Chapter 12

Introduction to Spectral Methods

The essential idea behind spectral methods is to approximate the desired solution $u(x,t)$ to a PDE by an expansion:

$$\boxed{u(x,t) \simeq \sum_{m=0}^{M} a_m(t)\phi_m(x)}$$

where $\phi_m(x)$ are spectral "modes" and $a_m(t)$ are coefficients or "amplitudes".

The best choice for the spectral modes depends on a number of factors including the type of the PDE, the type of spatial domain and the type of boundary conditions. Trigonometric functions and polynomials (Chebyshev or Legendre polynomials) are the most frequently used. In the trigonometric case the most natural indexing of modes is not from 0 to M, but it is useful to consider this generic indexing. Frequently we shall not explicitly write the summation limits.

12.1 Spectral Solution of the Heat Equation

Let us consider as a simple illustrative example a spectral solution to the heat equation with homogeneous Dirichlet boundary conditions on the interval $[0,\ell]$:

$$\begin{aligned} \text{PDE:} &\quad \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} && (12.1)\\ \text{BC:} &\quad u(0,t) = 0 \ \text{ and } \ u(\ell,t) = 0, && (12.2)\\ \text{IC:} &\quad u(x,0) = u^0(x) && (12.3) \end{aligned}$$

The numerical approximation is written:

$$u(x,t) \simeq U(x,t) = \sum_{m=0}^{M} a_m(t)\phi_m(x) \qquad (12.4)$$

Substituting this into the PDE gives:

$$\begin{aligned} \frac{\partial}{\partial t}\sum_m a_m(t)\phi_m(x) &= \frac{\partial^2}{\partial x^2}\sum_m a_m(t)\phi_m(x) && (12.5)\\ \sum_m \frac{da_m}{dt}\phi_m &= \sum_m a_m \frac{d^2\phi_m}{dx^2} && (12.6) \end{aligned}$$

For this problem it is appropriate to take the functions $\phi_m(x)$ to be:

$$\phi_m(x) = \sin(\beta_m x), \qquad (12.7)$$

with $\beta_m = m\pi/\ell$. Note $\phi_{m=0} \equiv 0$ so $m = 1$ to M.

We can then take the **exact second derivative of the modes**:

$$\frac{d^2}{dx^2}\phi_m(x) = \frac{d^2}{dx^2}\sin(\beta_m x) = -\beta_m^2 \sin(\beta_m x).$$

Then (12.6) becomes:

$$\sum_m \frac{da_m}{dt}\sin(\beta_m x) = \sum_m -\beta_m^2 a_m \sin(\beta_m x). \qquad (12.8)$$

Using the orthogonality of the sine functions on the interval $[0,\ell]$ this implies

$$\frac{da_m}{dt}(t) = -\beta_m^2 a_m(t), \quad m = 1,2,3,\ldots,M \qquad (12.9)$$

One see the close connect between the spectral approach and the separation of variables approach to finding the exact solutions. Note there are also many similarities to the amplitude-equation approach to the study of pattern formation in nonlinear PDEs.

While for the heat equation we can solve equations (12.9) in closed form, for any equation for which a numerical solution is necessary this would not be the case. These means discretizing time and time stepping the amplitudes using some standard time-stepping method.

Thus $a_m(t) \to a_m^n = a_m(t_n)$. In the simplest case of forward Euler time stepping we then have:

$$\frac{a_m^{n+1} - a_m^n}{\triangle t} = -\beta_m^2 a_m^n.$$

or

$$a_m^{n+1} = (1 - \triangle t\beta_m^2)a_m^n. \qquad (12.10)$$

The amplitudes a_m^0 at $n = 0$ are found from the initial condition $u^0(x)$ via the sine transform:

$$a_m^0 = 2/\ell \int_0^\ell \sin(\beta_m x)u^0(x)dx \qquad (12.11)$$

In practice this would probably be found numerically via discrete sine transform of the initial condition on a set of grid points:

$$a_m^0 = 2/J \sum_{j=1}^{J-1} \sin(\beta_m x_j) u^0(x_j) = 2/J \sum_{j=1}^{J-1} \sin(\pi m j/J) u^0(x_j) \tag{12.12}$$

The numerical solution to the heat equation (12.1)-(12.3) is then found as follows. The initial amplitudes a_m^0 are found from the initial $u^0(x)$ via equation (12.12). Then amplitudes are then time steppedvia (12.10), or some other standard method, until the final time is reached. The numerical solution at the final time (or any desired intermediate time) is found from:

$$u(x, t_n) \simeq U(x, t_n) = \sum_{m=1}^{M} a_m^n \sin(\beta_m x) \tag{12.13}$$

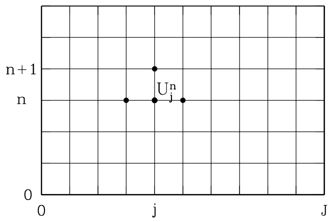
Note that the solution can be evaluated at any x in the spatial domain. In practice, however, the solution is generally evaluated on a grid x_j. For a regular grid this reduces again to discrete sine transform.

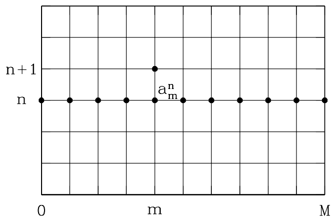
12.1.1 Caution

In this simple example, the evolution equations for the amplitudes decouple from one another. For more complicated problems such simple decoupling does not occur. Also, in this simple example the spectral modes all satisfied the boundary conditions for the problem. In general this will not be the case and addition effort must be made to obtain solutions which satisfy desired boundary conditions.

12.2 General Remarks on Finite-Difference and Spectral Approaches

Shown is a comparison of the finite-difference and spectral approximations. In the finite-difference approach the desired solution $u(x, t)$ is approximated by values U_j^n on a lattice (x_j, t_n). Space and time derivates are approximated by small differences on the lattice and substitution into the PDE gives a rule for advancing the solution from one time level to the next. The rules are (usually) local in space and time involving only a small neighborhoods of points(although for an implicit time-stepping method such local neighborhoods result in global coupling).





In the spectral approach the approximation to the solution $u(x, t)$ is given in terms of a set of values a_m^n on a mode-number × time lattice. The rule for advancing the amplitudes a_m from one time slice to the next are found from substituting the spectral expansion (12.4) into the PDE.

Orthogonal functions are to spectral methods as Taylor series are to finite-difference methods.

- Unlike the FD method, the "boundaries" in mode-number space $m = 0$ and $m = M$ have no correspondence with physical boundaries. Boundaries and boundary conditions appear as conditions on the modes ϕ_m.
- Time is treated the same, or at least similarly, in the two approaches. It is the treatment of space that is fundamentally different. As a result the treatment of space can to a large degree be discussed separately from the time-stepping methods. We shall often take a semi-discrete perspective in the spectral approach and treat time as a continuous variable assuming exact solutions to the evolution equations for the amplitudes.
- The approach we shall concentrate on is called the pseudo-spectral method. It utilizes both a spatial grid, as in finite-difference methods, and a spectral expansion. The spectral representation is used as much as possible. In particular, all spatial derivatives are computed spectrally. The physical mesh is used for handling non-uniformities (variable coefficients)and nonlinearities. To achieve maximum benefit it is necessary to transform between these two representations repeatedly, possibly several times per time step.

12.3 Advantages and Disadvantages

The greatest advantage of spectral over finite difference methods is the greater spatial accuracy. For smooth solutions, the discretization error of a spectral method decreases faster than any power of the resolution M:

$$\boxed{E^f = O(M^{-k}) \quad \textbf{for every } k.}$$

Spectral approximations are said to be exponentially accurate.

Recall, for **second-order finite differences we found:**

$$E^f = O(h^2)$$

For a given number of grid points $N = J$ we have $h = O(1/N)$ so

$$E^f = O(N^{-2})$$

12.3.1 Important Points

- The accuracy of spectral methods translates into fewer unknowns and thus greater speed and less memory for the same accuracy compared with FD methods.
- Given fast transforms, implicit time-stepping can be efficiently implemented.
- Spectral methods typically produce smaller artificial dissipation and dispersion in comparison to FD methods.

12.3.2 Advantages of finite difference methods

FD methods do have advantages. The primary advantage of FD methods is their flexibility. They are able to handle variable coefficients, general boundary conditions, free-boundaries etc. In higher dimensions they are able to handle irregular geometries more easily than spectral methods.

In many cases FD methods are easier to implement and thus same valuable programming time. Not only are these easier to implement, but the methods are flexible in that can often make minor modifications in existing program to handle new situation.

12.3.3 Disadvantages of spectral methods

The primary disadvantage of the spectral approach stems from its global nature. As a result highly localized solutions not well treated. General geometries can be difficult. Boundary conditions and in particular free boundaries can be very difficult. Generally more time is necessary for code development and modification (a small change in a PDE can result in significant code rewriting).

