Numerical Analysis

David Connelly
Spring 2022

Contents

1 Floating-point arithmetic

1.1 Floating-point formats

A single-precision number has structure

$$1 \text{ sign bit} + 8 \text{ exponent bits} + 23 \text{ mantissa bits}$$

which means, if the sign is s, the exponent is e, and the mantissa bits are $b_0, \ldots, b_{22}$, the value stored is

$$(-1)^s \cdot \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i}\right) \cdot 2^{e-127}$$

Similarly, a double-precision number has

$$1 \text{ sign bit} + 11 \text{ exponent bits} + 52 \text{ mantissa bits}$$

so that the number stored is

$$(-1)^s \cdot \left(1 + \sum_{i=1}^{52} b_{53-i} 2^{-i}\right) \cdot 2^{e-1023}$$

1.2 Condition numbers

If we have a function f and an approximation $\tilde{f}$ to that function, then if f is differentiable, the relative condition number is

$$\kappa = \left| \frac{x f'(x)}{f(x)} \right|$$

An important example is the condition number of a matrix — that is, of the problem that solves $Ax = b$ for x. We assume that b has error e, so that the error in the solution is $A^{-1}e$, so long as A is invertible. By taking the ratio of the relative error in the solution to the relative error in the data and applying the definition of the operator norm, we find

$$\kappa(A) = \|A\| \left\|A^{-1}\right\| \geq 1$$

Recall that if we take the inducing norm to be Euclidean, we then have

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

1.3 Cancellation errors

Subtraction is ill-conditioned for nearby inputs. Namely, suppose we wish to compute $x - y$ and we have only the approximations $\tilde{x} = x + \Delta x$ and $\tilde{y} = y + \Delta y$. Then

$$\begin{aligned} \tilde{x} - \tilde{y} &= x + \Delta x - y - \Delta y \\ &= (x - y) + (\Delta x - \Delta y) \end{aligned}$$

so that the relative error is

$$\frac{(\tilde{x} - \tilde{y}) - (x - y)}{x - y} = \frac{\Delta x - \Delta y}{x - y}$$

which can get arbitrarily large for $x \approx y$.

2 Matrix decompositions

2.1 LU decomposition

The LU decomposition writes $A = LU$ where L is lower-triangular and U is upper-triangular. For uniqueness in certain cases, we require the diagonal of L to be all ones. We also sometimes allow permutations, so that the decomposition is $PA = LU$.

Every square matrix admits a $PA = LU$ decomposition. If A is invertible and an LU decomposition exists, it is unique. A necessary and sufficient condition for existence is for the principal minors all to be nonzero. A matrix A can also admit infinitely many LU decompositions if A is singular.

The LU decomposition is computed in $\mathcal{O}\left(n^3\right)$ time with Gaussian elimination. The idea is simply to iteratively eliminate the elements below the main diagonal of A one column at a time, then invoke the fact that a lower-triangular matrix has a lower-triangular inverse.

2.2 QR decomposition

The QR decomposition writes $A = QR$, where Q is orthogonal and R is upper-triangular. For invertible A, requiring the diagonal entries of R to be positive ensures uniqueness. Moreover, the first k columns of Q form an orthonormal basis for the space spanned by the first k columns of A for all $1 \leq k \leq n$.

One means of computing the QR decomposition is by the Gram-Schmidt process. Here, we define an orthonormal basis spanning the column space of A by starting with the first column and iteratively subtracting the projections onto the basis vectors taken thus far from the next column.

Gram-Schmidt is numerically unstable, so modified Gram-Schmidt proceeds in a way that is algebraically identical but more likely to produce numerically orthogonal vectors. In this method, we first orthogonalize all the vectors to the first basis vector, than the second, and so on, rather than constructing each new basis vector in full one at a time.

