Numerical Methods for Fluid Flows

In these sections we give some introductory material for the numerical analysis of simple aspects of the partial differential equations commonly found in modelling geophysical flows. Accuracy and stability of numerical methods are covered.

(1) Definitions
(2) Modelling errors
(3) Methods for the approximation of derivatives
(4) Finite volume approximations
(5) Von Neumann stability analysis for turbulent flows
(6) Inversion of large sparse matrices arising from implicit methods
(7) Advective and diffusive flows in 1D
(8) Highly diffusive flows in 1D
(9) Finite volume and leapfroging for purely advective 1D flows
(10) Initial and boundary conditions for the leapfrog method for advective 1D flows
(11) Stability analysis for the leapfrog method for advective 1D flows
(12) Upstream upwind or donor cell method
(13) The Lax-Wendroff method
(14) The Crank-Nicholson method
(15) 1D Advection and diffusion with a source/sink term
(16) 2D Advective flows - double 1D discretizations
(17) Operator splitting methods for 2D advection - Strang splitting
(18) The discrete Poisson equation
(19) Jacobian evaluations

Numerical Methods for Geophysical Flows

Contents of this linked notes:

In this link we give a summary introduction to the elements of numerical analysis for geophysical flows. We consider only simple time dependent equations and diffusion which is sufficient to illustrate some of the main issues. In further sections we consider advection, sources and sinks, again in the context of a single equation. The reader is directed to Cushman-Roisin's text, "Introduction to Geophysical fluid dynamics" 2nd edition, for a much more complete account, especially Chapter 5 and Appendix C.2-C.4.

(1) Definitions:

In this section we summarize methods for some of the schemes used in computational fluid dynamics to approximate differential expressions $f(x)$ with finite difference approximations. Here a method is derived using Taylor series approximations with assumptions being made about the differential order of variable coefficients. A method $M(\Delta x)$ then satisfies

$$f(x) = M(\Delta x) + O(\Delta x^n)$$

where $\Delta x \to 0$ and where the positive integer $n \geq 1$ is the so-called **order** of the method, and $O(\Delta x^n) := O((\Delta x)^n)$ the **truncation error**. The implied constant can depend on the coefficient functions and parameters that appear in $f(x)$.

We say a method is **consistent** if

$$\lim_{\Delta x \to 0} M(\Delta X) = f(x).$$

We say a method is **convergent** if the solution to the method when set to zero for fixed Δx (the **discrete solution**) tends to the solution of the exact equation $f(x) = 0$ as Δx tends to 0.

The **accuracy order** of a method is the largest value of $n \geq 1$ such that

$$f(x) = M(\Delta x) + O(\Delta x^n),$$

in which case say it is nth order accurate. If $n = 1$ we say the method is **first-order accurate**.

We say a method is **overly-stable** if the Euclidean norm of the discrete solution tends to zero monotonically as $\Delta x \to 0$. It is **stable** if the norm of the discrete solution remains bounded on every bounded interval in the sense that for each $T > 0$, if x^n is the solution to the nth iteration (i.e. with $t = n\Delta x$) and x^0 the initial condition, then there is a fixed constant $C > 0$ (which might depend on T) such that

$$\|x^n\| \leq C\|x^0\|$$

when $n\Delta x \leq T$.

A method is **unstable** if the solution to the discrete equation grows significantly faster than the solution to the exact equation. It is **over stable** if the solution to the discrete equation decreases to zero.

Well-posed partial differential equation: A PDE is well posed if two conditions are satisfied. (a) A unique solution exists for each choice of data values and (b) the mapping from data values to solutions is continuous in some topology.

Lax-Richmeyer equivalence theorem: A consistent method for a linear partial differential equation for which the initial value problem is well-posed is convergent if and only if it is stable.

(2) Modelling errors: Modelling errors arise because the model equations used to describe a particular phenomena do not approximate all of its features. The model often is produced by simplifying equations, often making assumptions that a flow is more uniform in particular ways than it is in reality or by averaging over time, space or ensembles of solutions.
(2.1) Discretization errors: These errors arise when a set of model equations is discretized and the discrete equations solved for some particular value of parameters such as Δt and Δx. The discretization error is the difference between the exact equation and the solution of the discretized equation.