Pseudospectral method is the method of choice for obtaining maximum flexibility while maintaining the good convergence properties of spectral approximations.

Chapter 13

Fourier Pseudospectral Methods

The spectral approach consists of approximating the solution $u(x,t)$ of a PDE by

$$u(x,t) \simeq U(x,t) = \sum_{m=0}^{M} a_m(t)\phi_m(x) \tag{13.1}$$

To implement this in practice there are many issues which must be addressed.

- **What functions $\phi_m(x)$ to use.**
- **How to choose the a_m so as to "best" satisfy the PDE.**

Our choice will give the **Fourier pseudospectral approximation**.

- **The most important issue in practice is - Fourier transforms.**

13.1 Functions (modes) ϕ_m

For periodic problems (PDEs with periodic boundary conditions) the natural choice is trigonometric functions (sines, and cosines):

$$\phi_m(x) = \begin{cases} \sin(\beta_m x) \\ \cos(\beta_m x) \end{cases} \tag{13.2}$$

or equivalently complex exponentials:

$$\phi_m(x) = e^{i\beta_m x} \tag{13.3}$$

where $\beta_m = 2m\pi/\ell$.

This choice is the best in the sense that they give the most accurate approximation.

These are commonly referred to as **Fourier modes**.

13.2 Amplitudes a_m

Given a finite number of modes $\phi_m(x)$ we need to decide how to choose the amplitudes a_m. This is not completely trivial.

Intuitively we can understand the issue with a simple example. Consider a linear PDE:

$$\frac{\partial u}{\partial t}(x,t) = \mathcal{L}u(x,t) \tag{13.4}$$

for x in $[0,\ell]$ and a spectral approximation to the solution

$$U(x,t) = \sum_{m=0}^{M} a_m(t)\phi_m(x) \tag{13.5}$$

For now let $a_m(t)$ depend continuously on time. Later we will need to discretize time also, but this is not the issue here.

Substituting $U(x,t)$ into the left-hand-side and right-hand-side of the PDE gives two functions of (x,t),

$$\frac{\partial U}{\partial t}(x,t) \quad and \quad \mathcal{L}U(x,t). \tag{13.6}$$

In general, we cannot expect that by adjusting M coefficients $a_m(t)$ we can get these two functions to be identical for all x in $[0,\ell]$.

What we have is:

$$\frac{\partial U}{\partial t}(x,t) = \mathcal{L}U(x,t) + R(x,t) \tag{13.7}$$

where R is the residual. Unless $U(x,t)$ is an exact solution to the PDE (which generally it cannot be), then $R(x,t)$ will not be identically 0.

Different choices for the a_m will give different residuals. Or the other way around, different choices of the residual will dictate different choices for the a_m and hence different choices for the numerical approximation $U(x,t)$.

There are 3 common choices for conditions on R and hence 3 common choices for selecting the a_m. These result in so-called **Galerkin, tau, and pseudospectral methods**.

The pseudospectral method corresponds to the choice

$$R(x_j,t) = 0$$

on a set of grid of points x_j known as collocation points.

Evaluating

$$\frac{\partial U}{\partial t}(x,t) = \mathcal{L}U(x,t) + R(x,t) \tag{13.8}$$

on the grid gives

$$\left[\frac{\partial U}{\partial t}\right](x_j,t) = [\mathcal{L}U](x_j,t) + R(x_j,t) = [\mathcal{L}U](x_j,t) \tag{13.9}$$

This is J conditions if there are J grid (collocation) points x_j.

If done correctly, these J conditions uniquely determine the the amplitudes a_m.

13.2.1 Comments on the pseudospectral approach

In practice, one does not work so formally.

- The point is that in the pseudospectral approach one uses *both* a set of amplitudes a_m and a set values U_j on a collocation grid x_j.
- The advantage of the pseudospectral approach is that it can be applied readily to PDEs with variable coefficients, inhomogeneities and nonlinearities.
- Operations such as differentiation are best carried out in spectral space (because the modes $\phi_m(x)$ can be differentiated easily and exactly) whereas complications arising from inhomogeneities and nonlinearities are best handled in physical space where they are local.

Example: Ignore time and consider the variable coefficient operator

$$x\frac{\partial^2}{\partial x^2}.$$

Then in the Fourier pseudospectral approach where $\phi_m(x) = e^{i\beta_m x}$ one would first evaluate $\frac{\partial^2 U}{\partial x^2}$ spectrally

$$\frac{\partial^2 U}{\partial x^2}(x) = \sum_m -\beta_m^2 a_m e^{i\beta_m x} \tag{13.10}$$

then evaluate $x\frac{\partial^2 U}{\partial x^2}$ on the collocation grid.

$$x_j\frac{\partial^2 U}{\partial x^2}(x_j) = x_j\left(\sum_m -\beta_m^2 a_m e^{i\beta_m x_j}\right) \tag{13.11}$$

In a "pure" spectral approach one would only ever consider the spectral representation in terms of amplitudes a_m. The operator $x\frac{\partial^2}{\partial x^2}$ is complicated when expressed in the spectral representation.

13.3 Fourier Transforms

Because in the pseudospectral approach one uses both a set of amplitudes a_m and a set values U_j on a collocation grid x_j we need a way to transform between the two representation. In the case of periodic problems and Fourier modes, this is the Fourier transform. Or more accurately, the **discrete real-to-complex Fourier transform.**

Consider a set of values U_j on equally spaced grid points $x_j = jh$ for $j = 0, \dots, J-1$. This includes end point $j = 0$ but not end point $j = J$ because $U_J = U_0$ for a periodic function.

$$\textit{DFT:}\quad \mathbf{a_m} = \frac{1}{J}\sum_{j=0}^{J-1} \mathbf{U_j}\, e^{-i2\pi mj/J}, \quad m = -J/2+1, -J/2+2, \dots, J/2,$$

$$\textit{Inverse DFT:}\quad \mathbf{U_j} = \sum_{m=-J/2+1}^{J/2} \mathbf{a_m}\, e^{i2\pi mj/J} \quad j = 0, 1, \dots, J-1,$$

We have assumed J is even. **Note the maximum of m is $J/2$.**

13.3.1 Counting

Given J real values U_j, the discrete real-to-complex Fourier transform gives J complex amplitudes a_m

$$(U_0, U_1, U_2, \dots U_{J-1}) \xrightarrow{\text{DFT}} \left(a^r_{-J/2+1}, a^i_{-J/2+1}, \dots a^r_{J/2}, a^i_{J/2}\right) \tag{13.12}$$

Since each complex a_m has a real and imaginary part $a_m = a^r_m + ia^i_m$, the DFT would seem to have effectively doubled the number of values.

However, there is a symmetry in the a_m, namely $a_{-m} = a^*_m$. Hence the negative half of the spectrum: $a_{-1}, a_{-2}, \dots, a_{-J/2+1}$ is known from the positive half and thus is should not be counted. This leaves us with the $J/2+1$ complex values $a_0, a_1, \dots, a_{J/2}$. However, a_0 is necessarily real because $a^*_0 = a_{-0} = a_0$. Similarly $a_{J/2}$ is necessarily real (try to show this). Hence there are precisely J distinct real quantities in the a_m $m = -J/2+1, -J/2+2, \dots, J/2$. These are $a^r_0, a^r_1, a^i_1, \dots, a^r_{J/2-1}, a^i_{J/2-1}, a^r_{J/2}$. So that

$$(U_0, U_1, U_2, \dots U_{J-1}) \xrightarrow{\text{DFT}} \left(a^r_0, a^r_1, a^i_1, \dots, a^r_{J/2-1}, a^i_{J/2-1}, a^r_{J/2}\right)$$

13.4 The Fast Fourier Transform (FFT) Libraries

The **fast Fourier transform (FFT)** is technique for performing discrete Fourier transforms in a computationally efficient way. Details can be found many places. The essential point is that a naive implementation of the DFT requires computational work J^2 for J grid values. The FFT computes the same transform with computational work $J\log_2 J$, which for large J is much less.

There are many libraries available that perform fast discrete real-to-complex Fourier transforms.

The problem is that given real data U_j there is no universal agreement about how the complex amplitudes a_m are returned. Hence you need a manual and work out what is returned from any given FFT. We shall use this ordering.

$$(U_0, U_1, U_2, \dots U_{J-1}) \underset{\text{FFT}^{-1}}{\overset{\text{FFT}}{\Longleftrightarrow}} \left(a^r_0, a^r_1, a^i_1, \dots, a^r_{J/2-1}, a^i_{J/2-1}, a^r_{J/2}\right)$$

This ordering is almost never used in practice. Also, calling an FFT routine followed by an inverse FFT routine almost never gives you back what you started with.

Chapter 14

Fourier Pseudospectral Time stepping

In previous chapters we discussed spectral methods in general and introduced the Fourier pseudospectral approach. In this chapter we will consider in detail how to implement Fourier pseudospectral time stepping for PDEs with periodic boundary conditions.

