Introduction to Finite Differences

Consider the heat equation on a finite interval subject to Dirichlet boundary conditions and arbitrary (i.e. user specified) initial conditions:

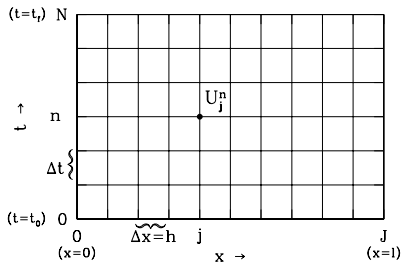
$$\begin{aligned} \text{PDE:} \quad & \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 \le x \le \ell, \quad t > 0 \\ \text{BC:} \quad & u(0,t) = \gamma_0 \quad \text{and} \quad u(\ell,t) = \gamma_\ell, \quad t > 0 \\ \text{IC:} \quad & u(x,0) = u^0(x), \quad 0 \le x \le \ell. \end{aligned}$$

Assume the γ's are time-independent.

Space-time discretization

We need a numerical representation for the function $u(x,t)$ and the for operators $\frac{\partial}{\partial t}$ and $\frac{\partial^2}{\partial x^2}$.

In the simplest case the dependent variable u is represented by values on a uniform grid or lattice in space and time.



- Divide the interval $[0,\ell]$ into J equally spaced intervals of size $\triangle x$ or h. Hence $h = \triangle x = \ell/J$. There are a total of $J+1$ gridpoints labeled x_j. For the uniform grid $x_j = j\triangle x, \;\; j = 0, J$
- n labels timesteps, t_n denotes discrete time values, and $\triangle t$ is the timestep. When needed, t_0 and t_f will denote the initial and final times and N will denote the total number of time steps. Usually $t_0 = 0$ and $t_n = n\triangle t$.

Also we frequently need N to represent other quantities and only seldom do we need a notation for the total number of timesteps.

- U denotes the numerical approximation to u. U_j^n will denote the numerical solution at timestep n and gridpoint j. If the numerical solution exactly agrees with the true solution, then $U_j^n = u(x_j, t_n)$. One goal, though not the only one, of a numerical scheme is to make

$$\lim_{h,\triangle t \to 0} |U_j^n - u(x_j,t_n)| = 0$$

Explicit Euler solution

Basic methodology of finite-difference schemes - approximate the derivatives appearing in the partial differential equation with combinations (differences) of the values on the grid. Here we consider the simplest case.

Approximate time derivative by the *forward difference*:

$$\frac{\partial u}{\partial t}|_{j,n} = \frac{\partial u}{\partial t}(x_j,t_n) \simeq \frac{U_j^{n+1} - U_j^n}{\triangle t}$$

Approximate space-derivative by the second-order *center difference*:

$$\frac{\partial^2 u}{\partial x^2}|_{j,n} = \frac{\partial^2 u}{\partial x^2}(x_j,t_n) \simeq \frac{U_{j-1}^n - 2U_j^n + U_{j+1}^n}{h^2}$$

The heat equation becomes:

$$\frac{U_j^{n+1} - U_j^n}{\triangle t} = \frac{U_{j-1}^n - 2U_j^n + U_{j+1}^n}{h^2}$$

Solving for U_j^{n+1}:

$$U_j^{n+1} = U_j^n + \nu(U_{j-1}^n - 2U_j^n + U_{j+1}^n)$$

where $\nu \equiv \triangle t/h^2$.

This is an explicit expression for the U's at the $n+1st$ timestep in terms of the U's at the nth timestep. Taking into account the BCs and IC we have the explicit, or forward, Euler scheme for the heat equation:

$$\boxed{\begin{aligned} U_j^0 &= u^0(x_j), \quad 0 \le j \le J \\ U_j^{n+1} &= \begin{cases} \gamma_0, & j = 0 \\ U_j^n + \nu(U_{j-1}^n - 2U_j^n + U_{j+1}^n), & 0 < j < J, \\ \gamma_\ell, & j = J \end{cases} \end{aligned}}$$

At each new timestep, the dependent variable at each interior grid point is computed from values at three gridpoints at the preceding timestep.

More on finite-difference formulas

Difference operators

The following difference operators are frequently useful:

- forward difference:

$$\begin{aligned}\Delta_{+t}u(x,t) &\equiv u(x,t+\triangle t)-u(x,t)\\ \Delta_{+x}u(x,t) &\equiv u(x+\triangle x,t)-u(x,t)\end{aligned}$$

- backward difference:

$$\begin{aligned}\Delta_{-t}u(x,t) &\equiv u(x,t)-u(x,t-\triangle t)\\ \Delta_{-x}u(x,t) &\equiv u(x,t)-u(x-\triangle x,t)\end{aligned}$$

- central difference:

$$\begin{aligned}\delta_t u(x,t) &\equiv u(x,t+\frac{1}{2}\triangle t)-u(x,t-\frac{1}{2}\triangle t)\\ \delta_x u(x,t) &\equiv u(x+\frac{1}{2}\triangle x,t)-u(x-\frac{1}{2}\triangle x,t)\end{aligned}$$

  From this follows the second-order central difference:

$$\delta_x^2 u(x,t) \equiv u(x-\triangle x,t)-2u(x,t)+u(x+\triangle x,t)$$

- central difference using double interval:

$$\Delta_{0x}u(x,t) \equiv \frac{1}{2}(\Delta_{+x}+\Delta_{-x})u(x,t)=\frac{1}{2}(u(x+\triangle x,t)-u(x-\triangle x,t))$$

Derivation of finite-difference formulas

Suppose we want to approximate $\frac{\partial^r u}{\partial x^r}$ to some order of accuracy as a weighted sum of values of u on a (possibly non-uniform) grid $\ldots, x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2}, \ldots$:

$$\frac{\partial^r u}{\partial x^r}(x_j) \simeq \sum_k w_k u_{j+k}$$

where $u_{j+k}=u(x_{j+k})$. How are the weights w_k chosen?