Another method uses Householder reflections. The idea here is to choose an orthogonal matrix that reflects the first column of A across a hyperplane such that it ends up colinear to e_1. If a_1 is the first column, we can take

$$\begin{aligned} v_1 &= a - \|a\| e_1 \\ Q_1 &= I - 2\frac{vv^\top}{v^\top v} \end{aligned}$$

and then $Q_1 A$ has nothing below the first diagonal entry. We can then recurse. Both the (modified) Gram-Schmidt method and the method of Householder reflections have a time complexity of $\mathcal{O}\left(n^3\right)$.

Note that A need not be square. If A is $m \times n$ with $m \geq n$, then we can write $A = QR$ with $Q \in \mathbb{R}^{m \times m}$ and $R \in \mathbb{R}^{m \times n}$. Since R is upper-triangular, the bottom $m - n$ rows of R are zero. This decomposition is useful for solving overdetermined systems in the least-squares sense. The solution to such a system $Ax = b$ is $x = R_1^{-1} Q_1^\top b$, where R_1 is the first m rows of R and Q_1 is the first n columns of Q.

2.3 Eigenvalue decomposition

If a square matrix A has linearly independent eigenvectors, then it can be written $A = QDQ^{-1}$, where D is diagonal with the eigenvalues along the diagonal, and the columns of Q are the eigenvectors of A. If A is symmetric, Q will be orthogonal so that $A = QDQ^\top$.

One standard way of way of computing the eigenvalues is to iteratively decompose $A_k = QR$ and then set $A_{k+1} = RQ$. Note that this operation preserves eigenvalues, as

$$\begin{aligned} A_k v &= \lambda v \\ \implies QRv &= \lambda v \\ \implies Q(RQ)Q^\top v &= \lambda v \\ \implies A_{k+1}\left(Q^\top v\right) &= \lambda\left(Q^\top v\right) \end{aligned}$$

It turns out that A_k tends to converge to an upper-triangular matrix, which of course means its eigenvalues can be read off of the diagonal. The QR algorithm with an ordinary routine for the QR decomposition is $\mathcal{O}\left(n^3\right)$, but specialized modifications can be made to reduce it to $\mathcal{O}\left(n^2\right)$.

2.4 Singular value decomposition

Let $A = \mathbb{R}^{m\times n}$. Then the singular value decomposition writes $A = U\Sigma V^\top$, where

$$\begin{aligned} U &\in \mathbb{R}^{m\times m} \text{ has eigenvectors of } AA^\top \text{ as columns} \\ \Sigma &\in \mathbb{R}^{m\times n} \text{ has square roots of eigenvalues of } AA^\top \text{ and } A^\top A \text{ as diagonal entries} \\ V &\in \mathbb{R}^{n\times n} \text{ has eigenvectors of } A^\top A \text{ as columns} \end{aligned}$$

When A is real (the case considered here) then U and V will be orthogonal. The SVD has applications including computation of the pseudoinverse and of low-rank approximations that are optimal under the Frobenius norm.

2.5 Cholesky decomposition

A real, positive semi-definite matrix A can be decomposed $A = LL^\top$, where L is lower-triangular. This decomposition is of obvious utility for solving linear systems involving A.

The classical Cholesky algorithm is $\mathcal{O}\left(n^3\right)$ and proceeds by decomposing

$$A = \begin{pmatrix} \alpha & * \\ a & B \end{pmatrix}$$
$$L = \begin{pmatrix} \lambda & 0 \\ \ell & C \end{pmatrix}$$

so that, substituting into $A = LL^\top$, we have

$$\begin{pmatrix} \alpha & * \\ a & B \end{pmatrix} = \begin{pmatrix} \lambda^2 & 0 \\ \lambda\ell & \ell\ell^\top + CC^\top \end{pmatrix}$$

Immediately, we see that we should take $\lambda = \sqrt{\alpha}$ and then $\ell = a/\lambda$. We then have $B - \ell\ell^\top = CC^\top$, where everything on the left-hand side is known and C is supposed to be lower-triangular, so we can recurse to solve for C.