14.0.1 Grid values

As usual we let U_j to be the numerical approximation to u on a grid. We use a uniform grid $x_j = jh$ where $h = \ell/J$.

A small complication in implementation and notation arises because the grid contains $J+1$ points and yet U_J is not (and cannot be) included when calling FFT library routines.

Due to periodic boundary conditions $U_J = U_0$, so that U_J is redundant, and there are only J distinct real numbers representing our solution.

For proper treating FFTs, throughout this chapter we shall take $\mathbf{U}$ to be an array of J values

$$\mathbf{U} = (U_0, U_1, \ldots, U_{J-1})^T. \tag{14.1}$$

We have consistently used $J+1$ grid points in our actual computer implementations of PDE solvers and we wish to continue to do so. Hence in the actual implementation of pseudospectral time stepping we shall continue to use $J+1$ values for the solution and we shall take into account this additional grid point through a boundary operator.

14.0.2 Amplitudes

In practice there are a variety of conventions for ordering the Fourier amplitudes a_m obtained numerically by FFT libraries.

To facilitate our treatment we shall use a generic ordering (form of storage) for the real a^r_m and imaginary

a^i_m parts of these complex amplitudes and let $\mathbf{a}$ denote the real vector:

$$\mathbf{a} = (a^r_0, a^r_1, a^i_1, a^r_2, a^i_2, \ldots, a^r_{J/2-1}, a^i_{J/2-1}, a^r_{J/2})^T \tag{14.2}$$

$\mathbf{a}$ is a real vector of length J.

Recall that necessarily $a^i_0 = a^i_{J/2} = 0$ and these are not included in the list.

In practice when implementing our time-stepping schemes, proper account must be taken of the ordering actually produced by the FFT routine used. In addition, the amplitudes are generally not normalized and require an additional normalization (such as division by J).

14.0.3 Discrete Fourier Transforms

We shall use the following notation for the FFTs relating the $\mathbf{a}$'s and $\mathbf{U}$'s:

$$\mathbf{a} = \text{FFT}\ \{\mathbf{U}\}$$
$$\mathbf{U} = \text{FFT}^{-1}\ \{\mathbf{a}\}$$

or

$$\mathbf{U} \xrightarrow{\text{FFT}} \mathbf{a}$$
$$\mathbf{U} \xleftarrow{\text{FFT}^{-1}} \mathbf{a}$$

Both $\mathbf{U}$ and $\mathbf{a}$ are real vectors of length J.

14.1 Differentiation

Before considering the time-dependent problem, it is useful to consider how to take spatial derivatives in the pseudospectral approximation. For concreteness consider the operator:

$$\mathcal{L} = \frac{\partial^2}{\partial x^2}$$

We then need to compute $\mathbf{LU}$ where $\mathbf{L}$ is our (spectral) approximation to $\mathcal{L}$.

Note:

$$\mathbf{L} \neq \frac{1}{h^2}\begin{bmatrix} & & \\ 1 & -2 & 1 \\ & & \\ & & \end{bmatrix}$$

We never consider the action of $\mathbf{L}$ directly. Instead we use our spectral expansion:

$$U(x) = \sum_m a_m e^{i\beta_m x}, \quad \beta_m = 2\pi m/\ell$$

Then:

$$\begin{aligned} [\mathbf{LU}]_j &\equiv [\mathcal{L}U(x)]_{x=x_j} \\ &= \left[\mathcal{L}\sum_m a_m e^{i\beta_m x}\right]_{x=x_j} \\ &= \left[\sum_m a_m \mathcal{L} e^{i\beta_m x}\right]_{x=x_j} \\ &= \left[\sum_m a_m(-\beta_m^2)e^{i\beta_m x}\right]_{x=x_j} \\ &= \sum_m -\beta_m^2 a_m e^{i\beta_m x_j} \end{aligned}$$

So the second derivative is spectral space is just multiplication by $-\beta_m^2$.

We can define the second-derivative matrix in spectral space as:

$$\hat{\mathbf{L}} \equiv \begin{bmatrix} -\beta_0^2 &&&&&&& \\ & -\beta_1^2 &&&&&& \\ && -\beta_1^2 &&&&& \\ &&& \ddots &&&& \\ &&&& -\beta_m^2 &&& \\ &&&&& -\beta_m^2 && \\ &&&&&& \ddots & \\ &&&&&&& -\beta_{J/2}^2 \end{bmatrix} = \text{“}diag\{-\beta_m^2\}''$$

So that differentiation in spectral space is:

$$\hat{\mathbf{L}}\mathbf{a}$$

While we write this as a matrix, it is diagonal so very easy to implement in spectral space.

Since $\mathbf{a}$ and $\mathbf{U}$ are related via FFTs, we have:

$$\mathbf{LU} = \mathrm{FFT}^{-1}\left\{\hat{\mathbf{L}}\mathbf{a}\right\} = \mathrm{FFT}^{-1}\left\{\hat{\mathbf{L}}\ \ \mathrm{FFT}\left\{\mathbf{U}\right\}\right\}$$

$$\boxed{\mathbf{LU} = \mathrm{FFT}^{-1}\,\hat{\mathbf{L}}\,\mathrm{FFT}\,\mathbf{U}}$$

Note that this means:

$$\mathbf{L} = \mathrm{FFT}^{-1}\,\hat{\mathbf{L}}\,\mathrm{FFT} \tag{14.3}$$

so $\mathbf{L}$ is obtained from the diagonal matrix $\hat{\mathbf{L}}$ by a unitary transformation.

We use the right-hand-side of the above as a method of computing $\mathbf{LU}$. In individual steps this would be:

$$\boxed{\mathbf{U} \xrightarrow{\mathrm{FFT}} \mathbf{a} \xrightarrow{\hat{\mathbf{L}}} \hat{\mathbf{L}}\mathbf{a} \xrightarrow{\mathrm{FFT}^{-1}} \mathbf{LU}}$$

I like to think in terms of the following diagram:

Other derivatives can be computed in a similar way.

14.2 Pseudospectral time stepping

We can now consider a method for time stepping the heat equation:

$$\frac{\partial u}{\partial t} = \mathcal{L}u = \frac{\partial^2 u}{\partial x^2}$$

Letting $u(x,t) \to U_j^n$ where:

$$U_j^n = U(x_j, t_n) = \sum_m a_m(t_n)e^{i\beta_m x_j} = \sum_m a_m^n e^{i\beta_m x_j}$$

First consider forward Euler time stepping:

$$\frac{\partial u}{\partial t} \simeq \frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\triangle t} = \mathbf{L}\mathbf{U}^n$$

then as usual this becomes:

$$\mathbf{U}^{n+1} = (\mathbf{I} + \triangle t\mathbf{L})\mathbf{U}^n = \mathbf{A}_+\mathbf{U}^n$$

Again, $\mathbf{L}$ and hence $\mathbf{A}_+$ are not the simple tridiagonal matrices from the finite-difference approach. These matrices are, however, diagonal in spectral space. We have:

$$\begin{aligned} \mathbf{U}^{n+1} &= \mathbf{FFT}^{-1}\,(\mathbf{I} + \triangle t\hat{\mathbf{L}})\ \mathbf{FFT}\ \mathbf{U}^n \\ &= \mathbf{FFT}^{-1}\ \hat{\mathbf{A}_+}\ \mathbf{FFT}\ \mathbf{U}^n \end{aligned}$$

where

$$\hat{\mathbf{A}_+} \equiv \begin{bmatrix} 1-\triangle t\beta_0^2 &&&&& \\ & \ddots &&&& \\ && 1-\triangle t\beta_m^2 &&& \\ &&& 1-\triangle t\beta_m^2 && \\ &&&& \ddots & \\ &&&&& 1-\triangle t\beta_{J/2}^2 \end{bmatrix} = \text{“}diag\{1-\triangle t\beta_m^2\}'' \tag{14.4}$$

We can write our time-stepping method as:

$$\boxed{\mathbf{U}^n \xrightarrow{\text{FFT}} \mathbf{a^n} \xrightarrow{\hat{\mathbf{A}_+}} \mathbf{a^{n+1}} = \hat{\mathbf{A}_+}\mathbf{a^n} \xrightarrow{\text{FFT}^{-1}} \mathbf{U^{n+1}}}$$

Backward Euler and Crank-Nicolson methods can be treated in exactly the same way. Note for example that for backward Euler:

$$\begin{aligned}\mathbf{U}^{n+1} &= \text{FFT}^{-1}\,(\mathbf{I} - \triangle t\hat{\mathbf{L}})^{-1}\,\text{FFT}\,\mathbf{U}^n \\ &= \text{FFT}^{-1}\,\hat{\mathbf{A}_-}^{-1}\,\text{FFT}\,\mathbf{U}^n\end{aligned}$$

where

$$\hat{\mathbf{A}_-} \equiv \begin{bmatrix} 1+\triangle t\beta_0^2 & & & & & \\ & \ddots & & & & \\ & & 1+\triangle t\beta_m^2 & & & \\ & & & 1+\triangle t\beta_m^2 & & \\ & & & & \ddots & \\ & & & & & 1+\triangle t\beta_{J/2}^2 \end{bmatrix} = \text{“}diag\{1+\triangle t\beta_m^2\}'' \quad (14.5)$$

Since $\hat{\mathbf{A}_-}$ is diagonal hence finding its inverse is trivial: $[\hat{\mathbf{A}_-}^{-1}]_{jj} = 1/\hat{\mathbf{A}}_{-jj}$.