The trick is to express all the $u_{j+k}=u(x_{j+k})$ as Taylor series expansions about point x_j:

$$\begin{aligned}u_j &= u_j\\ u_{j+1} &= u_j+\frac{\partial u}{\partial x}|_j(x_{j+1}-x_j)+\frac{\partial^2 u}{\partial x^2}|_j\frac{(x_{j+1}-x_j)^2}{2}+\ldots\\ u_{j-1} &= u_j+\frac{\partial u}{\partial x}|_j(x_{j-1}-x_j)+\frac{\partial^2 u}{\partial x^2}|_j\frac{(x_{j-1}-x_j)^2}{2}+\ldots\\ &\vdots\end{aligned}$$

where $|_j$ means evaluated at x_j. Multiplying each equation by corresponding weight and summing we have:

$$\sum_k w_k u_{j+k}=u_j\sum_k w_k+\frac{\partial u}{\partial x}|_j\sum_k w_k(x_{j+k}-x_j)+\ldots\frac{\partial^r u}{\partial x^r}|_j\sum_k w_k\frac{(x_{j+k}-x_j)^r}{r!}+\ldots$$

The left-hand-side is the sum that we want to approximate $\frac{\partial^r u}{\partial x^r}$. To achieve this for generic functions u, we must have all terms preceding $\frac{\partial^r u}{\partial x^r}$ on the-right-side be zero and the coefficient of $\frac{\partial^r u}{\partial x^r}$ be one. In addition we would like as many terms as possible following $\frac{\partial^r u}{\partial x^r}$ also to be zero. This is satisfied by choosing the weights such that:

$$\begin{aligned}\sum_k w_k &= 0\\ \sum_k w_k(x_{j+k}-x_j) &= 0\\ &\vdots\\ \sum_k w_k\frac{(x_{j+k}-x_j)^r}{r!} &= 1\\ \sum_k w_k\frac{(x_{j+k}-x_j)^{r+1}}{(r+1)!} &= 0\\ &\vdots\end{aligned}$$

These equations are linear in the weights and thus can be solved for the w_k in terms of the differences $x_{j+k}-x_j$.

The number of gridpoints necessary to approximate a derivative depends on the order r of the derivative and the number of high-order terms in the Taylor series that one want to make zero. Frequently on uniform grids symmetries enter such that with q weights it is possible to satisfy more than q conditions.

Error Analysis

The analysis of errors in numerical schemes is important for the following reasons:

- It tells us where errors come from and where we should concentrate efforts to reduce errors.
- It allows comparison of different schemes.
- It provides a powerful basis for testing programs.

Roundoff Error

Roundoff errors arise due to finite precision computations. The one rule with regard to roundoff is:

Where possible avoid subtracting nearly equal numbers.

This rule implies, for example, that one should not take $\tan(x)$ for large x.

Truncation Error

Truncation error $T(x,t)$ is the error in approximating differential operators and PDEs by discrete representations such as finite differences.

For a PDE written as $\mathcal{F}u = 0$,

$$\boxed{T(x,t) = \mathcal{F}^A u(x,t)}$$

where $\mathcal{F}^A$ is the approximation to the differential equation and u is an exact solution to the PDE.

Example: The truncation error for the explicit Euler scheme for the heat equation is

$$\begin{aligned} T(x,t) &= \frac{u(x,t+\triangle t) - u(x,t)}{\triangle t} - \frac{u(x-h,t) - 2u(x,t) + u(x+h,t)}{h^2} \\ &= \frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) + \frac{\triangle t}{2}\frac{\partial^2 u}{\partial t^2}(x,t) - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(x,t) + \dots \\ &= \frac{\triangle t}{2}\frac{\partial^2 u}{\partial t^2}(x,t) - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(x,t) + \dots \end{aligned}$$

so

$$T_{h,\triangle t}(x,t) = O(\triangle t) + O(h^2)$$

We assume $|\frac{\partial^2 u}{\partial t^2}|$ and $|\frac{\partial^4 u}{\partial x^4}|$ are bounded in the space-time domain of interest. The truncation error goes to zero everywhere in the domain as $\triangle t$ and h go to zero. Hence the explicit Euler approximation is **consistent** with the partial differential equation.

Discretization Error

The **discretization error** e_j^n at a point of the computational grid is the difference between the numerical and exact solutions:

$$\boxed{e_j^n \equiv U_j^n - u(x_j, t_n)}$$

where u and U_j^n satisfy the same initial conditions, i.e. $U_j^0 = u(x_j, 0)$.

Frequently we are interested in:

$$\boxed{E^f \equiv \max_j |e_j|, \text{ for fixed final time } t_f}$$

also called the discretization error (though usage varies).

For the explicit Euler solution of the heat equation this can be bounded using the truncation error as long as $\nu \leq 1/2$.

$$E^f \leq t_f \bar{T}.$$

where $\bar{T}$ is a bound on the truncation error: $|T_j^n| \leq \bar{T}$.

$$E^f \to 0, \text{ for } \triangle t, h \to 0 \text{ such that } \nu = \triangle t / h^2 \leq 1/2$$

The numerical solution is said to **converge** to the exact solution.