3 Linear systems

3.1 Jacobi iteration

If we wish to solve the system $Ax = b$, we can decompose $A = L + D + U$, where L is lower-triangular, D is diagonal, and U is upper-triangular. Then Jacobi iteration consists of setting

$$x_{k+1} = D^{-1}\left(b - (L+U)x_k\right)$$

A sufficient condition for convergence is that A be strictly diagonally dominant — that is, for all i we have

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}|$$

3.2 Gauss-Seidel method

Another iterative method involves writing $A = L_* + U$, where U is strictly upper-triangular but L_* has entries on the diagonal. Then the iteration is

$$x_{k+1} = L_*^{-1}\left(b - Ux_k\right)$$

where multiplication by L_* is easy to evaluate by substitution. Gauss-Seidel is guaranteed to converge if A is diagonally dominant, as before; it is also sure to converge if A is symmetric positive-definite.

3.3 Conjugate gradients

The conjugate gradient method offers an iterative way to solve large systems $Ax = b$ when A is symmetric positive-definite. The method involves finding a set of vectors p_i that are pairwise "conjugate" — that is, orthogonal under the inner product induced by A — and that thus constitute a basis. Then the coefficients of x in that basis are easily obtained.

To find such p_k, we begin by setting $p_0 = b - Ax_0$. Then at each step we define the residual $r_k = b - Ax_k$. Instead of using the residual as the next direction, we first us Gram-Schmidt orthogonalization to make the new direction conjugate to the previous directions. That is, we take

$$p_k = r_k - \sum_{i=1}^{k-1} \frac{p_i^\top A r_k}{p_i^\top A p_i} p_i$$

Then we set

$$\begin{aligned} \alpha_k &= \frac{p_k^\top r_k}{p_k^\top A p_k} \\ x_{k+1} &= x_k + \alpha_k p_k \end{aligned}$$

4 Interpolation

Suppose we have a sequence of $n+1$ points (x_i, y_i) with $0 \le i \le n$ with $x_i \ne x_j$ when $i \ne j$. We will seek a polynomial p such that $p(x_i) = y_i$ for each i. In particular, we will find that there exists a unique interpolating p of degree at most n.

Uniqueness is easy to check. If p and q are interpolating polynomials of degree at most n, then $w \equiv p - q$ is of degree at most n and has $n+1$ zeros — namely, the x_i. But a nonzero polynomial of degree at most n can have at most n zeros, so we must have $w = 0$ identically, and so $p = q$.

4.1 Lagrange basis

Given the data, we define

$$\ell_i(x) = \prod_{j \ne i} \frac{x - x_j}{x_i - x_j}$$

Note that $\ell_i(x_j) = \delta_{ij}$, and so

$$p(x) = \sum_{i=0}^{n} y_i \ell_i(x)$$

is an interpolating polynomial. Each ℓ_i is a product of n monomials, and thus p is of degree at most n.

4.2 Newton basis

If more data points are added, the Lagrange interpolant must be computed again from scratch. Instead, we can work in the Newton basis. Define the recurrence

$$\begin{aligned} [y_i] &= y_i \\ [y_i, \dots, y_{i+k}] &= \frac{[y_{i+1}, \dots, y_{i+k}] - [y_i, \dots, y_{i+k-1}]}{x_{i+k} - x_i} \end{aligned}$$

Then the interpolating polynomial can be written

$$p(x) = y_0 + \sum_{i=1}^{n} \left([y_0, \dots, y_i] \prod_{j=0}^{i-1} (x - x_j) \right)$$

which has the advantage that new points can be incorporated simply by computing the next term.

4.3 Error bounds

If p is the interpolating polynomial of degree at most n, then

$$\|f-p\|_\infty \leq \frac{1}{(n+1)!}\left\|f^{(n+1)}\right\|_\infty \left\|\prod_{i=0}^{n}(x-x_i)\right\|_\infty$$

when the interval is $[-1,1]$.