(2.2) Iteration errors: These errors arise for iterative methods since the the iterations must be terminated at some finite value of n. The iteration error is the difference between the solution to the discrete equation as $n \to \infty$ and the solution to the nth iterate. It depends on n and should tend to zero as $n \to \infty$.

(2.3) Rounding errors: These errors arise since only a finite number of digits are used to solve the discretized equations.

Acknowledgement: Section 4.8, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(3) Methods for the approximation of derivatives

(3.1) Explicit Euler approximation to the first derivative: Methods are derived using Taylor series approximations combined and manipulated in various ways, assuming that all derivatives which are used exist and are continuous. If $t^n := n\Delta t$ and $u^n := u(t^n)$, with all other variables and parameters appearing in u suppressed, then

$$\left.\frac{du}{dt}\right|_{t^n} = \frac{u^{n+1} - u^n}{\Delta t} + O(\Delta t).$$

This is the first order Euler method. It is called **explicit** because if we consider the equation

$$\frac{du}{dt} = F(t, u),$$

then we can write $u^{n+1} = u^n + \Delta t F(t^n, u^n) := u^n + \Delta t F^n$, so u^{n+1} can be determined directly once we have the value of u^n. It is also called a **forward method**, and is first order.

If we write

$$u^{n+1} = u^n + \Delta t F(t^n, u^{n+1}),$$

given u^n, to find u^{n+1} we need to solve an equation which could be difficult. The method is called **implicit** or a **backward** method and is again first order.

(3.2) Semi-implicit trapezoidal method: Combining the explicit and implicit Euler methods by taking the average of the right hand sides, we get

$$u^{n+1} = u^n + \Delta t \frac{F^n + F^{n+1}}{2}.$$

It is second order, the best of the so-called **two-point methods**. We can increase the order of accuracy by using points between t^n and t^{n+1} and higher order polynomial interpolations for F.

(3.3) Leap-frog method: This method uses the values of u at t^{n-1} and t^n to obtain an explicit second order method when approximating the first derivative of u.

$$u^{n+1} = u^{n-1} + 2\Delta t F^n.$$

(3.4) Fourth order approximation to the first derivative: Using undetermined coefficients, with evaluations of $u(t^j)$ at five points with $n-2 \leq j \leq n+2$, to get $u^{n-2}, u^{n-1}, u^n, u^{n+1}, u^{n+2}$ so

$$\left.\frac{du}{dt}\right|_{t^n} \approx a_{-2}u^{n-2} + a_{-1}u^{n-1} + a_0 u^n + a_1 u^{n+1} + a_2 u^{n+2}.$$

We impose the conditions that the sum of the a_j is 1 and that the discrete solution must be consistent, and then optimize the set of solutions such that the truncation errors of orders two, three and four should be zero when we set $\Delta t = 0$. We can then solve explicitly 5 equations in 5 unknowns for the coefficients a_j, obtaining eventually

$$\left.\frac{du}{dt}\right|_{t^n} = \frac{4}{3}\left(\frac{u^{n+1} - u^{n-1}}{2\Delta t}\right) - \frac{1}{3}\left(\frac{u^{n+2} - u^{n-2}}{4\Delta t}\right) + O(\Delta t^4).$$

(3.5) Second order approximation to the second derivative: Using the Taylor expansion method, expanding forward and backwards about t^n, we get

$$\left.\frac{d^2u}{dt^2}\right|_{t^n} = \frac{u^{n-1} - 2u^n + u^{n+1}}{\Delta t^2} + O(\Delta t^2).$$

(3.6) Predictor-corrector methods: These methods overcome the difficulty inherent in implicit methods of the need to evaluate F^{n+1} when u^{n+1} is not yet available. First we make an initial guess $u^* \approx u^{n+1}$, the **predictor**. This might be found, for example, by making a forward Euler method step:

$$u^* := u^n + \Delta t F(t^n, u^n).$$

Then use a trapezoidal interpolation between u^n and u^*, the corrector step, to get

$$u^{n+1} := u^n + \Delta t \frac{F(t^n, u^n) + F(t^{n+1}, u^*)}{2}.$$

With this value of u^{n+1} we get a second order method at roughly twice the cost of the Euler method and using only two preliminary direct evaluations of u^n and u^*.

