\infty}^{\infty} \frac{ik2\pi}{L} \hat{u}_k e^{ik2\pi x/L}.$$

The conclusion is the same: the Fourier series of the derivative is  $ik2\pi/L$  times the Fourier series of the function. This is straightforward compared

to differentiating the function in physical space via limits

$$u'(x) = \lim_{h \rightarrow 0} \frac{u(x+h) - u(x)}{h}$$

Differentiation of the Fourier representation is a “diagonal” operation that acts on one mode at a time. (We say it’s a diagonal operation because if  $\hat{v}$  is the infinite vector of Fourier coefficients then  $\hat{v} = \Lambda \hat{u}$  for a diagonal matrix  $\Lambda$ .)

Similarly, convolution of two functions, an integral operator in physical space, is a mode-by-mode multiplication in Fourier space. And certain other integral operators (the Hilbert transform, for example) are diagonal operations in the Fourier representation.

However, *nonlinear* operations such as multiplication in physical space generally correspond to infinite sums in Fourier space. In this way, some operations are easier in physical space and others are easier in Fourier space.

## Discrete Fourier Transform

Suppose  $u_0, \dots, u_{n-1}$  are an array of  $n$  real or complex numbers. The forward *discrete Fourier transform* (FFT) is the transformation

$$\hat{u}_k^d = \frac{1}{n} \sum_{j=0}^{n-1} u_j e^{-ik2\pi j/n}, \quad k = 0, \dots, n-1. \quad (1)$$

As with the continuous transform, there are several conventions for choice of the prefactor. You may easily verify that the inverse transformation is

$$u_j = \sum_{k=0}^{n-1} \hat{u}_k^d e^{ik2\pi j/n} \quad j = 0, \dots, n-1 \quad (2)$$

using the orthogonality relation

$$\frac{1}{n} \sum_{k=0}^{n-1} e^{i\ell 2\pi k/n} e^{-ij 2\pi k/n} = \delta_{j\ell}. \quad (3)$$

To see why this orthogonality relation holds, first note that if  $\ell = j$  then it’s clearly true. If  $\ell \neq j$  and  $0 \leq \ell, j \leq n-1$  then we know that  $1 - n \leq \ell - j \leq n-1$  and so

$$e^{i2\pi(\ell-j)/n} \neq 1 \implies \frac{1}{n} \sum_{k=0}^{n-1} \left( e^{i2\pi(\ell-j)/n} \right)^k = \frac{1}{n} \frac{1 - \left( e^{i2\pi(\ell-j)/n} \right)^n}{1 - e^{i2\pi(\ell-j)/n}} = 0$$

The discrete Fourier transform and inverse are both linear operations that take  $n$  numbers to  $n$  numbers. They can be represented by matrices: (1) corresponds to  $\hat{u}^d = Qu$  for some matrix  $Q$  and (2) corresponds to  $u = \tilde{Q}\hat{u}^d$ . You can check that (3) implies that  $\tilde{Q}Q = I$ . If we had chosen different prefactors in the definitions of  $Q$  and  $\tilde{Q}$  and had taken them to be  $1/\sqrt{n}$  and  $1/\sqrt{n}$  instead of  $1/n$  and  $1$ , respectively, then we would have  $\tilde{Q} = Q^*$  and the discrete Fourier transform would correspond to multiplication by a unitary matrix.