4.4 Chebyshev nodes

The Chebyshev nodes on $(-1,1)$ are

$$x_i = \cos\left(\frac{2k-1}{2n}\pi\right)$$

for $1 \leq i \leq n$. They are the x-coordinates of n points spaced equally along a unit semicircle. The Chebyshev nodes are good for polynomial interpolation in that if p is the interpolant of degree n fit at these nodes, the error bound given above becomes

$$\|f-p\|_\infty \leq \frac{1}{2^n(n+1)!}\left\|f^{(n+1)}\right\|_\infty$$

5 Quadrature

Quadrature refers to the numerical computation of integrals such as

$$\int_a^b f(x)\,\mathrm{d}x$$

We will generally consider evaluating f at nodes $a = x_0 \leq \ldots \leq x_n = b$. Often these nodes will be equally spaced within the interval, in which case we refer to the grid spacing as h.

5.1 Trapezoidal rule

The trapezoidal quadrature is obtained simply by treating each subinterval as a trapezoid and calculating its area. If the grid is equally spaced, we obtain

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{2}\Big(f(x_0) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n)\Big)$$

Let E be the magnitude of the error. Then

$$E \leq \frac{(b-a)^3}{12n^2}\max_{\xi\in[a,b]}\left|f(\xi)\right|$$

such that the error is $\mathcal{O}\left(h^2\right)$.

5.2 Simpson's rule

Simpsons's rule on a single interval is

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

which can be obtained by fitting a quadratic interpolant at the endpoints a and b as well as the midpoint $(a+b)/2$. The magnitude of the error satisfies

$$E \leq \frac{1}{90}\left(\frac{b-a}{2}\right)^5 \max_{\xi\in[a,b]}\left|f^{(4)}(\xi)\right|$$

5.3 Gaussian quadrature

A Gaussian quadrature rule approximates

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i)$$

where the choice of nodes x_i and weights w_i determines the method. In particular, there is the Gauss-Legendre rule, which is optimal in the sense that it is exact for polynomials of degree at most $2n-1$.

6 Finite difference methods for ordinary differential equations

We begin by noting that finite difference approximations to derivatives are generally derived from Taylor series expansions. We can demonstrate this by showing that the centered difference is accurate to second order in h. We have

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \mathcal{O}\left(h^3\right) \\ f(x-h) &= f(x) - f'(x)h + \frac{1}{2}f''(x)h^2 + \mathcal{O}\left(h^3\right) \end{aligned}$$

so that

$$\begin{aligned} \frac{f(x+h) - f(x-h)}{2h} &= \frac{2f'(x)h + \mathcal{O}\left(h^3\right)}{2h} \\ &= f'(x) + \mathcal{O}\left(h^2\right) \end{aligned}$$

where we have benefited from the second-order terms in each Taylor expansion cancelling. For reference, we also record the canonical centered difference approximation of $f''(x)$, which is derived similarly.

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} + \mathcal{O}\left(h^2\right)$$

6.1 Consistent one-step methods

We consider the initial value problem

$$\begin{aligned} u'(t) &= f(u,t) \\ u(0) &= u_0 \end{aligned}$$

Recall that if f is Lipschitz we are guaranteed a solution at least for some time.

A general one-step method is of the form

$$\frac{U^{n+1} - U^n}{h} = \varphi_h\left(U^n\right)$$

where the method is *consistent* if $\varphi_0(u) = f(u)$. We assume that φ_h is Lipschitz continuous in u with constant L (which is generally related to the Lipschitz constant of f). For a particular step size h, the local truncation error is

$$\tau^n = \frac{u_{n+1} - u_n}{h} - \varphi_h\left(u_n\right)$$

where $u_n \equiv u\left(t_n\right)$. The global error is

$$E^n = U^n - u_n$$

Combining the update step with the definition of the local truncation error gives

$$E^{n+1} = E^n + h\Big(\varphi_h\left(U^n\right) - \varphi_h\left(u_n\right)\Big) - h\tau^n$$