Acknowledgement: Section 1.10, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(4) Finite volume approximations

To illustrate this method we use the 1D advection plus diffusion equation for temperature $T = T(x,t)$. We use the finite volume method to solve the equation

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} = k_T \frac{\partial^2 T}{\partial x^2},$$

which we write in the form

$$\frac{\partial T}{\partial t} + \frac{\partial q}{\partial x} = 0 \text{ where the "flux" } q = uT - k_T \frac{\partial T}{\partial x},\ q = q(x,t).$$

Label distinct points in the domain of the flow by

$$x_{\frac{1}{2}} \leq \cdots x_{j-\frac{1}{2}} < x_{j+\frac{1}{2}} \leq x_{m-\frac{1}{2}}$$

and set $I_j := [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ the jth "cell", and integrate with respect to x over I_j to get

$$\frac{d}{dt}\int_{I_j} T\ dx + q_{j+\frac{1}{2}} - q_{j-\frac{1}{2}} = 0, \text{ where } q_{j\pm\frac{1}{2}}(t) := q(x_{j\pm\frac{1}{2}}, t).$$

Fixing t and defining $\Delta x_j := x_{j+1/2} - x_{j-1/2}$ we define an average temperature

$$\overline{T}_j(t) = \frac{1}{\Delta x_j}\int_{I_j} T\ dx.$$

Fixing j we then get an exact equation for the time evolution of the average temperatures

$$\frac{d\overline{T}_j}{dt} + \frac{q_{j+\frac{1}{2}} - q_{j-\frac{1}{2}}}{\Delta x_j} = 0.$$

Next, using a subscript for the nth time point $t^n = n\Delta t$ and integrating over $t \in [t^n, t^{n+1}] =: K_n$, with $\Delta t_n := t^{n+1} - t^t$, we get with fixed j

$$\overline{T}_j^{n+1} - \overline{T}_j^n = \frac{\int_{K_n} q_{j+\frac{1}{2}}\ dt - \int_{K_n} q_{j-\frac{1}{2}}\ dt}{\Delta x_j}.$$

Setting the time average of the flux to

$$\widehat{q}_{j\pm\frac{1}{2}} := \frac{1}{\Delta t_n}\int_{K_n} q_{j\pm\frac{1}{2}}\, dt,$$

we get the final form of the discretized temperature evolution equation in 1D, namely

$$\frac{\overline{T}_j^{n+1}-\overline{T}_j^n}{\Delta t_n} + \frac{\widehat{q}_{j+\frac{1}{2}}-\widehat{q}_{j-\frac{1}{2}}}{\Delta x_j} = 0,\ 1\le j\le m-1,\ 0\le n\le N.$$

Note that all terms in this system of equations are numerical and exact. To close this system, we need to be able to calculate the $\hat{q}$ in terms of the $\bar{T}$, which requires approximations. Howver, the exact system is very useful because of its easy to derive conservation properties. For example it exhibits local conservation of flux across grid cell spacial boundaries. Furthermore, the equations when manipulated and added show that

$$\frac{d}{dt}\int_{x_{\frac{1}{2}}}^{x_{m-\frac{1}{2}}} T\ dx = \widehat{q}_{\frac{1}{2}} - \widehat{q}_{m-\frac{1}{2}}.$$

That is to say we get global conservation in that the total heat content evolves over time according to the flux of heat at the boundaries of the domain.
Acknowledgement: Chapter 3, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

Acknowledgement: Section 3.8, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(5) Von Neumann stability analysis for turbulent diffusion:

In this section we study the stability of numerical methods using so-called von Neumann stability analysis. This analyses the stability of a generic Fourier mode. A method is stable is all of the applicable modes are stable.
(5.1) Euler method stability: Consider a PDE for the one spacial dimension diffusion of a tracer concentration $c(t,z)$ with diffusivity κ

$$\frac{\partial c}{\partial t} = \kappa\frac{\partial^2 c}{\partial z^2}$$

on a domain $(t, z) \in [0, T] \times [0, h]$. Choose $\Delta z > 0$ and $\Delta t > 0$ and discretize $[0, h]$ with equally spaced points $z_j = (j - 3/2)\Delta z,\ j = 1, \ldots, m$ so $\Delta z = h/(m-2)$. Choose boundary conditions of the so-called Neumann type so

$$\frac{\partial c}{\partial z} = 0, \text{ at } z = 0, h \text{ for all } t \geq 0.$$