Crank-Nicolson time stepping follows similarly.

Question: Why not time step the modes exactly?

$$\hat{\mathbf{A}} \equiv \begin{bmatrix} \exp(-\triangle t\beta_0^2) & & & & & \\ & \ddots & & & & \\ & & \exp(-\triangle t\beta_m^2) & & & \\ & & & \exp(-\triangle t\beta_m^2) & & \\ & & & & \ddots & \\ & & & & & \exp(-\triangle t\beta_{J/2}^2) \end{bmatrix} = \text{“}diag\{e^{-\triangle t\beta_m^2}\}'' \qquad (14.6)$$

14.3 Nonlinearity in the Pseudospectral Approach

Consider the PDE:

$$\frac{\partial u}{\partial t} = \mathcal{L}u + \mathcal{N}(u)$$

with $\mathcal{L}$ and $\mathcal{N}()$ linear and nonlinear operators, respectively.

Examples would be:

$$\begin{aligned}
&\textbf{Fisher equation:} && \frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + ku(1-u) \\
&\textbf{Burger's equation:} && \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \\
&\textbf{Korteweg-deVries equation:} && \frac{\partial u}{\partial t} + \frac{\partial^3 u}{\partial x^3} + 6u\frac{\partial u}{\partial x} = 0 \\
&\textbf{Swift-Hohenberg equation:} && \frac{\partial u}{\partial t} = u - u^3 - (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - 1)^2 u
\end{aligned}$$

There are many possible methods to use. The simplest $O(\triangle t^2)$ is probably the Crank-Nicolson/Adams-Bashforth method:

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\triangle t} = \frac{1}{2}\left\{\mathbf{L}\mathbf{U}^{n+1} + \mathbf{L}\mathbf{U}^n\right\} + \frac{1}{2}\left\{3\mathbf{N}(\mathbf{U}^n) - \mathbf{N}(\mathbf{U}^{n-1})\right\}$$

where $N_j(\mathbf{U}^n)$ is the numerical approximation to $\mathcal{N}(u)$ at grid point j.

Solving for $\mathbf{U}^{n+1}$ we obtain:

$$\mathbf{U}^{n+1} = (\mathbf{I} - \triangle t\mathbf{L})^{-1}\left\{(\mathbf{I} + \triangle t\mathbf{L})\mathbf{U}^n + \frac{\triangle t}{2}\left(3\mathbf{N}(\mathbf{U}^n) - \mathbf{N}(\mathbf{U}^{n-1})\right)\right\} \qquad (14.7)$$

We now wish to act with the linear operators in spectral space. Apply discrete Fourier Transform and its inverse to the right-hand-side to obtain:

$$\mathbf{U}^{n+1} = \text{FFT}^{-1}\,\text{FFT}\,(\mathbf{I} - \triangle t\mathbf{L})^{-1}\left\{(\mathbf{I} + \triangle t\mathbf{L})\mathbf{U}^n + \frac{\triangle t}{2}\left(3\mathbf{N}(\mathbf{U}^n) - \mathbf{N}(\mathbf{U}^{n-1})\right)\right\} \qquad (14.8)$$

or

$$\mathbf{U}^{n+1} = \text{FFT}^{-1}\,(\mathbf{I} - \triangle t\hat{\mathbf{L}})^{-1}\left\{(\mathbf{I} + \triangle t\hat{\mathbf{L}})\,\text{FFT}\,\mathbf{U}^n + \text{ FFT}\,\frac{\triangle t}{2}\left(3\mathbf{N}(\mathbf{U}^n) - \mathbf{N}(\mathbf{U}^{n-1})\right)\right\}$$

where hats denote linear operators in spectral space.

Chapter 15

APPENDIX - Fourier Transforms

In this appendix I provide a comprehensive summary of various transform pairs. Notation varies in the literature and I present here a self-consistent treatment.

15.1 Sine Transform

Given a function $u(x)$ on the interval $[0,\ell]$, the sine transform and its inverse are given by:

$$\textit{Sine Transform:}\quad a_m = \frac{2}{\ell}\int_0^\ell u(x)\sin(\pi m x/\ell)dx \tag{15.1}$$

$$\textit{Inverse Sine Transform:}\quad u(x) = \sum_{m=1}^{\infty} a_m \sin(\pi m x/\ell) \tag{15.2}$$

To prove that these are indeed inverses of one another, one uses the *orthogonality relation* for sine functions.

$$\int_0^\ell \sin(\pi m x/\ell)\sin(\pi m' x/\ell)dx = \frac{\ell}{2}\delta_{mm'} \tag{15.3}$$

which is easy to prove by direct integration. One can also use the *completeness relation*:

$$\sum_{m=1}^{\infty}\sin(\pi m x/\ell)\sin(\pi m x'/\ell) = \frac{\ell}{2}\delta(x-x') \tag{15.4}$$

which is not easy to prove.

15.2 Discrete Sine Transform

There is a discrete form of the sine transformation, the discrete sine transform (DST). In this case we are given a set of value U_j for $j = 1,\ldots,J-1$ generally thought of as a discrete sampling of a function $u(x)$ on $[0,\ell]$, i.e. $U_j = u(jh)$, which $h = \ell/J$. For the DST the function is sampled only on the interior $j = 1,2,\ldots,J-1$. Then the discrete sine transform (DST) and its inverse are given by:

$$\textit{DST:}\quad a_m = \frac{2}{J}\sum_{j=1}^{J-1} U_j \sin(\pi m j/J), \quad m = 1,2,\ldots,J-1 \tag{15.5}$$

$$\textit{Inverse DST:}\quad U_j = \sum_{m=1}^{J-1} a_m \sin(\pi m j/J), \quad j = 1,2,\ldots,J-1 \tag{15.6}$$

One may think of (15.5) as arising from (15.1) by replacing the integral by a discrete (trapezoid) approximation.

Note that if one included the cases $m = 0$ and $m = J$ in (15.5) one would find that $a_0 = a_J = 0$. Hence there is not reason to include these. Similarly, U_0 and U_J would necessarily be zero by (15.6). Hence it is not possible to recover non-zero values of U_0 and U_J via the inverse DST and these are excluded.

The proof that the above pairs are inverses is easy given the orthogonality relation:

$$\sum_{j=1}^{J-1}\sin(\frac{\pi m j}{J})\sin(\frac{\pi m' j}{J}) = \frac{J}{2}\delta_{mm'} \tag{15.7}$$

Note that m and j play exactly the same role in $\sin(\frac{\pi m j}{J})$ so that the discrete analog of the completeness relation (15.4) holds as a direct consequence of (15.7) by relabelling j and m:

$$\sum_{m=1}^{J-1}\sin(\frac{\pi m j}{J})\sin(\frac{\pi m j'}{J}) = \frac{J}{2}\delta_{jj'} \tag{15.8}$$

Equations (15.7) and (15.8) are the same thing.

15.3 Cosine Transform

Given a function $u(x)$ on the interval $[0,\ell]$, the cosine transform and its inverse are given by:

$$\textit{Cosine Transform:}\quad a_m = \frac{2}{\ell}\int_0^\ell u(x)\cos(\pi m x/\ell)dx \tag{15.9}$$

$$\textit{Inverse Cosine Transform:}\quad u(x) = \frac{a_0}{2} + \sum_{m=1}^{\infty} a_m \cos(\pi m x/\ell) \tag{15.10}$$

$$= \sum_{m=0}^{\infty} w_m a_m \cos(\pi m x/\ell) \tag{15.11}$$

where the weight function w_m is given by:

$$w_j = \begin{cases} \frac{1}{2} & \text{if } j = 0 \\ 1 & \text{if } j > 0 \end{cases} \tag{15.12}$$

The proof that these are indeed inverses uses the orthogonality relation for cosine functions.

$$\int_0^\ell \cos(\pi m x/\ell)\cos(\pi m' x/\ell)dx = \frac{\ell}{2}\delta_{mm'} \tag{15.13}$$

which again is easy to prove by direct integration. These is also a difficult to prove completeness relation:

$$\sum_{m=0}^{\infty} w_m \cos(\pi m x/\ell)\cos(\pi m x'/\ell) = \frac{\ell}{2}\delta(x-x') \tag{15.14}$$

15.4 Discrete Cosine Transform

For the discrete cosine transform (DST) we consider a set of value U_j for $j = 0,\ldots,J$ again generally thought of as a discrete sampling of a function $u(x)$ on $[0,\ell]$. This time however we sample the function at the end points $j=0$ and $j=J$. Then the discrete cosine transform (DST) and its inverse are given by:

$$\textit{DCT:}\quad a_m = \frac{2}{J}\sum_{j=0}^{J} w_j U_j \cos(\pi m j/J) \tag{15.15}$$

$$\textit{Inverse DCT:}\quad U_j = \sum_{m=0}^{J} w_m a_m \cos(\pi m j/J) \tag{15.16}$$

$$\tag{15.17}$$

where now the weight function is:

$$w_j = \begin{cases} \frac{1}{2} & \text{if } j = 0, J, \\ 1 & \text{if } 0 < j < J \end{cases} \tag{15.18}$$

One may think of (15.15) as arising from (15.9) by replacing the integral by a trapezoid approximation. The actual proof that the above pairs are inverses comes from the following identity.

$$\sum_{j=0}^{J} w_j w_m \cos(\frac{\pi m' j}{J})\cos(\frac{\pi m j}{J}) = \frac{J}{2}\delta_{mm'} \tag{15.19}$$

Note that as the the DST, m and j play exactly the same role in this equation so that the discrete analog of the completeness relation (15.14) holds as a direct consequence of (15.19) by relabelling j and m.