so that, by Lipschitz continuity, we have

$$\begin{aligned}\left|E^{n+1}\right| &\leq h\left|\tau^n\right| + \left|E^n\right|(1+hL) \\ &\leq h\left|\tau^n\right| + (1+hL)h\left|\tau^{n-1}\right| + (1+hL)^2\left|E^{n-1}\right|\end{aligned}$$

and so on. If we continue the above process and assume $E^0 = 0$, we reach

$$\left|E^n\right| = h\sum_{m=1}^{n}(1+hL)^{m-1}\left|\tau^{n-m}\right|$$

Now, note that the function $e^\xi - 1 - \xi$ is zero at $\xi = 0$ and increasing for $\xi > 0$. It follows that $e^\xi \geq 1 + \xi$ for any positive ξ, and in particular, since h and L are positive, we have

$$\begin{aligned}(1+hL)^{m-1} &\leq e^{(m-1)hL} \\ &\leq e^{nhL} \\ &\leq e^{TL}\end{aligned}$$

where T is the largest time we consider, so $nh = t \leq T$. Then

$$\begin{aligned}\left|E^n\right| &\leq (nh)e^{TL}\max_m\left|\tau^m\right| \\ &\leq Te^{TL}\max_m\left|\tau^m\right|\end{aligned}$$

The only remaining dependence on h is in the size of the local truncation errors, which vanish as $h \to 0$ provided the method φ_h is consistent. Thus the method converges.

6.2 A-stability

The absolute stability of a method is determined with the standard test problem

$$\begin{aligned}u'(t) &= \lambda u \\ u(0) &= 1\end{aligned}$$

which of course has solution $u(t) = e^{\lambda t}$. The region of absolute stability of a method is the set S in the complex plane such that if $h\lambda \equiv z \in S$, then the numerical solution to the standard test problem decays to zero over time.

For example, consider the forward Euler method. The update step is

$$\begin{aligned}U^{n+1} &= (1+h\lambda)\,U^n \\ &= (1+z)^{n+1}U^0 \\ &= (1+z)^{n+1}\end{aligned}$$

so that, for the solution to decay (or at least not blow up), we need $|1+z| \leq 1$ and thus S is the circle of radius one around -1. For backward Euler, we instead have

$$\begin{aligned}U^{n+1} &= \frac{1}{1-h\lambda}U^n \\ &= \frac{1}{(1-z)^{n+1}}\end{aligned}$$

such that we now need $|1-z| \geq 1$ and thus S is the complex plane *except* the circle of radius one centered at 1. In particular, for backward Euler the region of absolute stability includes the entire half-plane $\mathrm{Re}(z) \leq 0$, such that if $\lambda \leq 0$ and the true solution does not blow up, the numerical solution will not either. This property is termed A-stability. It is a result that no explicit one-step method can be A-stable.

6.3 Common time stepping schemes

method	update step	order	stability region
forward Euler	$\frac{U^{n+1}-U^n}{h} = f\left(U^n\right)$	$\mathcal{O}\left(h\right)$	$\lvert z+1\rvert < 1$
backward Euler	$\frac{U^{n+1}-U^n}{h} = f\left(U^{n+1}\right)$	$\mathcal{O}\left(h\right)$	$\lvert z-1\rvert > 1$
trapezoidal rule	$\frac{U^{n+1}-U^n}{h} = \frac{f\left(U^n\right)+f\left(U^{n+1}\right)}{2}$	$\mathcal{O}\left(h^2\right)$	$\mathrm{Re}(z) < 0$
RK2	$\frac{U^{n+1}-U^n}{h} = f\left(U^n + hf\left(U^n\right)\right)$	$\mathcal{O}\left(h^2\right)$	$\left\lvert 1+z+z^2\right\rvert < 1$
AB2	$\frac{U^{n+1}-U^n}{h} = \frac{1}{2}\left(3f\left(U^n\right) - f\left(U^{n-1}\right)\right)$	$\mathcal{O}\left(h^2\right)$	blob in $\mathrm{Re}(z) < 0$ half-plane

