
The topics here will be in the book ‘Spectral methods in Matlab’ (Chapter 3); Another reference book is ‘Spectral methods in chemistry and physics.’

1 Spectral methods v.s. pseudo-spectral methods

1.1 Spectral method

A spectral method is to represent the solution of a differential equation in terms of a basis of some vector space and then reduce the differential equations to an ODE system for the coefficients. In this sense, the (continuous) Fourier Transform and the Finite Element methods are all spectral methods.

In particular, suppose we have some equation

\[ T(u) = f(x) \]

where \( T \) is some temporal-spatial operator for the differential equation. Then, under the basis we have the error:

\[ R(x) = T\left( \sum_i c_i(t)\phi_i(x) \right) - \sum_i \tilde{f}_i \phi_i \]

Pick test functions \( \{\chi_j\} \), which is the dual basis of \( \{\phi_i(x)\} \) (if the space is a Hilbert space like in FEM, these two are identical) and we require

\[ \langle \chi_j, R \rangle = 0. \]

This then gives a system of equations for \( c_j \).

- Advantages: Reduce differential operators to multiplications, and solving ODEs might be easier.
- Disadvantage: The multiplication in the spatial variable becomes convolution. Then, we may have a big matrix-vector multiplication.

1.2 Pseudo-spectral method

Here, represent the solution in terms of the basis but impose the equation only at discrete points. As we can see, the pseudo-spectral method in some sense is a spectral method in discrete space. There are some typical pseudo-spectral methods
• Fourier Pseudo-spectral method with trigonometric basis
• Chebyshev pseudo-spectral method with Chebyshev polynomial basis
• Collocation methods with polynomial basis (especially for ODEs solvers);
  (Some people also call the pseudo-spectral method the collocation
  method.)

For discrete points, we often choose \( \chi_j = \delta(x - x_j) \), so that we require
\[ R(x_j) = 0. \]

The pseudo-spectral method may have efficient algorithms, which makes
solving PDEs fast. For example, to compute \( a(x)u_x(x,t) \), using spectral
method where \( x \) is continuous, we have an infinite convolution. Even if we
truncate, we have a matrix multiplied with a vector. However, if we use the
Fourier pseudo-spectral method, we can make use of the FFT and IFFT to
change the function back into \( x \) space and perform the multiplication in \( x \)
space. This will save a lot of time.

In this course, we will only cover Fourier pseudo-spectral method. You
can read the Chebyshev yourself, which is also useful.

2 Basic theories for Discrete Fourier Transform (DFT)

Consider \( N \) (an even number) points \( v_1, v_2, \ldots, v_N \). We define the DFT
as
\[ \hat{v}_k = \sum_{n=1}^{N} e^{-ikx_n}v_n, \quad k \in \mathbb{Z} \]

where \( x_n = \frac{2\pi}{N} n \). This means that we regard \( v \) as a function on \((0, 2\pi]\) and
\( x_n \) is a sample point on the interval.

• Note that the definition here is a little different from the book. We
  use this because it’s more common. Also in Matlab, the summation is
  \( \sum_{n=1}^{N} e^{-ikx_{n-1}}v_n \), which will not cause a trouble, since we can imagine
  that we are doing a shifted sequence or \( v_n \) is associated with \( x_{n-1} \)
  instead of \( x_n \).

• Other people also use the definition as \( \sum_{n=0}^{N-1} e^{-ikx_n}v_n \). Of course, this
  is equivalent to ours if we regard \( x_N \) as \( x_0 \).
Different definitions (shifting the sequence) may result in a uniform factor phase for the DFT. However, as long as we have consistent inverse formula, it is just fine.

Properties

- \( \hat{v}_k = \hat{v}_{k+mN} \). In Matlab, the output is for \( k = 0, 1, 2, \ldots, N - 1 \) or \( k = 0, 1, 2, \ldots, N/2, -N/2 + 1, \ldots, -1 \). In other words, \( \text{fft}(v) \) gives a vector \( w \) such that \( w(1) = \hat{v}_0, \ldots, w(N) = \hat{v}_{N-1} \).
- If \( v \) is real valued, then \( \hat{v}_k = \overline{\hat{v}}_{-k} \), where the bar means complex conjugate. It follows that \( \hat{v}_{N/2} \) is real.
- Define the discrete convolution:

\[
    w_n = (u * v)_n = \sum_{j=1}^{N} u_j v_{n-j}
\]

where \( v_{n-j} = v_{N+n-j} \) if \( n - j < 0 \). Then,

\[
    \hat{w}_k = \hat{u}_k \hat{v}_k.
\]

- The discrete Parseval equality holds:

\[
    \sum_n u_n \overline{v}_n = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} \hat{u}_k \overline{\hat{v}}_k
\]

Let us prove the last two.

Proof. For the discrete convolution, we have

\[
    \hat{w}_k = \sum_{n=1}^{N} e^{-ikx_n} \sum_{j=1}^{N} u_j v_{n-j} = \sum_{j=1}^{N} u_j e^{-ikx_j} \sum_{n=1}^{N} e^{-ik(x_n-x_j)} v_{n-j}
\]

By the cyclic symmetry, we have the result.