Our main interest is the scaling of discretization error with $\triangle t$ and h:

$$\boxed{E^f = |O(\triangle t) + O(h^2)|}$$ [1]

We can investigate the final error E^f (at a fixed final time) as a function of $\triangle t$ and h. To make this precise, we need a **refinement path =** $(h_i, \triangle t_i), i = 0, 1, 2, \ldots$ with h_i and $\triangle t_i$ going to zero such that $\nu_i = \triangle t_i / h_i^2 \leq 1/2$. We compute the error E^f at points on the refinement path and verify the scaling of E^f. In practice for testing it is better to vary only one of $\triangle t$ and h while holding the other constant.

The scaling of the discretization error with $\triangle t$ and h provides a valuable check on the correctness of a program.

[1] *Note:* This is slightly subtle in that the discretization error for one-step is 2nd order in $\triangle t$, i.e.

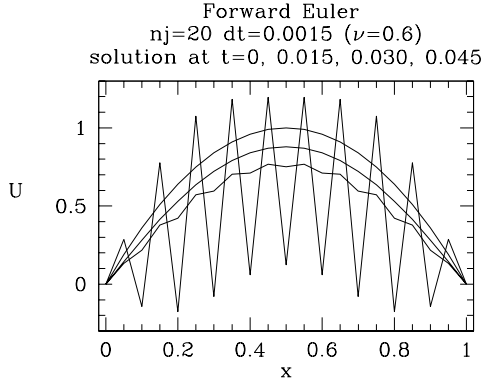
$$E^1 \leq \triangle t |O(\triangle t) + O(h^2)|$$

For *fixed* final time t_f the number of timesteps N necessary to reach t_f increases as $1/\triangle t$ decreases ($N \sim 1/\triangle t$). This cancels the leading $\triangle t$ in the one-step error. Because $N \sim 1/\triangle t$ we cannot consider $E = E^N$ with fixed N but instead we consider a fixed final time and hence the notation E^f.

Stability and Implicit Methods

Motivation

In the analysis of discretization error in the explicit Euler method for the heat equation, we needed to assume $\nu = \triangle t/h^2 \leq 1/2$ in order to obtain bounds on the error. Consider now what happens to the numerical solution using the explicit Euler method when $\nu = \triangle t/h^2 > 1/2$. After a relative small number of timesteps the solution develops a **numerical instability** which grows exponentially with the number of timesteps.



This purpose of this chapter is to understand why this happens and to introduce stable (implicit) methods to eliminate this problem.

ODEs

Most of the issues can be understood from the simplest ODE:

$$\dot{u} = \lambda u,$$

We focus on λ real, but one can also consider complex λ.

Compare the exact solution with a variety of timestepping schemes

- Exact solution:

$$u(t) = u(0)e^{\lambda t}$$

- The explicit or forward Euler timestepping:

$$\frac{U^{n+1} - U^n}{\triangle t} = \lambda U^n \quad \Rightarrow \quad U^{n+1} = (1 + \lambda \triangle t) U^n$$

- The implicit or backward Euler timestepping:

$$\frac{U^{n+1} - U^n}{\triangle t} = \lambda U^{n+1} \quad \Rightarrow \quad U^{n+1} = \frac{1}{1 - \lambda \triangle t} U^n$$

- The Crank-Nicolson scheme is given by an average of the explicit and implicit schemes:

$$\frac{U^{n+1} - U^n}{\triangle t} = \frac{\lambda U^n + \lambda U^{n+1}}{2} \quad \Rightarrow \quad U^{n+1} = \frac{1 + (\triangle t/2)\lambda}{1 - (\triangle t/2)\lambda} U^n$$

- All three schemes can be generalized to an arbitrary weighted average:

$$\frac{U^{n+1} - U^n}{\triangle t} = (1 - \theta)\lambda U^n + \theta \lambda U^{n+1} \quad \Rightarrow \quad U^{n+1} = \frac{1 + (1 - \theta)\lambda \triangle t}{1 - \theta \lambda \triangle t} U^n$$

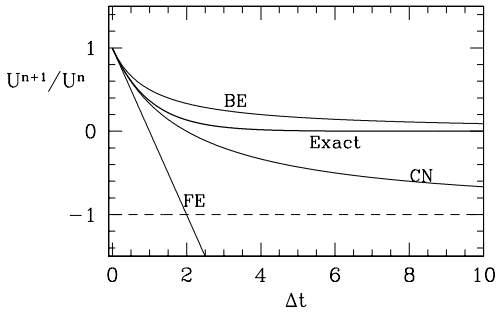
  where $0 \leq \theta \leq 1$. This is called the weighted average or θ-method.

Aside on terminology: **Implicit** timestepping refers to the fact that U^{n+1} appears on the right-hand-side of this equation. Hence it is an implicit equation for U^{n+1}, the value of u at the next timestep, in terms of U^n, the known value at the current timestep. For this simple ODE it is trivial to solve for U^{n+1}, but for general ODES this would not be so.

The terms **forward** and **backward** come about from a slightly different way of viewing these schemes. If the time appearing on the right-hand-side is taken as the reference point, then the derivative on the left-hand-side is a forward difference or a backward difference.

One-step analysis

For each of the three schemes let us consider the dependence of U^{n+1}/U^n on $\triangle t$ and compare this with the exact expression for $u(t + \triangle t)/u(t)$. The results are shown graphically for the case $\lambda = -1$.



We make the following observations.