Also, you can check that we have the conservation of discrete  $L^2$  norm

$$\frac{1}{n} \sum_{j=0}^{n-1} |u_j|^2 = \sum_{k=0}^{n-1} |\hat{u}_k^d|^2.$$

Naively, it takes  $\mathcal{O}(n)$  operations to compute  $\hat{u}_k^d$  using (1). And so to compute  $\hat{u}_0^d, \hat{u}_1^d, \dots, \hat{u}_{n-1}^d$  sequentially would take  $\mathcal{O}(n^2)$  operations. One of the most significant numerical algorithms of all time was the discovery that computing  $\hat{u}^d$  could be performed in time  $n \log n$  rather than  $n^2$ . This algorithm, discovered by Cooley & Tukey (*Mathematics of Computation*, 19(1965)297–301) is the *Fast Fourier Transform*. If it weren't for the FFT, spectral methods wouldn't be so popular. Their algorithm required that  $n = 2^s$  for some integer  $s$ . The FFT has now been generalised to other values of  $n$ . The fast fourier transform in matlab is based on the FFTW package ("the Fastest Fourier Transform in the West"). If you go to their webpage (<http://www.fftw.org>) you'll find, "Arbitrary-size transforms. (Sizes with small prime factors are best, but FFTW uses  $\mathcal{O}(n \log(n))$  algorithms even for prime sizes.)" I don't know the details behind this statement but it suggests that the constant  $C$  involved in the  $\mathcal{O}(n \log(n))$  is smaller for  $n$  that is made of small prime factors such as  $2^s 3^t 5^r$ . I find powers of 2 useful because they work well with convergence studies — dividing  $h$  by 2 corresponds to doubling  $n$ . That said, if you're doing a 3-d code then the work-increase in going from  $128^3$  intervals to  $256^3$  intervals might be prohibitive and so you might find yourself maxing out at  $173^3$  intervals, for example.

Note that the FFT is built for the discrete Fourier transform which is for trig functions. If you were working in a domain that had other eigenfunctions (Bessel functions in the radial direction and trig functions in the angular direction, say) then you would need a FBT for the radial stuff and a FFT for the angular stuff.

### Relation between discrete and continuous transforms

The interesting question is what happens if the original numbers  $u_j$  are samples of a smooth function  $u(x)$ : what is the relationship between the discrete Fourier coefficients,  $\hat{u}_k^d$ , as given by (1) and the continuous Fourier coefficients,  $\hat{u}_k^c$ , as given by (4)?

$$\hat{u}_k = \langle u, \phi_k \rangle = \frac{1}{L} \int_0^L u(x) e^{-ik2\pi x/L} dx, \quad \text{for } k = 0, \pm 1, \pm 2, \dots \quad (4)$$

The discrete Fourier transform has to do with a collection of numbers  $u_0, \dots, u_{n-1}$ . The continuous Fourier transform has to do with a function sampled at  $n$  points:  $x_j = jL/n$ . To relate the two, we assume that  $u_j = u(x_j)$ , where  $x_j = jL/n$  are equally spaced grid points. Then using the discrete Fourier transform (1) we readily calculate

$$\begin{aligned} \hat{u}_k^d &= \frac{1}{n} \sum_{j=0}^{n-1} u(jL/n) e^{-ik2\pi j/n} = \frac{1}{n} \sum_{j=0}^{n-1} \left( \sum_{\ell=-\infty}^{\infty} \hat{u}_\ell^c e^{i\ell 2\pi(jL/n)/L} \right) e^{-ik2\pi j/n} \\ &= \sum_{\ell=-\infty}^{\infty} \hat{u}_\ell^c \left( \frac{1}{n} \sum_{j=0}^{n-1} e^{i(\ell-k)2\pi j/n} \right). \end{aligned}$$

By (3), the sum in parentheses has the value 1 when  $\ell - k$  is *any integer multiple of  $n$*  and is zero otherwise. Thus  $\ell = k + mn$  and

$$\hat{u}_k^d = \sum_{m=-\infty}^{\infty} \hat{u}_{k+mn}^c.$$

And so, the discrete mode  $k$  collects all the energy from all the periodic images of mode  $k$  in the continuous spectrum.

This is the reason it is so important that we have a large enough value of  $n$  when choosing how to sample the function  $u(x)$ . The discrete Fourier transform (1) was defined for  $k = 0, 1, \dots, n-1$ . Consider  $k$  such that  $0 \leq k \leq \lfloor n/2 \rfloor$ . Then

$$\hat{u}_k^d = \hat{u}_k^c + \sum_{m=1}^{\infty} \hat{u}_{k-mn}^c + \sum_{m=1}^{\infty} \hat{u}_{k+mn}^c$$

and  $|k+mn| > \lfloor n/2 \rfloor$  for all  $m$ . That is, if  $0 \leq k \leq \lfloor n/2 \rfloor$  then  $\hat{u}_k^d$  equals  $\hat{u}_k^c$  plus energy from higher modes. If we have chosen  $n$  sufficiently large then

all of these contributions from higher modes will be at the level of round-off and so we are comfortable treating  $\hat{u}_k^d$  as the same thing as  $\hat{u}_k^c$ .