15.5 Fourier Transform

Given a function $u(x)$ on the interval $[0,\ell]$, the Fourier transform (FT) and its inverse are given by:

$$\textit{Fourier Transform:}\quad a_m = \frac{1}{\ell}\int_0^{\ell} u(x) e^{-i2\pi m x/\ell} dx \tag{15.20}$$

$$\textit{Inverse Fourier Transform:}\quad u(x) = \sum_{m=-\infty}^{\infty} a_m e^{i2\pi m x/\ell} \tag{15.21}$$

In this case the function $u(x)$ is often allowed to be complex, but we shall consider only the case of real $u(x)$.

One has the following orthogonality relation for complex exponentials:

$$\int_0^{\ell} e^{i2\pi m x/\ell} e^{-i2\pi m' x/\ell} dx = \ell\delta_{mm'} \tag{15.22}$$

which as usual is easy to prove by direct integration.

(Notice that here we have $\int_0^{\ell}\phi_m(x)\phi^*_{m'}(x)dx = \ell\delta_{mm'}$, where $\phi_m(x) = e^{i2\pi m x/\ell}$.)

One can also use the completeness relation:

$$\sum_{m=-\infty}^{\infty} e^{i2\pi m x/\ell} e^{-i2\pi m x'/\ell} = \ell\delta(x-x') \tag{15.23}$$

which is not easy to prove.

In the case which $u(x)$ is a real-valued function, the a_m will still be complex, but the following symmetry holds:

$$a_{-m} = a^*_m \tag{15.24}$$

where $*$ denotes complex conjugation. Proof:

$$a^*_m = \left[\frac{1}{\ell}\int_0^{\ell} u(x)e^{-i2\pi m x/\ell}dx\right]^* = \frac{1}{\ell}\int_0^{\ell} u^*(x)e^{i2\pi m x/\ell}dx = \frac{1}{\ell}\int_0^{\ell} u(x)e^{-i2\pi(-m)x/\ell}dx = a_{-m}$$

Using this fact, it is possible to re-write the FT pair in such a way that all quantities are real:

$$\textit{Inverse Fourier Transform:}\quad u(x) = \frac{A_0}{2} + \sum_{m=1}^{\infty}\left(A_m\cos(2\pi m x/\ell) + B_m\sin(2\pi m x/\ell)\right)$$

with:

$$\begin{aligned} \textit{Fourier Transform:}\quad A_m &= 2a^r_m = \frac{2}{\ell}\int_0^{\ell} u(x)\cos(2\pi m x/\ell)dx \\ B_m &= -2a^i_m = \frac{2}{\ell}\int_0^{\ell} u(x)\sin(2\pi m x/\ell)dx \end{aligned}$$

where a^r_m and a^i_m are the real and imaginary parts of a_m: $a_m = a^r_m + ia^i_m$.

Hence the FT of a real-valued function is equivalent to taking both sine and cosine transforms on the interval $[0,\ell]$ but with argument $2\pi m x/\ell$ rather than $\pi m x/\ell$.

The details are left for the reader, but the first few lines of the derivation are:

$$\begin{aligned} u(x) &= a_0 + \sum_{m=1}^{\infty} a_m e^{i2\pi m x/\ell} + \sum_{m=-1}^{-\infty} a_m e^{i2\pi m x/\ell} \\ &= a_0 + \sum_{m=1}^{\infty} a_m e^{i2\pi m x/\ell} + \sum_{m=1}^{\infty} a_{-m} e^{-i2\pi m x/\ell} \\ &= a_0 + \sum_{m=1}^{\infty}\left(a_m e^{i2\pi m x/\ell} + a^*_m e^{-i2\pi m x/\ell}\right) \end{aligned}$$

15.6 Discrete Fourier Transform

For the discrete Fourier transform (DFT) we again consider a set of value U_j for $j = 0,\ldots,J$ thought of as a discrete sampling of a function $u(x)$ on $[0,\ell]$ including end points $j=0$ and $j=J$. In this case we are going to consider several forms for the DFT. The first is given by:

$$\textit{DFT:}\quad a_m = \frac{1}{J}\sum_{j=0}^{J-1} U_j e^{-i2\pi mj/J}, \quad m = -J/2+1, -J/2+2, \ldots, J/2, \tag{15.25}$$

$$\textit{Inverse DFT:}\quad U_j = \sum_{m=-J/2+1}^{J/2} a_m e^{i2\pi mj/J} \quad j = 0, 1, \ldots, J-1, \tag{15.26}$$

where we have assumed J is even. This is the form we shall consider most frequently in this course.

The proof that the above pairs are inverses comes from the orthogonality of the complex exponentials:

$$\sum_{j=0}^{J-1} e^{-i2\pi mj/J} e^{i2\pi m' j/J} = J\delta_{mm'} \tag{15.27}$$

As with the other discrete transforms, this is also equivalent to the completeness relation in the continuous case.

Other standard and non-standard forms of the DFT can be derived by noting that if (15.25) is used to define a_m for all integer m, then the a_m are periodic with period J. Similarly, (15.26) gives periodic U_j with period J.

$$a_{m+kJ} = a_m \quad \text{and} \quad U_{j+kJ} = U_j$$

for all integer k. Proof:

$$\begin{aligned} a_{m+kJ} &= \frac{1}{J}\sum_{j=0}^{J-1} U_j e^{-i2\pi(m+kJ)j/J} \\ &= \frac{1}{J}\sum_{j=0}^{J-1} U_j e^{-i2\pi mj/J} e^{-i2\pi kj} \\ &= \frac{1}{J}\sum_{j=0}^{J-1} U_j e^{-i2\pi mj/J} = a_m \end{aligned}$$

Using this, the following is an equivalent form for the DFT and its inverse:

$$\textit{DFT:}\quad a_m = \frac{1}{J}\sum_{j=0}^{J} w_j U_j e^{-i2\pi mj/J}, \quad m = -J/2, -J/2+1, \ldots, J/2, \tag{15.28}$$

$$\textit{Inverse DFT:}\quad U_j = \sum_{m=-J/2}^{J/2} \hat{w}_m a_m e^{i2\pi mj/J} \quad j = 0, 1, \ldots, J-1, \tag{15.29}$$

where:

$$w_j = \begin{cases} \frac{1}{2} & \text{if } j = 0, J, \\ 1 & \text{if } 0 < j < J \end{cases} \tag{15.30}$$

and

$$\hat{w}_m = \begin{cases} \frac{1}{2} & \text{if } m = -J/2, J/2, \\ 1 & \text{if } -J/2 < m < J/2 \end{cases} \tag{15.31}$$

In effect, since $U_0 = U_J$, U_0 in (15.25) can been replaced by $(U_0 + U_J)/2$. Similarly for (15.29) since $a_{-J/2} = a_{J/2}$.

This is a non-standard form for the DFT. I give it here because it is the form one might guess from the continuous FT (15.20)-(15.21).

Probably the most common form for the DFT is given by using the periodicity in a_m to shift the sum in (15.26) and obtain:

$$\textit{DFT:}\quad a_m = \frac{1}{J}\sum_{j=0}^{J-1} U_j e^{-i2\pi mj/J}, \quad m = 0, 1, \ldots, J-1, \tag{15.32}$$

$$\textit{Inverse DFT:}\quad U_j = \sum_{m=0}^{J-1} a_m e^{i2\pi mj/J} \quad j = 0, 1, \ldots, J-1, \tag{15.33}$$

This form has the advantage that both sums are over the same range. However, it has the disadvantages that (1) it is not a natural choice given the continuous FT, (2) more importantly, we will consider real U_j and this form does not make it as obvious that U_j obtained by the inverse transformation will be real.