- All three schemes approximate the exact, exponential, relation for small $\triangle t$.
- The forward Euler seems particularly bad for moderate and large $\triangle t$. Importantly note that $|U^{n+1}| > |U^n|$ for $\triangle t > 2$.
- In many ways the backward Euler scheme appear to be the best because is never gets too far from the exact solution and has the correct limit for large $\triangle t$. Note, however, that for small $\triangle t$ the backward Euler scheme approximates the exact to the same accuracy as forward Euler.
- The Crank-Nicolson scheme is a better approximation to the exact solution for small $\triangle t$ than either of the other two methods. That is the Crank-Nicolson curve is seen to follow the exact solution better at small $\triangle t$. $|U^{n+1}| < |U^n|$ for all $\triangle t$.

For small $\triangle t$ expand each form in Taylor's series to obtain:

$$\begin{aligned}
\textit{Exact:}\ \frac{u(t+\triangle t)}{u(t)} &= e^{\lambda \triangle t} &&= 1 + \lambda\triangle t + \frac{1}{2}\lambda^2\triangle t^2 + \frac{1}{6}\lambda^3\triangle t^3 + \ldots \\
\textit{FE:}\ \frac{U^{n+1}}{U^n} &= 1 + \lambda\triangle t &&= 1 + \lambda\triangle t \\
\textit{BE:}\ \frac{U^{n+1}}{U^n} &= (1 - \lambda\triangle t)^{-1} &&= 1 + \lambda\triangle t + \lambda^2\triangle t^2 + \ldots \\
\textit{CN:}\ \frac{U^{n+1}}{U^n} &= (1 - \lambda\tfrac{\triangle t}{2})^{-1}(1 + \lambda\tfrac{\triangle t}{2}) &&= 1 + \lambda\triangle t + \frac{1}{2}\lambda^2\triangle t^2 + \frac{1}{4}\lambda^3\triangle t^3 + \ldots
\end{aligned}$$

Hence we see that FE and BE agree with Exact to $O(\triangle t)$ and makes an error $O(\triangle t^2)$, whereas CN agrees with Exact to $O(\triangle t^2)$ making an error $O(\triangle t^3)$. Note that these are the single-timestep errors which are one power of $\triangle t$ higher than the final truncation or discretization error.

N-step analysis

Consider now the result of repeatedly timestepping each scheme:

$$U^n = \alpha^n U^0$$

This issue is,

> *is* $|\alpha| \leq 1$ *whenever* $\lambda \leq 0$, i.e. are numerical solutions bounded whenever exact solutions are?

- For BE and CN methods one can easily establish that when $\lambda < 0$, $|\alpha| \leq 1$ for $0 \leq \triangle t < \infty$.
  BE and CN are **unconditionally stable** because there is no requirement on $\triangle t$ for stability.
- For the FE case, however, it is evident that $|\alpha|$ exceeds one and thus $|U^n|$ will diverge as $n \to \infty$. This is the essence of the numerical instability illustrated at the beginning of the chapter.
  The explicit Euler scheme is **conditionally stable** with **stability limit**

$$\boxed{\triangle t \leq \frac{2}{-\lambda}}$$

  Because instability occurs for $\alpha < -1$, it is oscillatory in time with U^{n+1} and U^n of opposite signs. This is one of the key signatures of numerical instabilities.

PDEs

Notation

- $\mathbf{U}^n$ will be the vector representing the numerical solution at timestep n:

$$\mathbf{U}^n = (U_0^n, U_1^n, U_2^n, \ldots, U_j^n, \ldots, U_{J-1}^n, U_J^n)^T$$

- $\mathbf{L}$ will be the matrix representing the finite-difference representation of a linear operator $\mathcal{L}$. For now we consider $\mathcal{L} = \frac{\partial^2}{\partial x^2}$ and

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

- $\mathbf{I}$ will denote the identity matrix.

With this notation we can easily express the 3 timestepping schemes as follows:

$$\boxed{\begin{array}{ll} \text{FE:} & \dfrac{\mathbf{U}^{n+1}-\mathbf{U}^n}{\triangle t} = \mathbf{L}\,\mathbf{U}^n \;\Rightarrow\; \mathbf{U}^{n+1} = (\mathbf{I}+\triangle t\mathbf{L})\,\mathbf{U}^n \\ \text{BE:} & \dfrac{\mathbf{U}^{n+1}-\mathbf{U}^n}{\triangle t} = \mathbf{L}\,\mathbf{U}^{n+1} \;\Rightarrow\; \mathbf{U}^{n+1} = (\mathbf{I}-\triangle t\mathbf{L})^{-1}\,\mathbf{U}^n \\ \text{CN:} & \dfrac{\mathbf{U}^{n+1}-\mathbf{U}^n}{\triangle t} = \dfrac{\mathbf{L}\,\mathbf{U}^n+\mathbf{L}\,\mathbf{U}^{n+1}}{2} \;\Rightarrow\; \mathbf{U}^{n+1} = (\mathbf{I}-\dfrac{\triangle t}{2}\mathbf{L})^{-1}\,(\mathbf{I}+\dfrac{\triangle t}{2}\mathbf{L})\,\mathbf{U}^n \end{array}}$$

All the above timestepping schemes are of the form:

$$\mathbf{U}^{n+1} = \mathbf{A}_-^{-1}\,\mathbf{A}_+\,\mathbf{U}^n = \mathbf{A}\,\mathbf{U}^n.$$

where

$$\mathbf{A}_+ \equiv (\mathbf{I}+(1-\theta)\triangle t\mathbf{L}) \qquad \mathbf{A}_- \equiv (\mathbf{I}-\theta\triangle t\mathbf{L}).$$

Stability analysis

All the above timestepping schemes are of the form:

$$\mathbf{U}^{n+1} = \mathbf{A}\,\mathbf{U}^n.$$

The stability analysis is reduced to determining whether the matrix $\mathbf{A}$ has any eigenvalues such that $|\alpha| > 1$.

