410 Tutorial 10: Introduction to Spectral Methods

November 2017

At the beginning of the course we looked at Lagrange polynomials for interpolating data on a grid. For $n+1$ points, the Lagrange interpolating polynomial was given by,

$$L(x) = \sum_{i=0}^{n} l_i(x) y_i \tag{1}$$

$$l_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} y_i \tag{2}$$

or, in Barycentric form as,

$$l(x) = \prod_{i=0}^{n} (x - x_i) \tag{3}$$

$$l'(x_j) = \prod_{i=0, i \neq j}^{n} (x_j - x_i) \tag{4}$$

$$w_j = \frac{1}{l'(x_j)} \tag{5}$$

$$l_j(x) = l(x) \frac{w_j}{x - x_j} \tag{6}$$

allowing us to interpolate smoothly between data points. The generalization to taking the derivative is obvious but messy; simply differentiate the Lagrange polynomial. Since this ends up being a bit of a mess in the above notation, for our purposes it will be better to think of the Lagrange polynomial as being the solution to the following system of equations,

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^2 p(x_i) = y_i \tag{7}$$

such that our a_n coefficients are given as the solution to the following matrix equation:

$$\begin{bmatrix} x_0^n & x_0^{n-1} & x_0^{n-2} & \dots & 1 \\ x_1^n & x_1^{n-1} & x_1^{n-2} & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n^n & x_n^{n-1} & x_n^{n-2} & \dots & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

then, the derivatives can be computed by simply differentiating the above polynomial. But, why would we be interested in doing this in the first place? We have already discussed robust methods for approximating derivatives on a finite grid, namely finite difference methods of various orders. The main thing to keep in mind about finite difference methods is that they are by definition local methods. In particular, the standard finite difference stencils can be though of as performing the above polynomial fit and differentiation on a small subset of the data points.

While this often works, there is much to be said for using global, or spectral, methods for computing derivatives and solving systems of equations. In particular, provided the functions we wish to approximate are sufficiently smooth and obey appropriate boundary conditions, one can often get comparable accuracy using ≈ 30 grid points in a spectral method as with using thousands or tens of thousands with a finite difference technique.

For the remainder of this tutorial, we will consider $f_1 = x$, $f_2 = \sin(\pi x)$ and the Runge function, $f_3 = \frac{1}{1+25x^2}$ with all functions defined on the domain $x\epsilon[-1, 1]$.

1 Lagrange Polynomials

1. For $n = 5, 10, 20, 30, 40$ compute the Lagrange interpolating polynomial for each of the functions given above and compare with the analytic functions.
2. Plot the errors between the interpolations and the true functions as a function of n
3. For $n = 5, 10, 20, 30, 40$ compute the derivative of the Lagrange interpolating polynomial for each of the functions given above and compare with the analytic derivatives.
4. Are all of the functions acting as expected?

The behaviour you are seeing with f_3 is referred to as Runge's phenomenon. It is similar to the Gibbs phenomenon of Fourier series and is characteristic of two properties of the function in question.

1. The magnitude of the derivatives grow quickly as the order of the interpolating polynomial is increased.
2. The equidistance between points on the grid leads to a Lebesgue constant that increases quickly with the degree of the polynomial.

Obviously then, if Lagrange Polynomials are unsuitable for approximating and taking derivatives of arbitrary functions, they will be a poor choice upon which to base our spectral methods.

The orthogonality of the Lagrange polynomials is given as follows:

$$\sum_{n=0}^{N-1} l_i(x_n) l_j(x_n) = \delta^i_{j'} \tag{8}$$

$$\tag{9}$$

In practice, the coefficients of transformations such as the Lagrange, Fourier and Chebyshev transforms are computed via the orthogonality condition rather than through matrix inversion.

2 Discrete Fourier Transforms

When one thinks of spectral methods, Fourier transforms are probably the first thing that comes to mind due to their host of useful properties for periodic functions. For a function defined at a discrete number of points x_i, the DFT (Discrete Fourier Transform) and inverse DFT are given by (one out of many definitions),

$$Y_k = \frac{1}{N} \sum_{n=0}^{N-1} y_n \exp^{\frac{-2\pi i}{N} nk} \tag{10}$$

$$y_n = \sum_{k=0}^{N-1} Y_k \exp^{\frac{2\pi i}{L} k \Delta x} \tag{11}$$

$$\tag{12}$$

with orthogonality given by:

$$\sum_{n=0}^{N-1} \exp^{(\frac{2\pi i}{N} kn)} \exp^{(\frac{2\pi i}{N} k' n)} = N \delta^k_{k'} \tag{13}$$

$$\tag{14}$$

1. For $n = 5, 10, 20, 30, 40$ compute the DFT for each of the functions given above. Use the inverse transform to interpolate to higher resolutions and compare with the analytic functions.