Choose an initial condition which is a single Fourier mode with fixed $j \geq 0$ and constants $C_0,\ C_1$

$$c(z, 0) := C_0 + C_1 \cos\left(\frac{j\pi z}{h}\right),\ j \geq 0,\ 0 \leq z \leq h.$$

Then the form

$$c(z, t) := C_0 + C_1 \cos\left(\frac{j\pi z}{h}\right) exp\left(-\frac{j^2\pi^2\kappa t}{h^2}\right),\ j \geq 0,\ 0 \leq z \leq h,\ t \geq 0,$$

is a solution to the system which satisfies the initial and boundary conditions. We call this the "the analytic solution". Fourier analysis establishes the space of solutions generated in a suitable topology by linear combinations of these j dependent solutions. Note that for fixed j the solution tends exponentially in time to the constant value C_0, which is the behaviour expected of diffusion with as usual $\kappa > 0$.

If we discretize the PDE using the second order numerical method for the second derivative in space and the explicit Euler method in time, we obtain solutions which grow exponentially in time. We will show how von Neumann stability analysis reveals this behaviour and the values of $\Delta t,\ \Delta z$ for which the solutions are unstable, without solving the equation.

First we recast the trial solutions, replacing z by $\kappa\Delta z$ and t by $n\Delta t$ to get the form

$$\begin{aligned} \widetilde{c}_j^n &:= A \exp(\mathrm{i}(k_z j\Delta z - \omega n\Delta t) \\ &= A \exp(\omega_i n\Delta t) \exp(\mathrm{i}(k_z \Delta z j - \omega_r \Delta t n) \\ &= \rho^n \exp(\mathrm{i}k_z \Delta z j), \end{aligned}$$

where A is complex, k_z real and positive, $\omega = \omega_r + i\omega_i$ complex, and $\widetilde{c}_j^n$ is a new variable representing the discrete solution. If $\omega_i > 0$ then the solutions will grow exponentially in time. Here we define the complex **amplification factor** ρ by

$$\rho := \exp(\Delta t(\omega_i - i\omega_r)) \implies |\rho| = exp(\omega_i \Delta t) \text{ and thus } \omega_i = \frac{\ln|\rho|}{\Delta t}.$$

Therefore stability corresponds to $|\rho| \leq 1$ so $-1 \leq \rho \leq 1$ if ρ is real.
If we substitute the expression in terms of the amplification factor for $\widetilde{c}_j^n$ into the discretized diffusion equation with the second order in space method for the second derivative and explicit Euler method for the time derivative of $\widetilde{c}$ and set

$$D = \frac{\kappa \Delta t}{(\Delta z)^2},$$

we get for the discretized equations

$$\widetilde{c}_j^{n+1} = \widetilde{c}_j^n + D(\widetilde{c}_{j+1}^n - 2\widetilde{c}_j^n + \widetilde{c}_{j-1}^n).$$

Substituting the form derived previously, namely $\widetilde{c}_j^n = \rho^n \exp(\mathrm{i} k_z \Delta z j)$, into these equations and simplifying by cancelling $\rho^n \exp(i k_z \Delta z j)$, we get

$$\rho = 1 - 4D \sin^2 \left(\frac{k_z \Delta z}{2} \right).$$

For stability, in particular we must satisfy the necessary condition

$$2D \sin^2(k_z \Delta z/2) \leq 1.$$

Stability for all wave numbers k_z requires in particular for those which satisfy

$$\frac{k_z \Delta z}{2} = \frac{\pi}{2} \pm n\pi,$$

so we must have $D \leq \frac{1}{2}$, for this mixed second order/first order method. In other words, that $\kappa \Delta t/(\Delta z)^2 \leq \frac{1}{2}$.

Quite a lot of information can be extracted from this condition. For example, if particular values $\Delta t,\ \Delta z$ satisfy it, but more accuracy is needed, say by decreasing Δz by a factor of 10, then the time step must be reduced by a factor of 100 to maintain the value of D. Thus the method would required number of computation steps should be increased by a factor of 1000!
We can also obtain information from the value of τ which is defined, for a particular Fourier mode, the ratio of the discrete solution and the analytic solution of the original equation. To see first note

$$c = A \exp(\mathrm{i}(k_z z - \omega t) \text{ with } \frac{\partial c}{\partial t} = \kappa \frac{\partial^2 c}{\partial z^2}$$

gives for the analytic damping coefficient the expression

$$\omega_i = -\kappa k_z^2.$$

For the numerical coefficient we have seen

$$\omega_i = \frac{\ln|\rho|}{\Delta t} = \frac{\ln\left|1 - 4D\sin^2\left(k_z\Delta z/2\right)\right|}{\Delta t}.$$