The **von Neumann** or **Fourier** method of stability analysis is based on *choosing* the eigenvectors of $\mathbf{A}$ to be trigonometric:

$$\phi_j = \cos(\beta j h)$$

or

$$\phi_j = e^{i\beta jh}$$

whichever is easier to work with. β is continuous and can take on any value. In this approach the corresponding values of α are simply computed.

The eigenvalues α of $\mathbf{A}$ are related to the eigenvalues λ of $\mathbf{L}$:

$$\begin{aligned} \textit{FE:} \quad \alpha &= 1 + \lambda\triangle t \\ \textit{BE:} \quad \alpha &= (1 - \lambda\triangle t)^{-1} \\ \textit{CN:} \quad \alpha &= (1 - \lambda\frac{\triangle t}{2})^{-1}(1 + \lambda\frac{\triangle t}{2}) \end{aligned}$$

Find the eigenvalue of $\mathbf{L}$:

$$\begin{aligned} \lambda\phi_j &= \frac{1}{h^2}(\phi_{j-1} - 2\phi_j + \phi_{j+1}) \\ \lambda e^{i\beta jh} &= \frac{1}{h^2}(e^{i\beta(j-1)h} - 2e^{i\beta jh} + e^{i\beta(j+1)h}) \\ \lambda &= \frac{1}{h^2}(e^{-i\beta h} - 2 + e^{i\beta h}) \\ \lambda &= \frac{2}{h^2}(\cos\beta h - 1) \end{aligned}$$

The eigenvalues of $\mathbf{A}$ for the forward Euler method are:

$$\alpha = 1 + \triangle t\lambda = 1 + \frac{2\triangle t}{h^2}(\cos\beta h - 1)$$

Note that $(\cos\beta h - 1) \leq 0$. Hence the only possibility for instability occurs for $\alpha < -1$. For any $\triangle t$, the minimum (most negative) value of α occurs for $\beta h = \pi$ giving:

$$\alpha_{min} = 1 - \frac{4\triangle t}{h^2}$$

For explicit Euler scheme to be stable, α_{min} must be greater than -1. This is equivalent to the condition

$$\boxed{\triangle t \leq \frac{h^2}{2} \quad \text{or} \quad \nu = \frac{\triangle t}{h^2} \leq \frac{1}{2}}$$

- The explicit Euler scheme is *conditionally stable*.
- The reader can easily show that the BE and CN schemes are *unconditionally stable*.

Form of the numerical instability. As before the instability occurs for $\alpha < -1$ and hence when instability develops it is oscillatory in time with U_j^{n+1} and U_j^n of opposite signs. However, now there is the additional dependence on space given by the eigenvector $\phi_j = \cos(\beta jh)$ (now I use the cosine form). The instability arises first from the eigenvector with $\beta h = \pi$ or $\phi_j = \cos(\pi j) = (-1)^j$. This normal mode changes sign in space at every other gridpoint. Hence when the instability develops it is oscillatory in space as well as time. Again this is one of the key signatures of a numerical instability.

Boundary Conditions

We shall consider the following boundary conditions on the interval $[0, \ell]$:

$$\begin{aligned} \alpha_0(t)u(0,t) + \beta_0(t)\frac{\partial u}{\partial x}(0,t) &= \gamma_0(t), \quad \alpha_0, \beta_0 \geq 0 \\ \alpha_\ell(t)u(l,t) + \beta_\ell(t)\frac{\partial u}{\partial x}(l,t) &= \gamma_\ell(t), \quad \alpha_\ell, \beta_\ell \geq 0 \end{aligned}$$

These boundary conditions are Dirichlet if $\beta = 0$ and Neumann if $\alpha = 0$ and Robin if both α and β are nonzero.

We consider methods for dealing with these cases which have the advantage that the same $J + 1$ spatial points x_j with $x_0 = 0$ and $x_J = \ell$ are used in each case.

Computationally for (1) and (1) there are two distinct cases: $\beta = 0$ and $\beta \neq 0$.

Dirichlet case: $\beta = 0$

We consider in detail the left boundary only, the right boundary follows similarly.

Without loss of generality we can write the left BC as:

$$u(0,t) = \gamma_0(t)$$

since α_0 cannot be zero if β_0 is.

Eq. () dictates the numerical solution at $x = 0$ for all time: $U_0^n = \gamma_0^n \equiv \gamma_0(n\triangle t)$. Thus U_0^n does not need to be computed from the PDE and and we need not (and cannot) impose the PDE at $j = 0$.

The following provides a simple implementation. Consider the heat equation and let $\mathbf{L}$ be the discretization of the second derivative. Put zeros in the rows of matrix $\mathbf{L}$ at Dirichlet boundary points

$$\mathbf{L} = \frac{1}{h^2}\begin{bmatrix} 0 & 0 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & & \ddots & \\ & & & & \end{bmatrix}$$

So $[\mathbf{LU}]_0 = 0$, $[\mathbf{LU}]_1 = (U_0 - 2U_1 + U_2)/h^2$. You may think of this as not imposing the PDE at the boundary.