Similarly, if  $\lfloor n/2 \rfloor < k \leq n-1$  then

$$\hat{u}_k^d = \hat{u}_{k-n}^c + \sum_{m=2}^{\infty} \hat{u}_{k-mn}^c + \sum_{m=0}^{\infty} \hat{u}_{k+mn}^c$$

Here, we note that  $|k-n| \leq \lfloor n/2 \rfloor$  and that for the first series,  $|k-mn| > n$  and for the second series  $|k+mn| > \lfloor n/2 \rfloor$ . And so in this case  $\hat{u}_k^d$  equals  $\hat{u}_{k-n}^c$  plus energy from higher modes. Again, if we have chosen  $n$  sufficiently large then all of these contributions from higher modes will be at the level of round-off and so we are comfortable treating  $\hat{u}_k^d$  as the same thing as  $\hat{u}_{k-n}^c$ .

To sum up, the discrete Fourier transform (1) gives us

$$\hat{u}_0^d, \hat{u}_1^d, \dots, \hat{u}_{n-1}^d.$$

To interpret  $\hat{u}_k^d$  as a Fourier coefficients we find the  $n$ -shift that brings  $k$  closest to 0:

$$\hat{u}_k^d = \begin{cases} \hat{u}_k^c & 0 \leq k \leq \lfloor n/2 \rfloor \\ \hat{u}_{k-n}^c & \lfloor n/2 \rfloor < k \leq n-1 \end{cases}$$

if  $n$  has been taken large enough so that all the higher energy modes have Fourier coefficients at the level of round-off error.

However, if  $u(x)$  is not a smooth enough function, or  $n$  is not large enough, then modes for small values of  $k$  will contain energy from much higher modes. This is called *aliasing error*.

The highest wave number that can be represented on a grid of length  $n$  has  $k = n/2$ , or  $\xi = \pi n/L$ . The wavelength is  $2\pi/\xi = 2L/n = 2h$ ; these modes are  $\pm 1$  at the grid points. This highest frequency is called the Nyquist frequency. In Figure 1, we take  $n = 8$  and plot  $\cos(32\pi x/8)$  and  $\cos(52\pi x/8)$  and sample at 8 points  $x_j = j$ . Because  $5 > n/2 = 4$ ,  $\cos(52\pi x/8)$  will be indistinguishable from  $\cos((n-5)2\pi x/8) = \cos(32\pi x/8)$  at the grid points, as observed.

### 3 Computing Derivatives

Now, suppose that we want to compute approximate values  $v_j \approx u'(x_j)$  of the derivative at the grid points  $x_j$ , given only the values  $u_j$  of the function at the grid points. This is precisely the problem we considered in finite

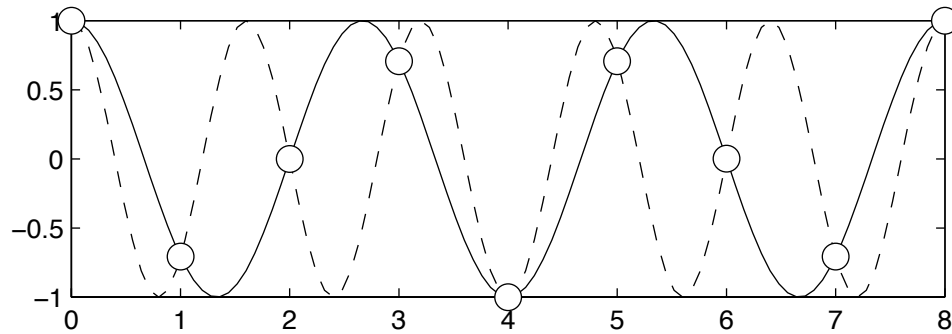


Figure 1: Aliasing error. The grid has  $n = 8$ , and the mode  $k_1 = 3$  is plotted with solid line and circles. The mode  $k_2 = 5 = n - k_1$  looks exactly the same on this grid. The highest unambiguous frequency is  $k = n/2 = 4$ .

differences, where we generally settled on the centered difference formula  $v_j = (u_{j+1} - u_{j-1})/2h$ . This expression has a second-order error  $\mathcal{O}(h^2)$ . In special cases, we had reasons to take a one-sided difference that was only first-order accurate. Conversely, if we wanted higher-order accuracy we could achieve it by using a larger stencil (more neighbors on either side).