This gives the ratio of the numerical damping factor to the analytic damping factor, using $D = \kappa\Delta t/\Delta z^2$, in the form

$$\tau := \frac{\text{numerical damping}}{\text{analytic damping}} = \frac{-\ln\left|1 - 4D\sin^2\left(k_z\Delta z/2\right)\right|}{Dk_z^2\Delta z^2}.$$

Expanding the sin and ln in a series about 0 then gives after some manipulation and simplifying

$$\tau = 1 + \left(2D - \frac{1}{3}\right)\left(\frac{k_z\Delta z}{2}\right)^2 + + O(k_z^4\Delta z^4).$$

If $D < 1/6$ then asymptotically we would have $\tau < 1$ so the numerical method would dampen at a rate per iteration n which is slower than the analytic mode. If however $D > 1/4$ we would have for a stable solution $-1 \leq \rho < 0$ for large k_z. Thus the amplitude sequence $\rho^1, \rho^2, \ldots$ would alternate in sign, leading to an oscillating and thus unphysical mode. To avoid this spurious behaviour in the method therefore we should ensure $1/6 < D < 1/4$, which is not at all obvious from the discrete equations of the method. Given the value of κ_E for a particular application, these conditions could be very difficult to meet.

(5.2) Stability of the Implicit Euler method:

With $D = \kappa\Delta t/\Delta z^2$ and $\widetilde{c}_k^n$ representing the solution at x_k at time step n as before, we for the implicit Euler method the discretized equations

$$\widetilde{c}_k^{n+1} = \widetilde{c}_k^n + D(\widetilde{c}_{k+1}^{n+1} - 2\widetilde{c}_k^{n+1} + \widetilde{c}_{k-1}^{n+1}),\ k = 2, \ldots, m-1.$$

Then with k_z the wave number in the z-axis direction, we have

$$\rho = 1 - 2\rho D(1 - \cos(k_z\Delta z) \implies 0 < \rho = \frac{1}{1 + 4D\sin^2(k_z\Delta z/2)} < 1.$$

Since $\rho = |\rho| < 1$, all modes are stable, so we say the numerical solution is unconditionally stable. From the stability point of view we can make arbitrary positive choices for $\Delta t,\ \Delta z$, but need to make them as small as possible to attain a reasonably close approximation to the analytic solution. We also have, using the analytic damping rate from Step (1) namely $-\kappa k_z^2$,

$$\tau := \frac{\text{numerical damping}}{\text{analytic damping}} = \frac{\ln|1 + 4D\sin^2(k_z\Delta z/2)|}{4D(k_z\Delta z)^2}.$$

If D is small, say $D = 0.1$, then τ is close to 1 and the damping rate of the computed solution is reasonable. However as D increases, say $D = 10$, then τ decreases rapidly as a function of $k_z\Delta z$, which is unsatisfactory.
Note the stability of the method is somewhat countered by the need to solve a sparse system of linear equations to move from time step t^n to step t^{n+1}, an added computational burden. Regarding an example of an approach to solving the discrete system which is needed at each time step, see the notes below on solving large sparse linear systems of equations.

(5.3) Leapfrog method instability:

The leapfrog method has the attraction of being direct. We go directly from step t^{n-1} to step t^{n+1}, using values at t^n only for the space derivative terms. However it has a disadvantage which has not arisen in any of the previous methods we have considered. We have for each k:

$$\widetilde{c}_k^{n+1} = \widetilde{c}_k^{n-1} + 2D(\widetilde{c}_{k+1}^n - 2\widetilde{c}_k^n + \widetilde{c}_{k-1}^n),\ D = \frac{\kappa\Delta t}{\Delta z^2}.$$

$$\rho = \frac{1}{\rho} - 8D\sin^2\left(\frac{k_z\Delta z}{2}\right) \implies \rho^2 + 2b\rho - 1 = 0 \text{ where } b = 4D\sin^2(k_z\Delta z/2) > 0.$$

This equation is quadratic with solutions $\rho = -b \pm \sqrt{b^2+1}$. The solution $\rho = -b + \sqrt{b^2+1} > 0$ is physical. For wave numbers with $k_z\Delta z \ll 1$ we have $b \ll 1$ so ρ is small and positive, but for the other solution $\rho = -b - \sqrt{b^2+1} < -1$ so the solution is unstable, no matter what the value of b. The numerical method will not be able to avoid this spurious solution, so its called unconditionally unstable, which is a pity. There are however ways of avoiding the phenomena, such as using an alternative method for diffusion at an earlier step and leapfrog for other terms at step n, and the like.