Next define the boundary operator $\mathbf{B}^n$ by:

$$\mathbf{B}^n\ \mathbf{U} = \mathbf{B}^n \begin{pmatrix} U_0 \\ U_1 \\ \\ \cdots \end{pmatrix} = \begin{pmatrix} \gamma_0^n \\ U_1 \\ \\ \cdots \end{pmatrix}$$

$\mathbf{B}^n$ sets the boundary values to the corresponding BC at time $t = n\triangle t$. Note: $\mathbf{B}$ is not a linear operator if $\gamma_0 \neq 0$.

Using this operator the three timestepping schemes can be written:

FE:	$\mathbf{U}^{n+1} = \mathbf{B}^{n+1}\ (\mathbf{I} + \triangle t\mathbf{L})\ \mathbf{U}^n$
BE:	$\mathbf{U}^{n+1} = (\mathbf{I} - \triangle t\mathbf{L})^{-1}\ \mathbf{B}^{n+1}\ \mathbf{U}^n$
CN:	$\mathbf{U}^{n+1} = (\mathbf{I} - \frac{\triangle t}{2}\mathbf{L})^{-1}\ \mathbf{B}^{n+1}\ (\mathbf{I} + \frac{\triangle t}{2}\mathbf{L})\ \mathbf{U}^n$

Neumann/Robin case: $\beta \neq 0$

Without loss of generality we can write the left BC as:

$$\frac{\partial u}{\partial x}(0,t) = \alpha_0(t)u(0,t) + \gamma_0(t)$$

since $\beta_0 \neq 0$.

In this case we do not know U_0^n directly from the boundary condition, and must find U_0^n by imposing the PDE at the boundary.

However, to use the second-order centered-difference approximation for $\frac{\partial^2 u}{\partial x^2}$ at grid point $j = 0$ would require "U_{-1}", i.e.

$$\frac{\delta_x^2}{h^2}U_0 = \frac{1}{h^2}(U_{-1} - 2U_0 + U_1)$$

and U_{-1} does not exist. Here we consider a heuristic approach in which () is used where U_{-1} is obtained from boundary condition ().

To obtain U_{-1} from the BC use the second-order centered-difference representation for the first derivative at the boundary:

$$\frac{\partial u}{\partial x}(0,t) = \frac{1}{2h}(U_1 - U_{-1}) + O(h^2)$$

Plugging this into the boundary condition () gives:

$$\frac{1}{2h}(U_1 - U_{-1}) = \alpha_0 U_0 + \gamma_0$$

Solving for U_{-1}:

$$U_{-1} = U_1 - 2h\alpha_0 U_0 - 2h\gamma_0$$

Then

$$\begin{aligned}\frac{\delta_x^2}{h^2}U_0 &= \frac{1}{h^2}(U_1 - 2h\alpha_0 U_0 - 2h\gamma_0 - 2U_0 + U_1) \\ &= \frac{1}{h^2}\left(-2(1+h\alpha_0^n)U_0^n + 2U_1^n\right) - \frac{2}{h}\gamma_0^n\end{aligned}$$

We define $\mathbf{L}^n$ by:

$$\mathbf{L}^n = \frac{1}{h^2}\begin{bmatrix} -2(1+h\alpha_0^n) & 2 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & & \ddots & \\ & & & & \end{bmatrix}$$

and define the boundary operator $\mathbf{B}^n$ in this case to be:

$$\mathbf{B}^n\ \mathbf{U} = \mathbf{B}^n \begin{pmatrix} U_0 \\ U_1 \\ \cdots \\ \end{pmatrix} = \begin{pmatrix} U_0 \\ U_1 \\ \cdots \\ \end{pmatrix} + \begin{pmatrix} \frac{-2\triangle t}{h}\gamma_0^n \\ 0 \\ \cdots \\ \end{pmatrix}$$

where $\alpha_0^n \equiv \alpha_0(n\triangle t)$, and as before $\gamma_0^n = \gamma_0(n\triangle t)$. $\mathbf{L}^n$ and $\mathbf{B}^n$ depend on n (time) if α and γ do.

The three timestepping schemes are then:

FE:	$\mathbf{U}^{n+1} = \mathbf{B}^n\ (\mathbf{I} + \triangle t \mathbf{L}^n)\ \mathbf{U}^n$
BE:	$\mathbf{U}^{n+1} = (\mathbf{I} - \triangle t \mathbf{L}^{n+1})^{-1}\ \mathbf{B}^{n+1}\ \mathbf{U}^n$
CN:	$\mathbf{U}^{n+1} = (\mathbf{I} - \frac{\triangle t}{2}\mathbf{L}^{n+1})^{-1}\ \mathbf{B}^{n+1/2}\ (\mathbf{I} + \frac{\triangle t}{2}\mathbf{L}^n)\ \mathbf{U}^n$

where $\mathbf{B}^{n+1/2}$ is defined by:

$$\mathbf{B}^{n+1/2}\ \mathbf{U} = \frac{1}{2}(\mathbf{B}^{n+1} + \mathbf{B}^n)\ \mathbf{U} = \begin{pmatrix} U_0 \\ U_1 \\ \cdots \\ \end{pmatrix} + \begin{pmatrix} \frac{-\triangle t}{h}(\gamma_0^n + \gamma_0^{n+1}) \\ 0 \\ \cdots \\ \end{pmatrix}$$

We will frequently write $\mathbf{B}$ for the appropriately defined boundary condition operator. The forms of this operator, and the timestep n at which it is evaluated depends on the boundary conditions and the method under consideration.