Most libraries that perform discrete Fourier transforms say they compute transforms of type (15.32)-(15.33). They are all equivalent of course.

Finally, we consider the sine/cosine forms of the DFT. As in the continuous case: U_j real implies $a_{-m} = a_m^*$. Then (15.26) gives:

$$\begin{aligned} U_j &= \sum_{m=-J/2+1}^{J/2} a_m e^{i2\pi mj/J} && (15.34) \\ &= a_0 + \sum_{m=1}^{J/2-1} a_m e^{i2\pi mj/J} + \sum_{m=-1}^{-(J/2-1)} a_m e^{i2\pi mj/J} + a_{J/2} e^{i2\pi(J/2)j/J} && (15.35) \\ &= a_0 + \sum_{m=1}^{J/2-1} \left(a_m e^{i2\pi mj/J} + a_m^* e^{-i2\pi mj/J} \right) + a_{J/2} e^{i\pi j} && (15.36) \\ &= a_0 + \sum_{m=1}^{J/2-1} \left(2a_m^r \cos(2\pi mj/J) + (-2a_m^i)\sin(2\pi mj/J) \right) + a_{J/2}\cos(\pi mj/(J/2)) && (15.37) \\ &= \frac{A_0}{2} + \sum_{m=1}^{J/2-1} \left(A_m \cos(2\pi mj/J) + B_m \sin(2\pi mj/J) \right) + \frac{A_{J/2}}{2}\cos(\pi mj/(J/2)) && (15.38) \end{aligned}$$

So

$$\textit{Inverse Fourier Transform:}\quad U_j = \sum_{m=0}^{J/2} \left(w_m A_m \cos(2\pi mj/J) + B_m \sin(2\pi mj/J) \right)$$

where

$$w_m = \begin{cases} \frac{1}{2} & \text{if } m = 0, J/2, \\ 1 & \text{if } 0 < m < J/2 \end{cases} \tag{15.39}$$

and A_m and B_m are given by:

$$\begin{aligned} \textit{Fourier Transform:}\quad A_m &= 2a_m^r = \frac{2}{J}\sum_{j=0}^{J} w_j U_j \cos(2\pi mj/J) \\ B_m &= -2a_m^i = \frac{2}{J}\sum_{j=1}^{J-1} U_j \sin(2\pi mj/J) \end{aligned}$$

Final remark on counting. It is important to realize that the DFT is simply a change of bases. In the case of complex U_j it is a change of bases in C^J. In the case of real U_j, our case, it is equivalent to a change of bases in R^J. That the counting is correct for this real case (so called real-to-complex-transform) can be seen by the as follows. Referring to (15.25), given J real values U_j, $j = 0, 1, \ldots, J-1$, we obtain J complex amplitudes a_m, $m = -J/2+1, -J/2+2, \ldots, J/2$. However, there is a symmetry in the a_m, namely $a_{-m} = a_m^*$. Hence the negative half of the spectrum: $a_{-1}, a_{-2}, \ldots, a_{-J/2+1}$ is known given the positive half and thus is should not be counted. This leaves us with the $J/2+1$ values $a_0, a_1, \ldots, a_{J/2}$. However, a_0 is necessarily real because $a_0^* = a_{-0} = a_0$. Similarly $a_{J/2}$ is necessarily real (try to show this). Hence there are precisely J distinct real quantities in the a_m $m = -J/2+1, -J/2+2, \ldots, J/2$. These are $a_0^r, a_1^r, a_1^i, \ldots, a_{J/2-1}^r, a_{J/2-1}^i, a_{J/2}^r$.

15.7 The Fast Fourier Transform (FFT)

The fast Fourier transform (FFT) is technique for performing discrete Fourier transforms in a computationally efficient way. Recall the discrete Fourier transform (DFT):

$$a_m = \frac{1}{J} \sum_{j=0}^{J-1} U_j e^{-i2\pi mj/J}, \quad m = 0, 1, \ldots, J-1. \tag{15.40}$$

where for simplicity here we consider the case $m = 0, 1, \ldots, J-1$. We shall only treat the forward transform, the inverse transform can be treated identically.

Let us do two things. First we replace J by N, because N is the usual notation for the size of the Fourier transform. Also let us drop the factor of $1/J = 1/N$ from the definition of a_m. In fact this normalization factor is drop from most, if not all, FFT packages. More on this point later.

Then we write our DFT as:

$$a_m = \sum_{j=0}^{N-1} T_{mj}\, U_j \tag{15.41}$$

where

$$T_{mj} \equiv e^{-i2\pi mj/N} = \left[e^{-i2\pi/N}\right]^{mj}.$$

is a $N \times N$ complex (in fact unitary) matrix.

From this it is clear that performing the DFT from U_j's to a_m's directly from the definition takes $O(N^2)$ work.

(Multiplying an $N \times N$ matrix by a N vector takes N^2 multiplications and additions. As written this would appear to be N^2 complex times real multiplications = $2N^2$ real multiplications. However, by the symmetry of the a_m's the range of m need only be from $m = 0$ to $m = N/2$ and careful counting gives this to be equivalent to N^2 real multiplications.)

In general a change of bases in R^N requires $O(N^2)$ work.

However, in the case of $N = 2^p$ for some p, the DFT can actually be done in $O(N \log_2 N)$ work. In fact N can be much more general, but the algorithms are much more efficient in the case of N the product of powers of small prime factors.

To see how this is possible, decompose the Fourier transform as:

$$\begin{aligned}
a_m &= \sum_{j=0}^{N-1} e^{-i2\pi mj/N} U_j & (15.42)\\
&= \sum_{j=0}^{N/2-1} e^{-i2\pi m(2j)/N} U_{2j} + \sum_{j=0}^{N/2-1} e^{-i2\pi m(2j+1)/N} U_{2j+1} & (15.43)\\
&= \sum_{j=0}^{N/2-1} e^{-i2\pi mj/(N/2)} U_{2j} + e^{-i2\pi m/N} \sum_{j=0}^{N/2-1} e^{-i2\pi mj/(N/2)} U_{2j+1} & (15.44)\\
&= \sum_{j=0}^{N/2-1} e^{-i2\pi mj/(N/2)} U_{2j} + W^m \sum_{j=0}^{N/2-1} e^{-i2\pi mj/(N/2)} U_{2j+1} & (15.45)
\end{aligned}$$

where $W \equiv e^{-i2\pi/N}$.

In this way we have obtained:

$$a_m = (FT\ \ even\ \ indices\ \ of\ \ U) + W^m\,(FT\ \ odd\ \ indices\ \ of\ \ U) \tag{15.46}$$

This decomposition can now be applied recursively. Schematically:

Summary

- Work in doing a FFT for vectors of length $N = 2^p$ is $N \log_2 N$.
- The FFT algorithm works on recursion, effectively transforming a full $N \times N$ matrix (with structure) into the product of sparse matrices.
- For small and moderate N (the exact value depends on many things) the direct approach with an efficient matrix-vector multiplication is more efficient than the “fast” approach.

Chapter 16

APPENDIX - Galerkin and Pseudospectral methods

In this appendix I consider an example showing the difference between Galerkin and pseudospectral approximations.

16.1 Introduction

Consider that we want to solve numerically:

$$\frac{\partial u}{\partial t}(x,t) = \mathcal{L}u(x,t) + g(x,t) \tag{16.1}$$

on some interval $[0, \ell]$ using a spectral approximation:

$$u(x,t) \simeq U(x,t) = \sum_{m=0}^{M} a_m(t)\phi_m(x) \tag{16.2}$$

where the $\phi_m(x)$ are the expansion modes (sines, cosines, complex exponentials, Chebyshev polynomials etc.) We shall work in the semi-discrete (exact time) approximation for most of this chapter.

The following is a central issue for spectral methods:

In general, it is not possible to choose the $a_m(t)$ such that $U(x,t)$ will be an exact solution to (16.1), that is no matter how we choose $a_m(t)$ we cannot exactly satisfy (16.1). We then must decide how to choose the $a_m(t)$ to "best" satisfy (16.1).

First it is intuitively obvious that we cannot expect to exactly satisfy (16.1) with a finite expansion in modes $\phi_m(x)$ Substituting $U(x,t)$ into the left-hand-side of (16.1) gives us a function of (x,t), $\frac{\partial U}{\partial t}(x,t)$, as does substituting on the right-hand-side: $\mathcal{L}U(x,t) + g(x,t)$. Consider some arbitrary fixed time, t_1. In general, we cannot expect that by adjusting M coefficients $a_m(t_1)$ we can get these two functions to be identical for all x in $[0, \ell]$.

Hence substituting our spectral approximation (16.2) into (16.1) we have:

$$\frac{\partial U}{\partial t}(x,t) = \mathcal{L}U(x,t) + g(x,t) + R(x,t) \tag{16.3}$$

where R is the residual.

Different choices for the a_m will give different residuals. Or the other way around, different choices of the residual will dictate different choices for the a_m and hence different choices for the numerical approximation $U(x,t)$.