7 Finite difference methods for partial differential equations

7.1 Diffusion

We begin with the problem

$$\begin{aligned} u_t &= u_{xx} \\ u(x,0) &= \eta(x) \\ u(0,t) &= a(t) \\ u(1,t) &= b(t) \end{aligned}$$

where without loss of generality we have set the coefficient to unity. We will consider first discretizing in space, giving the system of ordinary differential equations

$$\begin{aligned} U_0(t) &= a(t) \\ U_i'(t) &= \frac{U_{i-1}(t) - 2U_i(t) + U_{i+1}(t)}{h^2},\ 1 \le i \le m \\ U_{m+1}(t) &= b(t) \end{aligned}$$

where h is the spatial grid spacing. Such an approach is called the *method of lines*. The system can be written in matrix form as

$$U'(t) = AU(t) + g(t)$$

where

$$A = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \text{ and } g(t) = \frac{1}{h^2}\begin{pmatrix} a(t) \\ 0 \\ \vdots \\ 0 \\ b(t) \end{pmatrix}$$

Now, it turns out that the eigenvalues of A are

$$\lambda_p = \frac{2}{h^2}\Big(\cos(p\pi h) - 1\Big) \text{ for } 1 \le p \le m$$

Clearly, all eigenvalues are negative and real, and the largest possible eigenvalue is $-4/h^2$. If we solve the method of lines discretization with, say, forward Euler, for stability we need $k\lambda_{\max} \ge -2$, which works out to imply $k \le h^2/2$, a severe CFL condition. On the other hand, using trapezoidal time stepping, which is A-stable, we can take whatever time step we want, and this choice leads to the Crank-Nicolson method.

To determine the error, we note that in the interior the Crank-Nicolson truncation error is

$$\begin{aligned} \tau = & \frac{u(x,t+k) - u(x,t)}{k} \\ & - \frac{u(x-h,t) - 2u(x,t) + u(x+h,t)}{2h^2} \\ & - \frac{u(x-h,t+k) - 2u(x,t+k) + u(x+h,t+k)}{2h^2} \end{aligned}$$

To calculate the error, we will break the time part into two halves and expand one around t and the other around $t+k$. For the first term, we have (up to a factor of two)

$$\left(u_t + \frac{k}{2}u_{tt} + \frac{k^2}{6}u_{ttt}\right) - \left(u_{xx} + \frac{h^2}{12}u_{xxxx}\right) = \left(\frac{k}{2} - \frac{h^2}{12}\right)u_{tt} + \frac{k^2}{6}u_{ttt}$$

because $u_t = u_{xx}$ and so $u_{tt} = u_{xxxx}$. For the other term, we end up with

$$-\left(\frac{k}{2} + \frac{h^2}{12}\right)u_{tt} + \frac{k^2}{6}u_{ttt}$$

but where all derivatives are evaluated at $t+k$. The total truncation error will be the sum of these terms. Already, we can see that τ is $\mathcal{O}\left(h^2\right)$ and at least $\mathcal{O}\left(k\right)$, so the method converges. To check the order of convergence in space, we just look at the contributions from the first order terms. We have

$$\begin{aligned} -\frac{k}{2}\Big(u_{tt}(x,t+k) - u_{tt}(x,t)\Big) &= -\frac{k}{2}\Big(ku_{ttt}(x,t) + \cdots\Big) \\ &= -\frac{k^2}{2}u_{ttt}(x,t) \end{aligned}$$

so that this term is in fact also second-order in time, and thus Crank-Nicolson is $\mathcal{O}\left(h^2 + k^2\right)$.