Periodic BCs

Consider periodic boundary conditions imposed on the grid with $x_0 = 0$ and $x_J = \ell$. Then $u(0,t) = u(\ell,t)$ gives $U_0^n = U_J^n$. The PDE is easily imposed at all grid points. For example,

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2}(0,t) &= \frac{1}{h^2}(U_{-1}^n - 2U_0^n + U_1^n) + O(h^2) \\ &= \frac{1}{h^2}(U_{J-1}^n - 2U_0^n + U_1^n) + O(h^2)\end{aligned}$$

Thus the matrix $\mathbf{L}$ has the structure:

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

and is not tridiagonal, nor is it banded. For implicit methods this is a problem because the work required to invert $\mathbf{A}_- = (\mathbf{I} - \triangle t \mathbf{L})$ is proportional to the square of the bandwidth.

However, $\mathbf{A}_-$ is sparse – it contains mostly zero entries and it differs in only 6 places from a tridiagonal matrix. To invert such a matrix, one can turn to the *Sherman-Morrison formula*.

If

$$\hat{\mathbf{A}} = \mathbf{A} + \mathbf{u} \otimes \mathbf{v}$$

then

$$\hat{\mathbf{A}}^{-1} = \mathbf{A}^{-1} - \frac{(\mathbf{A}^{-1} \cdot \mathbf{u}) \otimes (\mathbf{v} \cdot \mathbf{A}^{-1})}{\mathbf{v} \cdot \mathbf{A}^{-1} \cdot \mathbf{u}}$$

If $\mathbf{u}$ and $\mathbf{v}$ are unit vectors, then $\hat{\mathbf{A}}$ differs from $\mathbf{A}$ in exactly one entry. Hence, $\mathbf{A}_-^{-1}$ can be written in terms of the inverse of a tridiagonal matrix together with 6 applications of the Sherman-Morrison formula.

General Linear Parabolic Eq in 1D

Most of the difficulties encountered in going to the general case are in the analysis of the discretization error and stability. Implementing the general case can be a simple extension of the constant coefficient heat equation.

Write the PDE as:

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

then discretize all quantities in space and time:

$$u(x,t) \rightarrow \mathbf{U}^n = \begin{pmatrix} U_0^n \\ U_1^n \\ \\ \cdots \\ \\ U_J^n \end{pmatrix} \quad \mathcal{L}(x,t) \rightarrow \mathbf{L}^n \quad g(x,t) \rightarrow \mathbf{g}^n = \begin{pmatrix} g_0^n \\ g_1^n \\ \\ \cdots \\ \\ g_J^n \end{pmatrix}$$

where $g_j^n \equiv g(jh, n\triangle t)$.

The matrix $\mathbf{L}^n$ can be obtained from finite-difference formulas for $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial u}{\partial x}$.

The three timestepping schemes are then:

FE:	$\mathbf{U}^{n+1} = \mathbf{B}\ \{(\mathbf{I} + \triangle t \mathbf{L}^n)\ \mathbf{U}^n + \triangle t \mathbf{g}^n\}$
BE:	$\mathbf{U}^{n+1} = (\mathbf{I} - \triangle t \mathbf{L}^{n+1})^{-1}\ \mathbf{B}\ \left\{\mathbf{U}^n + \triangle t \mathbf{g}^{n+1}\right\}$
CN:	$\mathbf{U}^{n+1} = (\mathbf{I} - \frac{\triangle t}{2}\mathbf{L}^{n+1})^{-1}\ \mathbf{B}\ \left\{(\mathbf{I} + \frac{\triangle t}{2}\mathbf{L}^n)\ \mathbf{U}^n + \frac{\triangle t}{2}(\mathbf{g}^n + \mathbf{g}^{n+1})\right\}$

where $\mathbf{B}$ represents the appropriate boundary operator.

Nonlinearity

Difficulties

Some of the numerical difficulties encountered with nonlinear equations are:

- Implicit methods are much more difficult to apply to nonlinear terms.

  Consider timestepping

  $$\frac{\partial u}{\partial t} = F(u)$$

  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. Solving this equation is not simply a matrix inversion.

  Implicit methods are not often used in treating nonlinearities in PDEs. Generally some sacrifices must be made in order of accuracy and/or stability.

- Relatively smooth initial conditions evolve into solutions requiring very fine spatial resolution.

  For example, solutions to Burgers equation

  $$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \epsilon\frac{\partial^2 u}{\partial x^2}$$

  can become locally very sharp.

- Testing is difficult because exact solutions are rare.

Some possible methods

- Explicit Euler method can always be used:

  $$\mathbf{U}^{n+1} = \mathbf{U}^n + \triangle t\mathbf{F}(\mathbf{U}^n).$$

  While not favored because it is low-order accurate ($O(\triangle t)$), it is nevertheless a good method for getting results quickly in a few cases.

- Bite the bullet and invert nonlinear operator, iteratively.

- Use Runge-Kutta or a multistep method.

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.

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.

Some cases are:

$k = 1,$	$\beta_1 = 1$	explicit Euler
$k = 2,$	$\beta_1 = \frac{3}{2}, \beta_2 = -\frac{1}{2}$	2nd order Adams – Bashforth
$k = 4,$	$\beta_1 = \frac{55}{24},\ \beta_2 = -\frac{59}{24},\ \beta_3 = \frac{37}{24},\ \beta_4 = -\frac{9}{24}$	4th order Adams – Bashforth

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 $O(\triangle t^3)$ and discretization error is one power of $\triangle t$ smaller, 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 timestep restriction is too severe for the schemes to be practical.
- By saving previous values of $f(U^n)$, one need only evaluate f once per timestep. This is significant for PDEs and is a significant advantage over other high-order methods such as Runge-Kutta in which multiple function evaluations are required at each timestep.
- The first timestep(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 discretizaton error.