There are 3 common choices for conditions on R and hence 3 common choices for selecting the a_m. These result in so-called Galerkin, tau, and pseudospectral methods. We will not consider the tau method as it applies to the case in which the bases functions $\phi_m(x)$ do not satisfy the boundary conditions for the problem and we will only consider cases in which the $\phi_m(x)$ individually satisfy the boundary conditions. Primarily will are interested in the pseudospectral method, but we will also consider the Galerkin approach as it too as wide application.

16.1.1 An Example

It is useful to consider an exactly soluble example. Consider the following partial differential equation:

$$\text{PDE:}\quad \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + g(x) = \frac{\partial^2 u}{\partial x^2} + x(1-x) \tag{16.4}$$

$$\text{BC:}\quad u(0,t) = 0 \quad\text{and}\quad u(1,t) = 0, \tag{16.5}$$

$$\text{IC:}\quad u(x,0) = u^0(x) = \frac{1}{2}\sin(\pi x) \tag{16.6}$$

so the equation has an inhomogeneous term is $g(x) = x(1-x)$. Solving this partial differential equationusing the usual separation of variables technique $u(x,t) = \sum_{m=1}^{\infty} a_m(t)\sin(\pi m x)$, one obtains the following ordinary differential equations for the $a_m(t)$:

$$\dot{a}_m = -\pi^2 m^2 a_m + \hat{g}_m \tag{16.7}$$

where the $\hat{g}_m$ are the coefficients of the sine transform of $g(x)$

$$\hat{g}_m \equiv 2\int_0^1 g(x)\sin(\pi m x)dx. \tag{16.8}$$

A simple calculation shows that $\hat{g}_m$ is nonzero for all odd m.

The solution of (16.7) is:

$$a_m(t) = a_m^0 e^{-\pi^2 m^2 t} + \frac{\hat{g}_m}{\pi^2 m^2}\left(1 - e^{-\pi^2 m^2 t}\right) \tag{16.9}$$

where for the initial condition (16.6) $a_1^0 = 1$ and all other a_m^0 are zero.

Note that we use the notation a_m for both the amplitudes in the exact solutions here and the amplitudes in the spectral approximation (16.2). For any given m, the two do not necessarily have the same value. The meaning should be clear from the context.

16.2 Galerkin Approximation

In the Galerkin approximation, we choose the a_m's in the spectral expansion $U(x,t) = \sum_{m=0}^{M} a_m(t)\phi_m(x)$ such that the residual is orthogonal to each of the $M+1$ modes $\phi_m(x)$:

$$\int_0^\ell \phi_m(x) R(x,t) dx = 0, \quad m = 0, 1, \ldots, M \tag{16.10}$$

$$\int_0^\ell \phi_m(x) \left\{ \frac{\partial U}{\partial t}(x,t) - \mathcal{L}U(x,t) - g(x,t) \right\} dx = 0, \quad m = 0, 1, \ldots, M \tag{16.11}$$

Intuitively, at each time we now have $M+1$ equations to satisfy with the $M+1$ coefficients a_m. Hence this should uniquely determine these coefficients.

Let us see what this implies for our example problem. First we shall seek a spectral approximation with $\phi_m(x) = \sin(\pi m x)$:

$$U(x,t) = \sum_{m=1}^{M} a_m(t)\phi_m(x) = \sum_{m=1}^{M} a_m(t)\sin(\pi m x) \tag{16.12}$$

where we start the sum at $m = 1$ because $\phi_0 = 0$. Of course this expansion only differs from the separation of variables solution in that here the sum is finite. Then (16.11) becomes:

$$\int_0^1 \sin(\pi m x) \left\{ \frac{\partial U}{\partial t}(x,t) - \frac{\partial^2 U}{\partial x^2}(x,t) - g(x,t) \right\} dx = 0, \quad m = 1, 2, \ldots, M \tag{16.13}$$

Substituting (16.12) into (16.13) and using the orthogonality of the sines:

$$\dot{a}_m = -\pi^2 m^2 a_m + \hat{g}_m \quad m = 1, 2, \ldots, M \tag{16.14}$$

where $\hat{g}_m$ is the continuous sine transform of $g(x)$ as defined in (16.8) except that here the range of m is finite.

Additional points:

- In the Galerkin approach, one really thinks in terms of the amplitudes a_m.
- For a "real" problem, the resulting ODEs for the a_m would need to be solved numerically.
- Galerkin projections of PDEs onto small number of modes are commonly used in many areas as a way of developing simplified models. The most famous example is the Lorenz equations giving a three-variable model for the weather.

16.3 Pseudospectral Approximation

The pseudospectral (PS) approximation is also known as collocation. In this case we begin by considering a spatial grid x_j, $0 \le j \le J$ in addition to our expansion for $U(x,t)$. These mesh points are called the collocation points. For an expansion in $M+1$ modes, $\phi_m(x)$, $m = 0, 1, \ldots, M$, we would also choose $M+1$ mesh points, i.e. $J = M$. In the pseudospectral approximation we impose the following condition on the residual:

$$\int_0^\ell \delta(x - x_j) R(x,t) dx = 0, \quad j = 0, 1, \ldots, J \tag{16.15}$$

so that $R(x_j, t) = 0$. This gives:

$$\frac{\partial U}{\partial t}(x_j,t) = \mathcal{L}U(x_j,t) + g(x_j,t) \quad j = 0, 1, \ldots, J \tag{16.16}$$

Hence in the pseudospectral approximation the PDE is exactly satisfied on the mesh (collocation) points.

Note that this does not mean that the numerical solution U is exactly equal to the exact solution u on the grid. Because the PDE is not satisfied exactly away from the grid points, the numerical and exact solutions will in general be different everywhere, including on the grid.

Let us consider the pseudospectral approximation for our example problem. Let the mesh points be given by $x_j = jh$ with $h = \ell/J = 1/J$. In this case we can exclude $j = 0$ and $j = J$ as they correspond to the boundaries where $u = 0$ by the boundary conditions. Hence we shall consider only the $J-1$ interior mesh points and likewise we consider the spectral expansion with $M-1$ modes:

$$U(x,t) = \sum_{m=1}^{M-1} a_m(t)\phi_m(x) = \sum_{m=1}^{M-1} a_m(t)\sin(\pi m x) \tag{16.17}$$

(This choice of limits is made because it is that used in defining discrete sine transforms. This is not a relevant difference between Galerkin and PS methods as we could always redefine M here.)

The PS condition on the residual gives us:

$$\frac{\partial U}{\partial t}(x_j,t) = \frac{\partial^2 U}{\partial x^2}(x_j,t) + g(x_j,t) \quad j = 1, 2 \ldots, J-1 \tag{16.18}$$

To see what this implies for the a_m, substitute (16.17) into this equation and solve for the resulting $\dot{a}_m$. Note that:

$$\sin(\pi m x_j) = \sin(\pi m j h) = \sin(\pi m j/J)$$

Hence to solve for $\dot{a}_m$, multiplying both sides of the equation by $\sin(\pi m j/J)$, sum over j, and use

$$\sum_{j=1}^{J-1} \sin(\frac{\pi m j}{J}) \sin(\frac{\pi m' j}{J}) = \frac{J}{2}\delta_{mm'}$$

The result is:

$$\dot{a}_m = -\pi^2 m^2 a_m + \frac{2}{J}\sum_{j=1}^{J-1} g_j \sin(\pi m j/J) \tag{16.19}$$

where $g_j = g(x_j)$. Or

$$\dot{a}_m = -\pi^2 m^2 a_m + \tilde{g}_m \tag{16.20}$$

where

$$\tilde{g}_m \equiv \frac{2}{J}\sum_{j=1}^{J-1} g_j \sin(\pi m j/J) \tag{16.21}$$

So that $\tilde{g}_m$ is the discrete sine transform of g_j, the values of g evaluated on the grid.

As for the Galerkin approximation, for a "real" problem, the resulting ODEs for the a_m would need to be solved numerically.

16.4 Discussion

Refer back to the equations for the amplitudes for the exact solution Eq. (16.7), and the Galerkin and pseudospectral approximations Eqs. (16.14) and (16.20).

For both the exact case and Galerkin approximation, the same coefficients appear, namely $\hat{g}_m$ the *continuous* sine transform of the function $g(x)$. The difference between the exact and Galerkin equations is that the Galerkin approximation contains only $M-1$ (non-zero) modes whereas the exact solution contains infinitely many modes. In the pseudospectral approximation, the equations for $\dot{a}_m$ contain instead $\tilde{g}_m$, the *discrete* sine transform of the function g evaluated on the collocation grid. Hence the PS approximation differs from the exact equations both in that there are a finite number of modes and in that the equations for the included modes are slightly different from the equations for the modes in the exact case.

The difference between the truncated exact equations and the PS equations is known as the **aliasing error**.