Now we want to see how to do things using Fourier transformations. Given  $u_j$  for  $j = 0, j = 1, \dots, j = n - 1$  we know by the discrete Fourier inversion formula that

$$\begin{aligned} u_j &= \sum_{k=0}^{n-1} \hat{u}_k^d e^{ik2\pi j/n} \\ &= \sum_{k=0}^{\lfloor n/2 \rfloor} \hat{u}_k^d e^{ik2\pi j/n} + \sum_{k=\lfloor n/2 \rfloor + 1}^{n-1} \hat{u}_k^d e^{-i(n-k)2\pi j/n} \end{aligned}$$

And so, if  $u_j = u(x_j)$  then we have a trigonometric polynomial which interpolates the sample points:

$$\tilde{u}(x) := \sum_{k=0}^{\lfloor n/2 \rfloor} \hat{u}_k^d e^{ik2\pi x/L} + \sum_{k=\lfloor n/2 \rfloor + 1}^{n-1} \hat{u}_k^d e^{-i(n-k)2\pi x/L}.$$

We then differentiate this trigonometric polynomial

$$\tilde{u}'(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} ik \frac{2\pi}{L} \hat{u}_k^d e^{ik2\pi x/L} + \sum_{k=\lfloor n/2 \rfloor + 1}^{n-1} -i(n-k) \frac{2\pi}{L} \hat{u}_k^d e^{-i(n-k)2\pi x/L}.$$

and sample it at the points  $x_j = jh$  to create a vector  $v$ :

$$v_j := \tilde{u}'(x_j) \approx u'(x_j) = \sum_{k=0}^{\lfloor n/2 \rfloor} ik \frac{2\pi}{L} \hat{u}_k^d e^{ik2\pi j/n} + \sum_{k=\lfloor n/2 \rfloor + 1}^{n-1} -i(n-k) \frac{2\pi}{L} \hat{u}_k^d e^{ik2\pi j/n}.$$

That is,  $\{v_j\}$  is the discrete inverse transform of the vector

$$\hat{v}_k^d = \begin{cases} ik \frac{2\pi}{L} \hat{u}_k^d & 0 \leq k \leq \lfloor n/2 \rfloor \\ -i(n-k) \frac{2\pi}{L} \hat{u}_k^d & \lfloor n/2 \rfloor + 1 \leq k \leq n-1 \end{cases} \quad (5)$$

There is one tricky thing we need to keep track of. If  $n$  is even then the above approach will take a real-valued function  $u$  and return a complex-valued function. This is because to get a real-valued function we need complex conjugates to balance out. Specifically, if there is an  $e^{i2\pi x/L}$  we need an  $e^{-i2\pi x/L}$  and the coefficients multiplying them must be complex conjugates:

$$u(x) = \sum_{\ell=-\infty}^{\infty} \hat{u}_\ell^c e^{i\ell 2\pi x/L} \implies \bar{u}(x) = \sum_{\ell=-\infty}^{\infty} \overline{\hat{u}_\ell^c} e^{-i\ell 2\pi x/L} = \sum_{\ell=-\infty}^{\infty} \hat{u}_{-\ell}^c e^{i\ell 2\pi x/L}$$

and so  $u(x) = \bar{u}(x)$  if and only if  $\overline{\hat{u}_{-\ell}^c} = \hat{u}_\ell^c$  for all  $\ell$ . For the discrete Fourier transform, if  $n$  is even then  $\lfloor n/2 \rfloor = n/2$  and the above will have a term  $e^{in/2 2\pi x/L}$  with no corresponding complex conjugate term  $e^{-in/2 2\pi x/L}$ . For this reason, when implementing the spectral derivative, we take the discrete inverse transform of

$$\hat{v}_k^d = \begin{cases} ik \frac{2\pi}{L} \hat{u}_k^d & 0 \leq k < n/2 \\ 0 & k = n/2 \\ -i(n-k) \frac{2\pi}{L} \hat{u}_k^d & n/2 < k \leq n-1 \end{cases} \quad (6)$$

If  $n$  is odd then (6) is the same thing as (5). If  $n$  is even then (6) will handle the stray term in a way that ensures that the inverse discrete Fourier transform of  $\hat{v}^d$  is real-valued if  $u$  is real-valued. In principle, setting that coefficient to zero is introducing an error. However, if our computation is well-resolved then we have chosen  $n$  large enough so that  $\hat{u}_{n/2}^d$  is at the level of round-off and the lost information is also at the level of round-off.