(5.4) Stability for multi-dimensional diffusion:
The example equation is

$$\frac{\partial c}{\partial t} = \mathscr{A}\frac{\partial^2 c}{\partial x^2} + \mathscr{A}\frac{\partial^2 c}{\partial y^2} + \kappa\frac{\partial^2 c}{\partial z^2}$$

where $\mathscr{A}$ is the horizontal diffusion coefficient and κ the vertical coefficient. Then, suppressing the indices i, j, k we get

$$\begin{aligned}\widetilde{c}(t^{n+1}, x_i, y_j, z_k) = \widetilde{c}^{n+1} = \widetilde{c}^n &+ \frac{\mathscr{A}\Delta t}{\Delta x^2}(\widetilde{c}^n_{i+1} - 2\widetilde{c}^n + \widetilde{c}^n_{i-1}) \\ &+ \frac{\mathscr{A}\Delta t}{\Delta y^2}(\widetilde{c}^n_{j+1} - 2\widetilde{c}^n + \widetilde{c}^n_{j-1}) \\ &+ \frac{\kappa\Delta t}{\Delta z^2}(\widetilde{c}^n_{k+1} - 2\widetilde{c}^n + \widetilde{c}^n_{k-1}).\end{aligned}$$

Substituting a single mode of the shape

$$\widetilde{c}^n := B\rho^n \exp(\mathrm{i}(iw_x\Delta x + jw_y\Delta y + kw_z\Delta z))$$

for a stable solution we must have necessarily

$$\frac{\mathscr{A}\Delta t}{\Delta x^2} + \frac{\mathscr{A}\Delta t}{\Delta y^2} + \frac{\kappa\Delta t}{\Delta z^2} \leq \frac{1}{2}.$$

For geophysical applications, since diffusion is only really significant in the vertical direction, it is sometimes worthwhile to make the code implicit only in the vertical z direction:

$$\begin{aligned}\widetilde{c}(t^{n+1}, x_i, y_j, z_k) = \widetilde{c}^{n+1} = \widetilde{c}^n &+ \frac{\mathscr{A}\Delta t}{\Delta x^2}(\widetilde{c}^n_{i+1} - 2\widetilde{c}^n + \widetilde{c}^n_{i-1}) \\ &+ \frac{\mathscr{A}\Delta t}{\Delta y^2}(\widetilde{c}^n_{j+1} - 2\widetilde{c}^n + \widetilde{c}^n_{j-1}) \\ &+ \frac{\kappa\Delta t}{\Delta z^2}(\widetilde{c}^{n+1}_{k+1} - 2\widetilde{c}^{n+1} + \widetilde{c}^{n+1}_{k-1}).\end{aligned}$$

Acknowledgement: Section 5.4, “Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(6) Inversion of large sparse matrices arising with implicit methods:

There exist many iterative solvers for the large sparse matrix equations which appear with linear implicit methods. Here $\mathbf{x}$ is a vector of unknowns at nodal points in the domain, $\mathbf{A}$ a matrix of coefficients, and $\mathbf{b}$ a vector of data values, normally including boundary values, sources or sinks. These can be found often in software library black-box routines with names like "Jacobi" or "Gauss-Sidel", with quite simple basic structures. The most difficult and fussy task is constructing mappings between the model variables and a given library routine. Note that in geophysical fluid dynamics only a small number of numerical (not time) iterations is normally needed.
The most important numerical measure to be aware of is the **condition number** of the coefficient matrix $\mathbf{A}$. See
`nhigham.com/2021/06/08/bounds-for-the-matrix-condition-number/`