In this example, the aliasing error is the difference between $\tilde{g}_m$ and $\hat{g}_m$. For any g appearing in PDE (16.4) it can be shown that:

$$\tilde{g}_m = \hat{g}_m + \sum_{k=1}^{\infty} \hat{g}_{m+2kM} - \hat{g}_{2kM-m} \tag{16.22}$$

16.4.1 So why use PS approximation

Given the aliasing errors that occur with pseudospectral methods, why use them? The answer is that in most cases the aliasing error is quite small and pseudospectral methods are very flexible and can be applied readily to PDEs with inhomogeneities and nonlinearities. In the pseudospectral approach we have both a spatial grid (collocation points) and a spectral expansion, and both are used. The numerical solution can either be represented by the amplitudes a_m or by the values on the grid $U_j = U(x_j)$. Operations such as differentiation are best carried out in spectral space (because the derivative operator is local in spectral space and the modes $\phi_m(x)$ can be differentiated exactly) whereas complications arising from inhomogeneities and nonlinearities are best handled in physical space where they are local.

Chapter 17

APPENDIX - Nonlinearity

17.1 General Remarks

Nonlinear PDEs arise in many areas of application: fluid flow, combustion, biological patterns, etc.

Nonlinearity is one of the primary reasons that numerical solutions of PDEs are necessary.

Consider two examples of nonlinear PDEs:

- Nonlinear reaction-diffusion equations:
$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(u) \tag{17.1}$$
where $f(u)$ represents the "chemical reaction" terms and does not involve any derivatives of u. $\frac{\partial^2 u}{\partial x^2}$ represents diffusion. A simple example of a reaction term with nonlinearity is $f(u) = u - u^3$.
- Burgers equation (1D fluid):
$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + a\frac{\partial^2 u}{\partial x^2} \tag{17.2}$$
where $-u\frac{\partial u}{\partial x}$ represents nonlinear "self-advection" present in fluid flow and $\frac{\partial^2 u}{\partial x^2}$ represents diffusion due to fluid viscosity.

In the first case the nonlinearity does not involve derivatives of u. Such nonlinearities are generally the easiest to deal with. In the second case the nonlinearity involves derivatives.

17.1.1 Difficulties

Some of the difficulties encounter with nonlinear equations are:

- Implicit methods are now much more difficult. Consider time stepping
$$\frac{\partial u}{\partial t} = F(u)$$

by implicit Euler method where $F()$ is a nonlinear operator. Then:
$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\triangle t} = \mathbf{F}(\mathbf{U}^{n+1}) \Rightarrow (\mathbf{I} - \triangle t\mathbf{F})(\mathbf{U}^{n+1}) = \mathbf{U}^n.$$
Here $(\mathbf{I} - \triangle t\mathbf{F})$ is a nonlinear operator. Inverting this operator is not simply a matrix inversion. Consider for example the two nonlinear equations just discussed.
- Nonlinearity can give rise to shocks and other singularities. As a result, relatively smooth initial conditions evolve into solutions requiring very fine spatial resolution. For example, the equation:
$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} \tag{17.3}$$
develops shocks.

  In Burgers equation viscosity ($a\frac{\partial^2 u}{\partial x^2}$) prevents shocks from becoming infinite, but solutions still become very sharp.
- Testing is difficult because exact solutions are rare.

17.1.2 Possible methods

The explicit Euler method can always be used:
$$\mathbf{U}^{n+1} = \mathbf{U}^n + \triangle t\mathbf{F}(\mathbf{U}^n).$$

While this method is not favored because it is low-order accurate in time ($O(\triangle t)$), and because of the stability constraint on the time step, this is nevertheless a good method for getting results quickly in a few cases. It is also acceptable if one is not too concerned about computation time.

From the first remark above it is evident that methods such as Crank-Nicolson are not a viable option for treating nonlinearities in PDEs. Generally some sacrifices must be made in terms of order of accuracy and/or stability.

We will consider one of the most common methods for treating nonlinear terms in PDEs. The nonlinear terms are treated explicitly, but more accurately than with the forward Euler method.

17.2 Multistep Methods for Nonlinear ODEs

Consider **multistep methods** for solving the nonlinear ODE:
$$\frac{du}{dt} = f(u).$$

These schemes advance solution for time t to time $t + \triangle t$ using not only $f(U^n)$ and (possibly) $f(U^{n+1})$, but also $f(U^{n-1})$, $f(U^{n-2})$, ...

Generally:
$$U^{n+1} = U^n + \triangle t\sum_{i=0}^{k} \beta_i f(U^{n+1-i})$$

If $\beta_0 = 0$ then $f(U^{n+1})$ is not used and the method is *explicit*. Of these the **Adams-Bashforth** schemes are the most common. For these the β_i are chosen such that the method is exact if $f(u(t))$ is a polynomial of degree $(k-1)$ in t. The schemes have one-step errors which are $O(\triangle t^{k+1})$.

Some cases are:

$$k = 1, \qquad \beta_1 = 1 \qquad \text{explicit Euler} \tag{17.4}$$

$$k = 2, \qquad \beta_1 = \tfrac{3}{2}, \beta_2 = -\tfrac{1}{2} \qquad \text{2nd order Adams} - \text{Bashforth} \tag{17.5}$$

$$k = 4, \quad \beta_1 = \tfrac{55}{24},\ \beta_2 = -\tfrac{59}{24},\ \beta_3 = \tfrac{37}{24},\ \beta_4 = -\tfrac{9}{24} \qquad \text{4th order Adams} - \text{Bashforth} \tag{17.6}$$

In detail the second order Adams-Bashforth scheme is:

$$U^{n+1} = U^n + \frac{\triangle t}{2}\left\{3f(U^n) - f(U^{n-1})\right\}$$

The one-step error is $O(\triangle t^3)$. As usual, the global discretization error is one power of $\triangle t$ smaller than the one-step error, so $E^f = O(\triangle t^2)$).

Comments:

- The Adams-Bashforth methods give higher-order accuracy in time and yet are explicit.
- Numerical stability is an issue and for large k the time step restriction is too server for the schemes to be practical.
- By saving previous values of $f(U^n)$, one need only evaluate f once per time step. This is significant for the case of PDEs and is a significant advantage over other high-order methods such as Runge-Kutta in which multiple function evaluations are required at each time step.
- The first time step(s) cannot be multistep. Some number of steps must be made at lower-order accuracy or using another method, such as Runge-Kutta. When using second-order Adams-Bashforth, it is common to take one explicit Euler step at the beginning. This has no importance for the global discritizaton error.

17.3 Nonlinear PDEs

Write the PDE as:

$$\frac{\partial u}{\partial t} = \mathcal{L} \cdot u + \mathcal{N}(u)$$

where $\mathcal{L}$ and $\mathcal{N}$ are linear and nonlinear operators respectively.

One can then treat the linear operator using Crank-Nicolson as before and treat the nonlinear operator using 2nd order Adams-Bashforth. This has the advantage of achieving overall $O(\triangle t^2)$ accuracy and allowing reasonable size for $\triangle t$. In detail:

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\triangle t} = \frac{1}{2}\left\{\mathbf{L}\mathbf{U}^{n+1} + \mathbf{L}\mathbf{U}^n\right\} + \frac{1}{2}\left\{3\mathbf{N}(\mathbf{U}^n) - \mathbf{N}(\mathbf{U}^{n-1})\right\}$$

where $N_j(\mathbf{U}^n)$ is the numerical approximation to $\mathcal{N}(u)$ at grid point j.

Solving for $\mathbf{U}^{n+1}$ we obtain:

$$\mathbf{U}^{n+1} = (\mathbf{I} - \triangle t\mathbf{L})^{-1}\left\{(\mathbf{I} + \triangle t\mathbf{L})\mathbf{U}^n + \frac{\triangle t}{2}\left(3\mathbf{N}(\mathbf{U}^n) - \mathbf{N}(\mathbf{U}^{n-1})\right)\right\}$$

With boundary conditions properly accounted for and with the possibility of an inhomogeneous term in the equation, this can be written:

$$\boxed{\mathbf{U}^{n+1} = (\mathbf{I} - \frac{\triangle t}{2}\mathbf{L}^{n+1})^{-1}\ \mathbf{B}\ \left\{(\mathbf{I} + \frac{\triangle t}{2}\mathbf{L}^n)\ \mathbf{U}^n + \frac{\triangle t}{2}\left(3\mathbf{N}(\mathbf{U}^n) - \mathbf{N}(\mathbf{U}^{n-1})\right) + \frac{\triangle t}{2}(\mathbf{g}^n + \mathbf{g}^{n+1})\right\}}$$

where $\mathbf{B}$ is the boundary condition operator appropriate to the type of BC applied.

The contribution to the global error E^f from time discretization if $O(\triangle t^2)$.

The scheme is not unconditionally stable. There is a maximum $\triangle t$, determined from $\mathbf{N}$ and $\mathbf{U}$. For many PDEs this restriction is less severe than the restriction $\triangle t/h^2 \leq 1/2$.