We have

\[
    \sum_{k=-N/2+1}^{N/2} \hat{u}_k \overline{\hat{v}}_k = \sum_k (\sum_m u_m e^{-ikx_m})(\sum_n \overline{v}_n e^{ikx_n}) = \sum_m u_m \overline{v}_n \sum_k e^{ik(x_n-x_m)}
\]

The sum on the right hand side is \( N \delta_{m,n} \), so we have \( N \sum_n u_n \overline{v}_n \). \( \square \)
Remark 1. Some people want to define
\[ \hat{v}_k = \frac{1}{N} \sum_{n=1}^{N} e^{-ikx_n} v_n, \]
then the Parseval is
\[ \left( \frac{2\pi}{N} \right) \sum_{n} u_n \overline{v}_n = 2 \pi \sum_{k=-N/2+1}^{N/2} \hat{u}_k \overline{\hat{v}}_k. \]

Inverse DFT

The inverse DFT is given by
\[ v_n = \frac{1}{N} \sum_{k=0}^{N-1} e^{ikx_n} \hat{v}_k = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} e^{ikx_n} \hat{v}_k. \]

(In Matlab convention, we should have \( v_n = \frac{1}{N} \sum_{k=0}^{N-1} e^{ikx_n-1} \hat{v}_k \), as \( v_n \) is associated with \( x_{n-1} \).

Proof. First of all, \( e^{-ik(x_n-x_j)} = e^{-ikx_n-j} \),
\[ \hat{w}_k = \sum_{n} u_j v_{n-j} e^{-ikx_n} = \sum_{j} u_j \sum_{n} v_{n-j} e^{-ikx_n} \]
\[ = \sum_{j} u_j e^{-ikj} \sum_{n} v_{n-j} e^{-ikx_n-j} = \sum_{j} u_j e^{-ikj} \hat{v}_k = \hat{u}_k \hat{v}_k. \]

For the second,
\[ \frac{1}{N} \sum_{k=0}^{N-1} e^{ikx_n} \hat{v}_k = \frac{1}{N} \sum_{k=0}^{N-1} e^{ikx_n} \sum_{m} v_m e^{-ikx_m} = \frac{1}{N} \sum_{m} v_m \sum_{k} e^{ik(x_n-x_m)} \]
Clearly, \( \sum_{k} e^{ik(x_n-x_m)} = \sum_{k} e^{ik(n-m)h} \). If \( n = m \), the sum is \( N \). If \( n \neq m \), the sum equals zero by the geometric sum since \( [e^{i(k(n-m)h)]}^N = 1 \). Hence,
\[ \frac{1}{N} \sum_{m} v_m \sum_{k} e^{ik(x_n-x_m)} = \frac{1}{N} \sum_{m} v_m N \delta_{n-m} = v_n. \]
3 FFT: fast algorithms to compute DFT

The most frequently used one is the Cooley-Tukey algorithm (the algorithm was independently discovered also by Gauss).

The idea is based on the simple fact. Let \( N \) be even, then

\[
\sum_{n=1}^{N} u_n e^{-i\frac{2\pi}{N} n} = \sum_{m=1}^{N/2} u_{2m} e^{-i\frac{2\pi}{N} m} + e^{i\frac{2\pi}{N}} \sum_{m=1}^{N/2} u_{2m-1} e^{-i\frac{2\pi}{N} m}
\]

The DFT of an array of size \( N \) is reduced to 2 DFT of arrays with size \( N/2 \) plus extra \( N \) operations. By this way, the whole complexity is \( O(N \log N) \).

4 Fourier Differentiation

For general \( x \), we can define a function

\[
f(x) = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} e^{i k x} \hat{v}_k,
\]

and then we use \( f^{(m)}(x_j) \) to approximate the derivatives \( v^{(m)}(x_j) \).

4.1 Algorithm for Fourier differentiation

The above idea generates the following Fourier differentiation:

- Given \( v = (v_1, \ldots, v_N) \), compute \( \hat{v} \).
- Define \( \hat{w}_k = (ik)^m \hat{v}_k \), for \( k = -N/2+1, \ldots, N/2 \) (or \( k = 0, \ldots, N/2, -N/2+1, \ldots, -1 \) in Matlab.)
- Compute the inverse DFT (inverse FFT) and get \( w_k \), the real part of which is the approximation of the derivative.

We take real part because numerical error may introduce some complex values.

**Corollary 1.** For the Fourier differentiation, without taking the real part, the discrete integration by parts by Fourier differentiation holds.

This is a corollary of the Paserval equality.

**Exercise:** Will it matter if we use \( f(x) = \frac{1}{N} \sum_{k=0}^{N-1} e^{i k x} \hat{v}_k \) to compute the derivatives? If it matters, which one is better? Choose a periodic smooth
function and code up to check out. Can you explain this? (Hint: Use the Aliasing formula in the next lecture.)

However, the $k = N/2$ mode is a little bit strange. Assume $v$ is a real array. Then,

$$\hat{v}_{N/2} = \bar{v}_{-N/2} = \bar{v}_{N/2}.$$  

This means $\hat{v}_{N/2}$ is real. This mode will contribute derivative

$$\hat{v}_{N/2} \frac{iN}{2} e^{i \frac{N}{2} x_j}.$$  

Clearly, this part is imaginary at the given grid points.