2. Plot the errors between the interpolations and the true functions as a function of n

3. For $n = 5, 10, 20, 30, 40$ compute the derivative of the inverse DFT for each of the functions given above and compare with the analytic derivatives.

4. Are all of the functions acting as expected?

2.1 Aliasing

What we are running into in the above, is known as aliasing. When looking at a discrete number of points, the DFT is invariant under frequency shifts since,

$$\exp^{\frac{-2\pi i}{L} k x_j} = \exp^{\frac{-2\pi i}{L}(k+mN) x_j} \tag{15}$$

with m an integer. In other words, our going by our definition above, we are using oscillatory functions above the Nyquist frequency of our data! We can ameliorate this by using negative frequencies and shifting the resulting inverse transform:

$$Y_k = \frac{1}{N} \sum_{n=-N/2+1}^{N/2} y_n \exp^{\frac{-2\pi i}{N} nk} \tag{16}$$

$$y_{n+N/2-1} = \sum_{k=0}^{N-1} Y_k \exp^{\frac{2\pi i}{L} kx} \tag{17}$$

$$\tag{18}$$

1. For $n = 5, 10, 20, 30, 40$ compute the DFT for each of the functions given above. Use the inverse transform to interpolate to higher resolutions and compare with the analytic functions.
2. Plot the errors between the interpolations and the true functions as a function of n
3. For $n = 5, 10, 20, 30, 40$ compute the derivative of the inverse DFT for each of the functions given above and compare with the analytic derivatives.
4. Are all of the functions acting as expected?

We see now that this version of the Fourier transform works well for smooth periodic data. However near the edges of the domain of non-periodic functions there are significant Gibb's oscillations. This is due to the fact that we are attempting to impose smoothness and periodicy on data which is intrinsically non-periodic. As such, we should reserve Fourier and related transforms for data which is periodic[1] (or at the very least asymptotes to the same value at the edges of the domain).

3 Chebyshev Polynomials

When we used the Lagrange interpolating polynomials we ran into the issue of Runge's phenomenon primarily as a result of the uniform periodic spacing of our

[1] Some people get around this by adding a smooth interpolant to the end of their domain, but this is cumbersome and not very elegant if fairly effective

grid. With the Fourier type transforms, we were unable to adequately represent non-periodic data. Chebyshev polynomials fix this by spacing grid points with the following density:

$$\rho \approx \frac{N}{\pi\sqrt{1-x^2}} \tag{19}$$

The simplest set of points satisfying this relation are the Chebyshev points

$$x_j = cos(j\pi/N) \tag{20}$$

For the case of a uniformly spaced grid, our basis functions turned out to be the Lagrange polynomials. With the spacing above, our basis functions are the Chebyshev polynomials,

$$T_n(x) = \cos(n \arccos(x)) \tag{21}$$

which obey the following orthogonality relationship (analogous to those of the Fourier transform or Larange polynomials)

$$\int_{-1}^{1} T_0(x)T_0(x)\frac{dx}{1-x^2} = \pi \tag{22}$$

$$\int_{-1}^{1} T_m(x)T_n(x)\frac{dx}{1-x^2} = \delta_m^n \frac{\pi}{2} \tag{23}$$

or on a discrete grid:

$$\sum_{k=0}^{N-1} T_0(x_k)T_0(x_k) = N \tag{24}$$

$$\sum_{k=0}^{N-1} T_i(x_k)T_j(x_k) = \frac{N}{2}\delta_j^i \tag{25}$$

$$\tag{26}$$

1. For $n = 5, 10, 20, 30, 40$ compute the Chebyshev polynomial for each of the functions given above and compare with the analytic functions.

2. Plot the errors between the interpolations and the true functions as a function of n

3. For $n = 5, 10, 20, 30, 40$ compute the derivative of the Chebyshev for each of the functions given above and compare with the analytic derivatives.

4. Are all of the functions acting as expected?