7.2 Advection

The advection equation is

$$u_t + au_x = 0$$

typically with some initial data and boundary conditions provided (for most of the following analysis, we will take periodic boundary condtions on $[0,1]$. One straightforward method of lines discretization is

$$U_j'(t) = -\frac{a}{2h}\Big(U_{j+1}(t) - U_{j-1}(t)\Big)$$

for $1 \leq j \leq m+1$ and U_{m+1} is the point on the right boundary. Then we have $U' = AU$ where

$$A = -\frac{a}{2h}\begin{pmatrix} 0 & 1 & & & -1 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 1 & & & -1 & 0 \end{pmatrix}$$

Note that $A = -A^\top$ and so the eigenvalues are pure imaginary. So forward Euler, for example, can never be stable with a fixed ratio k/h.

To achieve second-order convergence, we Taylor expand and write

$$\begin{aligned} u(x, t+k) &= u(x,t) + ku_t + \frac{1}{2}k^2 u_{tt} \\ &= u(x,t) - kau_x + \frac{k^2a^2}{2}u_{xx} \end{aligned}$$

which suggests the discretization

$$\frac{U_j^{n+1} - U_j}{k} = -\frac{a}{2h}\left(U_{j+1}^n - U_{j-1}^n\right) + \frac{ka^2}{2h^2}\left(U_{j-1}^n - 2U_j^n + U_{j+1}^n\right)$$

and this is the Lax-Wendroff method.

We can determine the stability condition for Lax-Wendroff via Von Neumann analysis (to be discussed later). Some more or less straightforward algebra finds the amplitude function to be

$$g(\xi) = 1 + \nu^2(\cos z - 1) - i\nu \sin z$$

where $z = h\xi$ and $\nu = (ka)/h$. We want $|g(\xi)| \leq 1$, and so we calculate

$$\begin{aligned} |g(\xi)|^2 &= \left[1 + \nu^2(\cos z - 1)\right]^2 + \nu^2 \sin^2 z \\ &= 1 - \nu^2(1 - \cos z)^2\left(1 - \nu^2\right) \end{aligned}$$

after a few lines of algebra. We want $|g(\xi)|$ to be bounded by unity, which means the last factor in the rightmost term must be positive. Thus we have the condition

$$|\nu| = \left|\frac{ka}{h}\right| \leq 1$$

which is the right condition. We could also obtain this result by noting that the eigenvalues of the matrix for the spatial discretization of Lax-Wendroff satisfy

$$k\lambda_p = -i\nu \sin(p\pi h) + \nu^2\Big(\cos(\pi p h) - 1\Big)$$

and demanding that these values lie in the stability region for forward Euler.

We also consider the upwind discretization

$$\frac{U_j^{n+1} - U_j^n}{k} = -\frac{a}{h}\left(U_j^n - U_{j-1}^n\right)$$

for $a > 0$. This method acknowledges the asymmetry in the true solution to the advection equation, but is first order in space and time and still has the stability requirement $0 \leq (ka)/h \leq 1$.

7.3 Lax equivalence theorem

Suppose h is related to k by some fixed rule. Then many methods in which we are interested can be written

$$U^{n+1} = A(k)U^n + g^n(k)$$

Such a method is called *Lax-Richtmeyer stable* if, for every time T, there is a constant C_T such that $\|A(k)^n\| \leq C_T$ for all $k > 0$ and n with $nk \leq T$. The Lax equivalence theorem states that if such a method is consistent (so $\tau \to 0$ as h and k get small) and Lax-Richtmeyer stable, then it is convergent.

Note that for Lax-Richtmeyer stability it suffices to exhibit a constant α such that $\|A(k)\| \leq 1 + \alpha k$ for small enough k, for then

$$\|A(k)^n\| \leq (1 + \alpha k)^n \leq e^{\alpha nk} \leq e^{\alpha T} \equiv C_T$$

whenever $nk \leq T$.

7.4 Von Neumann analysis

Von Neumann analysis provides an alternate method of determining stability conditions for linear equations that *does* generally require constant coefficients but *does not* rely on the scheme in question being expressible as a method of lines. The idea is to track the amplification of individual Fourier modes.