Example: Let the problem to be solved for $\mathbf{x}$ be $\mathbf{Ax} = \mathbf{b}$. Write

$$\mathbf{A} = \mathbf{B} - \mathbf{C} \iff \mathbf{C} = \mathbf{B} - \mathbf{A},$$

where B is easy to invert. For example one could choose a diagonal matrix with all entries nonzero. Choose a starting vector $\mathbf{x^{(0)}}$ and set up the iterative method in the form

$$\mathbf{Bx^{(n+1)}} = \mathbf{Cx^{(n)}} + \mathbf{b} \implies \mathbf{x^{(n+1)}} = \mathbf{x^{(n)}} + \mathbf{B^{-1}}(\mathbf{b} - \mathbf{Ax^{(n)}}).$$

If $\mathbf{x}$ is the solution (we are assuming $\mathbf{A}$ is invertible), then letting $n \to \infty$ we get

$$\mathbf{x} = \mathbf{x} + \mathbf{B^{-1}}(\mathbf{b} - \mathbf{Ax}) \implies \mathbf{Ax} = \mathbf{b}.$$

This shows the iterative method gives the true solution in the limit.

(7) Advective and diffusive flows in 1D:

Here we consider the equation for the concentration of a tracer, which might be for example heat or salt, but not u, v, w:

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} + w\frac{\partial c}{\partial z} = \frac{\partial}{\partial x}\left(\mathscr{A}\frac{\partial c}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mathscr{A}\frac{\partial c}{\partial y}\right) + \frac{\partial}{\partial z}\left(\nu_E\frac{\partial c}{\partial z}\right) + A - Bc,$$

where A, B are constants. This involves considerable additional complexity than equations considered so far. Methods and issues to do with its numerical analysis will be considered in this section. First we consider its scale analysis.
Paramount is the relative scale of the advective terms and the diffusion terms. Let U be a horizontal speed scale, Δc a scale for the change in concentration and K a diffusivity scale i.e. for $\mathscr{A}$ or ν_E. Then

$$\frac{\text{advection}}{\text{diffusion}} = \frac{U\Delta c/L}{K\Delta c/L^2} = \frac{UL}{K} =: Pe.$$

The dimensionless ratio Pe is called the **Peclet number**. If $Pe \ll 1$ (typically $Pe \leq 0.1$ then diffusion is significantly larger than advection, and we can, at least in a preliminary analysis, drop the advective terms from the equation. If $Pe \gg 1$ (typically $Pe \geq 100$) then the reverse applies and we can drop the diffusion terms from the equation. We may need to consider situations in which diffusion is important only in one direction, say the vertical and advection only in the horizontal. For GFD flows generally advection dominates diffusion.

Acknowledgement: Chapter 6, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(8) Highly diffusive 1D flows:

We consider the example of 1D flows in the x-direction, with u positive and constant with no source or sink ($A = B = 0$) and constant diffusivity $\mathscr{A}$. Then the equation simplifies to the form

$$u\frac{dc}{dx} = \mathscr{A}\frac{d^2c}{dx^2} \quad \Longrightarrow \quad c(x) = C_0 + C_1\exp(ux/\mathscr{A}).$$

If the domain of the flow is $[a, b]$, then because of the exponential increase at the boundary b, we have a boundary layer of "thickness" $\mathscr{A}/u$. If this is smaller than the grid size then diffusion must be neglected, at least near the downstream boundary $x = b$. Thus, if we retained the diffusive term for a generally highly advective flow,

near a relevant boundary extra care must be taken since the diffusive term will dominate in that region.

Acknowledgement: Section 6.4, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(9) Finite volumes and leapfroging for purely advective 1D flows:

Consider the discretization of a highly advective 1D flow, completely ignoring diffusion with initial condition $c = c_0(x)$ and assume $u > 0$ is constant. The equation and its analytic solution is

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = 0 \implies c = c_0(x - ut).$$

Integrating with respect to x from $x_{i-1/2}$ to $x_{i+1/2}$ with the flux $q_{i-1/2} := uc|_{i-1/2}$, and with $\overline{c}_i$ the average value of the concentration c over the ith cell $\left[x_{i-1/2}, x_{i+1/2}\right]$, we get

$$\frac{d\overline{c}_i}{dt} + \frac{q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}}{\Delta x} = 0.$$