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:

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

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 is $O(\triangle t^2)$.

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

Introduction to Spectral Methods

The essential idea is to approximate the solution $u(x,t)$ by an expansion in a finite set of spectral modes $\phi_m(x)$ and their time-dependent amplitudes $a_m(t)$:

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

Trigonometric functions and orthogonal polynomials (Chebyshev or Legendre polynomials) are the most frequently used. The most natural indexing of modes is not always from 0 to M, and so the summation limits will often not be given, but it is understood to be finite.

Spatial derivatives of $u(x,t)$ are then expressed in terms of spatial derivatives of the ϕ_m, e.g.

$$\frac{\partial^2 u}{\partial x^2}(x,t) \simeq \sum_m a_m(t)\phi_m''(x)$$

So in the trigonometric case $\phi_m = e^{i\beta_m x}$

$$\frac{\partial^2 u}{\partial x^2}(x,t) \simeq \sum_m -\beta^2 a_m(t)\phi_m(x)$$

and we have "exactly" differentiated our approximation.

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

The treatment of time is the generally the same in both methods.

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:

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

The approximation is 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

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

Comments

- 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 methods can be efficiently implemented.
- Spectral methods typically produce smaller artificial dissipation and dispersion in comparison to FD methods.
- Finite-difference and finite-element methods are generally more flexible. 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.

Fourier Pseudospectral Timestepping

Pseudospectral methods can be applied to PDEs with inhomogeneities and nonlinearities. In the pseudospectral approach we have both a spatial grid (collocation points) and a spectral expansion. The numerical solution can either be represented by the amplitudes a_m or by the values on the grid U_j. Operations such as differentiation are best carried out in spectral space (because the modes $\phi_m(x)$ can be differentiated exactly and the derivative operator often local in spectral space). Complications arising from inhomogeneities and nonlinearities are best handled in physical space where they are local.

We consider in detail how to implement pseudospectral timestepping for problems with periodic boundary conditions for which Fourier modes are the appropriate choice.

Amplitudes and Grid values

Let $\mathbf{U}$ denote the vector of values on a uniform grid $x_j = jh$ where $h = \ell/J$. Let $\mathbf{a}$ denote the vector of amplitudes. The spectral expansion relates these

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

Hence

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

Thus the amplitudes and grid values are simply related by the Discrete Fourier Transform (DFT) which shall be written:

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

There are a variety of conventions for ordering the Fourier amplitudes a_m because we are generally interested in real-to-complex DFTs taking J real values U_j on a grid to J complex amplitude $a_m = a_m^r + i a_m^i$. The property $a_m^* = a_{-m}$ implies that the complex amplitudes are actually described by J real quantities and the highest spatial frequency is $J/2$.

We use a generic ordering (storage) for a_m^r and a_m^i and let $\mathbf{a}$ denote the real vector:

$$\mathbf{a} = (a_0^r, a_1^r, a_1^i, a_2^r, a_2^i, \ldots, a_{J/2-1}^r, a_{J/2-1}^i, a_{J/2}^r)^T$$

Necessarily $a_0^i = a_{J/2}^i = 0$ and these are not included in the list. $\mathbf{a}$ is a real vector of length J.

In practice proper account must be taken of the ordering actually produced by the discrete Fourier transform function used. The amplitudes are generally not normalized and require an additional normalization (such as division by J).

Heat Equation

Consider timestepping the heat equation

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

by the forward Euler method

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

then as usual this becomes:

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

$\mathbf{L}$ and $\mathbf{A}_+$ are not the tridiagonal matrices from the finite-difference approach.

The operators are applied in spectral space,

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

where they are diagonal

$$\hat{\mathbf{A}_+} \equiv \begin{bmatrix} 1-\triangle t\beta_0^2 & & & & & & \\ & \ddots & & & & & \\ & & 1-\triangle t\beta_m^2 & & & & \\ & & & 1-\triangle t\beta_m^2 & & & \\ & & & & \ddots & & \\ & & & & & 1-\triangle t\beta_{J/2}^2 \end{bmatrix} = diag\{1-\triangle t\beta_m^2\}$$

We can write our timestepping scheme as:

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

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

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

where

$$\hat{\mathbf{A}_-} \equiv \begin{bmatrix} 1+\triangle t\beta_0^2 & & & & & & \\ & \ddots & & & & & \\ & & 1+\triangle t\beta_m^2 & & & & \\ & & & 1+\triangle t\beta_m^2 & & & \\ & & & & \ddots & & \\ & & & & & 1+\triangle t\beta_{J/2}^2 \end{bmatrix} = diag\{1+\triangle t\beta_m^2\}$$

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

Crank-Nicolson timestepping follows similarly.

Note that $\mathbf{L}$, $\mathbf{A}_+$, etc. are diagonal only when the boundary conditions are periodic, permitting the use of Fourier expansions. For Dirichlet boundary conditions, for example, polynomial expansions are appropriate; the resulting matrices representing differential operators are banded or structured, but not diagonal.

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. (Here we shall assume that $\mathcal{N}()$ does not depend on $\frac{\partial u}{\partial x}$, $\frac{\partial^2 u}{\partial x^2}$, etc. although this assumption in not essential.)

Discretize space and time in the usual way and timestep with 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:

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

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:

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

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

where hats denote linear operators in spectral space.