Mat1062: Computational Methods for PDE

Mary Pugh

March 27, 2008

1 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 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 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

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

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

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

We then differentiate this trigonometric polynomial

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

and sample it at the points $x_j = jh$:

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

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

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

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. 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. 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

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

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 $\hat{u}_{n/2}^{\mathrm{d}}$ is 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 trigometric polynomial of degree n that passes exactly through the given points.

2. Given $\{\hat{u}_k^{\mathrm{d}}\}$, compute the new coefficients $\{\hat{v}_k^{\mathrm{d}}\}$ via the rule (1).

3. Take the inverse discrete transform of $\{\hat{v}_k^{\mathrm{d}}\}$ 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 taking derivatives using more and more neighbors. 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 solution. If the solution is underresolved, it will give dramatically *bad* answers, like any high-order method.

Here is an example where we seek

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

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 — we're testing the two approaches on one of the eigenfunctions. In fact, because differentiation is a linear operation, if we're trying to find the derivative of u and we've sampled u at enough points so that $\hat{u}^{\mathrm{d}}_{n/2} \sim 10^{-16}$ (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

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

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 required: 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 will be hard to see what is wrong.