Here is the algorithm:

1. Take the discrete Fourier transform of the given discrete data  $\{u_j\}$ . This is equivalent to interpolating the function by a trigonometric polynomial of degree  $n$  that passes exactly through the given points.
2. Given  $\{\hat{u}_k^d\}$ , compute the new coefficients  $\{\hat{v}_k^d\}$  via the rule (6).
3. Take the inverse discrete transform of  $\{\hat{v}_k^d\}$  to get values of the derivative at the grid points in physical space.

This is a *global* algorithm, since each output value  $v_j$  depends on each input value  $u_j$ . This is the logical limit of doing finite difference approximations of  $u'$  using larger and larger stencils. Its accuracy as a function of  $h$  is better than any power of  $h$ , as long as the grid is fine enough to resolve all features of the function  $u$ . If  $u$  is underresolved, it will give dramatically *bad* answers for  $u'$ , like any high-order method.

Here is an example where we seek

$$u'(0) \quad \text{for} \quad u(x) = \sin(4x)$$

The exact answer is  $u'(0) = 4$ . Below, we use a centered difference to approximate this derivative, as well as the spectral method. Note that the spectral method does poorly until  $n$  is large enough to resolve the function. And once it is large enough to resolve the function, it gets the derivative correct to the level of round-off error.

$n$	$h = 2\pi/n$	f.d. error	ratio	sp. error
2	3.1416e+00	4.0000e+00	1.0000e+00	4.0000e+00
4	1.5708e+00	4.0000e+00	1.0000e+00	4.0000e+00
8	7.8540e-01	4.0000e+00	2.7519e+00	4.0000e+00
16	3.9270e-01	1.4535e+00	3.6453e+00	4.4409e-16
32	1.9635e-01	3.9873e-01	3.9085e+00	3.5527e-15
64	9.8175e-02	1.0202e-01	3.9769e+00	3.5527e-15
128	4.9087e-02	2.5653e-02	3.9942e+00	1.7764e-15
256	2.4544e-02	6.4224e-03		8.4377e-15

In this example, it looks as if we've given a real edge to the spectral approach by testing the two approaches on one of the eigenfunctions of the spectral approach. In fact, because differentiation is a linear operation, if we're trying to find the derivative of  $u$  where  $u$  is any periodic function and we've sampled  $u$  at enough points so that  $\hat{u}_{n/2}^d \sim 10^{-16}$  (and so there is no

aliasing error) then the spectral accuracy of differentiation for trig functions will lead to spectral accuracy for the differentiation of  $u$ . As an example, consider

$$u(x) = e^{\sin(x)} \quad \text{on } [-\pi, \pi].$$

For this function,  $u'(0) = 1$ . We repeat the above experiment and find:

$n$	$h = 2\pi/n$	f.d. error	ratio	sp. error
2	3.1416e+00	1.0000e+00	3.9707	1.0000e+00
4	1.5708e+00	2.5184e-01	11.066	9.6339e-02
8	7.8540e-01	2.2759e-02	14.735	1.1563e-03
16	3.9270e-01	1.5446e-03	15.689	2.3023e-08
32	1.9635e-01	9.8453e-05	15.923	3.1086e-15
64	9.8175e-02	6.1832e-06	15.981	2.8866e-15
128	4.9087e-02	3.8691e-07	15.995	8.1046e-15
256	2.4544e-02	2.4189e-08		3.5527e-15

Here we note two things. First of all, the finite difference approximation is converging to the true solution faster than expected: the ratios are going to 16 rather than 4. I'll let you puzzle that over. Secondly, the spectral approximation of the derivative is somewhat better than the finite difference approximation of the derivative when  $n = 2, 4, 8,$  and  $16$  but it only becomes spectrally accurate (i.e. at the level of round-off error) once  $n = 32$ . If we look at the power spectrum, we find that it's only when  $n \geq 32$  that we have no aliasing error.

To summarize, the *good* things about spectral methods are

- Spectacular accuracy on smooth periodic problems, and
- Ability to handle some nonlocal integral effects.

The *bad* things are

- Terrible accuracy if the solution is not smooth, or cannot be smoothly extended to a periodic functions (if the boundary conditions are not homogeneous Neumann or Dirichlet).
- The grid must be uniform (equally spaced).
- It is very "finicky:" if the code isn't completely correct it will not do anything reasonable, and it can be hard to debug.