For example, we can derive again the diffusive CFL condition for forward Euler. We set $U_j^n = e^{ijh\xi}$ and seek to write $U_j^{n+1} = g(\xi)U_j^n$, where $g(\xi)$ is the wavenumber-dependent amplification function. We have

$$\begin{aligned} U_j^{n+1} &= U_j^n + \frac{k}{h^2}\left(U_{j-1}^n - 2U_j^n + U_{j+1}^n\right) \\ &= e^{ijh\xi} + \frac{k}{h^2}\left(e^{i(j-1)h\xi} - 2e^{ijh\xi} + e^{i(j+1)h\xi}\right) \\ &= e^{ijh\xi} + \frac{k}{h^2}\left(e^{-ih\xi} - 2 + e^{ih\xi}\right)e^{ijh\xi} \end{aligned}$$

whence we conclude

$$g(\xi) = 1 + \frac{2k}{h^2}\Big(\cos(h\xi) - 1\Big)$$

Assuming we are considering the Cauchy problem (so there are no boundary issues to worry about), we can derive a condition for Lax-Richtmeyer stability by ensuring the grid can be chosen such that $|g(\xi)| \leq 1$. The cosine term is bounded in norm by 1, so we see that

$$1 - \frac{4k}{h^2} \leq g(\xi) \leq 1$$

and so we need $4k/h^2 \leq 2$, or

$$\frac{k}{h^2} \leq \frac{1}{2}$$

which is the same condition we found earlier.

7.5 Modified equations

We can check if there is a partial differential equation that a particular method solves better than the equation we set out to solve. For example, for the upwind method, we seek a v that satsifies

$$v(x, t + k) = v(x, t) - \frac{ka}{h}\Big(v(x, t) - v(x - h, t)\Big)$$

exactly. If we Taylor expand v and keep the $\mathcal{O}(k)$ terms, we eventually find

$$v_t + au_x = \frac{1}{2}\left(hav_{xx} - kv_{tt}\right)$$

$$= \frac{1}{2}ah\left(1 - \frac{ka}{h}\right)v_{xx}$$

up to terms that are $\mathcal{O}\left(k^2\right)$. Thus we see that v is the solution to an advection-diffusion equation, which can help predict behavior to expect in the numerical solution. Lax-Wendroff has the modified equation

$$v_t + av_x + \frac{1}{6}h^2 a\left(1 - \frac{k^2a^2}{h^2}\right)v_{xxx} = 0$$

and so we conclude the method is dispersive.

8 Spectral methods

8.1 Fast Fourier transform

Given a smooth, periodic function on $[0, 2\pi]$ with grid spacing $h = 2\pi/N$ and nodes $x_j = jh$, the discrete Fourier transform defines the Fourier amplitudes

$$\hat{f}_k = \frac{1}{N}\sum_{j=0}^{N-1} f(x_j)\exp\left\{-i\frac{2\pi jk}{N}\right\}$$

and the inverse transform recovers the node values

$$f(x_j) = \sum_{-(N-1)/2}^{(N-1)/2} \hat{f}_k \exp\left\{i\frac{2\pi jk}{N}\right\}$$

The discrete transform and its inverse can be computed in $\mathcal{O}\left(N\log N\right)$ time with the fast Fourier transform.

8.2 Solving partial differential equations

Consider the Korteweg-de Vries equation

$$u_t = 3\left(u^2\right)_x - u_{xxx}$$

We can note that in Fourier space we have a system of ordinary differential equations

$$\hat{u}_t = 3ik\left(\widehat{u^2}\right)_k + ik^3\hat{u}_k$$

The idea is to track that Fourier coefficients, updating at every time step by casting back into real space with the fast Fourier (inverse) transform to compute the nonlinearity. Note that we should oversample to avoid aliasing effects. This approach is spectrally accurate in space, such that

$$\|u(x) - \phi(x)\| \sim e^{-N}$$

where N is the number of nodes. Here we have *ignored* the error from the time integration, which will generally be the largest source of error, so we should use some higher-order time stepping scheme to maintain good accuracy.