To relate the fluxes $q_{i-1/2}$ to the average concentrations $\overline{c}_i$ over the i cell, we replace the flux values $q_{i\pm\frac{1}{2}}$ with the discretized approximations

$$\widetilde{q}_{i-\frac{1}{2}} := -u \times \left(\frac{\overline{c}_i - \overline{c}_{i-1}}{2}\right) \implies \frac{d\overline{c}_i}{dt} = -u\left(\frac{\overline{c}_{i+1} - \overline{c}_{i-1}}{2\Delta x}\right).$$

Summing i over all cells we get cancellation apart from the first and last cells. This means the total amount of tracer is conserved. Multiplying by $\overline{c}_i$ on the right and summing we get

$$\frac{d}{dt}\left(\sum_i (\overline{c}_i)^2\right) = -\frac{u}{\Delta x}\sum_i \overline{c}_i\overline{c}_{i+1} + \frac{u}{\Delta x}\sum_i \overline{c}_i\overline{c}_{i-1}$$

This shows variance is conserved also. Note however that at this stage of the derivation time has not been discretized. If we do this, the conservation properties no longer hold.

However, we **claim** that trapezoidal time discretization conserves variance. To see this consider the equation for each cell labelled i with discrete tracer field $\widetilde{c}_i$:

$$\frac{d\widetilde{c}_i}{dt} + \mathscr{L}(\widetilde{c}_i) = 0,$$

where $\mathscr{L}$ is a linear discretization operator which is assumed to satisfy for any tracer field $\widetilde{c}_i$

$$\sum_i \widetilde{c}_i \mathscr{L}(\widetilde{c}_i) = 0,$$

Discretizing the time derivative in a trapezoidal manner and using the linearity of $\mathscr{L}$ we get

$$\frac{\widetilde{c}_i^{n+1} - \widetilde{c}_i^n}{\Delta t} = -\frac{\mathscr{L}(\widetilde{c}_i^{n+1}) - \mathscr{L}(\widetilde{c}_i^n)}{2} = -\tfrac{1}{2}\mathscr{L}(\widetilde{c}_i^{n+1} + \widetilde{c}_i^n).$$

Multiplying by $\widetilde{c}_i^{n+1} + \widetilde{c}_i^n$, summing, and using the assumption on $\mathscr{L}$ applied to the field $(\widetilde{c}_i^{n+1} + \widetilde{c}_i^n)$, we get

$$\sum_i \frac{(\widetilde{c}_i^{n+1})^2 - (\widetilde{c}_i^n)^2}{\Delta t} = -\tfrac{1}{2}(\widetilde{c}_i^{n+1} + \widetilde{c}_i^n)\mathscr{L}(\widetilde{c}_i^{n+1} + \widetilde{c}_i^n) = 0.$$

This method is unconditionally stable and conserves variance. However, there maybe problems with accuracy, a need to solve a system of linear equations at each time step, and to maintain the identity satisfied by $\mathscr{L}$.

(10) Initial and boundary conditions for the leapfrog method for advective 1D flows:

Consider the equation derived earier:

$$\frac{d\overline{c}_i}{dt} + \frac{q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}}{\Delta x} = 0.$$

Let $\widehat{q}_{i-\frac{1}{2}}$ be the time average of the advective flux uc across the cell interface between cells $i-1$ and i over the time interval Δt between t^{n-1} and t^{n+1}. Integrating the given equation with respect to t over this interval we get

$$\overline{c}_i^{n+1} = \overline{c}_i^{n-1} - 2\frac{\Delta t}{\Delta x}(\widehat{q}_{i+\frac{1}{2}} - \widehat{q}_{i-\frac{1}{2}}),$$

where we have used the estimate

$$\widehat{q}_{i-\frac{1}{2}} := \frac{1}{\Delta t}\int_{t^{n-1}}^{t^{n+1}} uc|_{i-\frac{1}{2}}\ dt \implies \widetilde{q}_{i-\frac{1}{2}} = u\left(\frac{\widetilde{c}_i^n + \widetilde{c}_{i-1}^n}{2}\right).$$

Setting

$$C := \frac{u\Delta t}{\Delta x},$$

the so-called **Courant number**, we then get a discretization method for further analysis, namely

$$\widetilde{c}_i^{n+1} = \widetilde{c}_i^{n-1} - C(\widetilde{c}_{i+1}^n - \widetilde{c}_{i-1}^n).$$

Note that unlike other dimensionless numbers, the Courant number will be negative if $u < 0$.

First note that since the solution to the original PDE

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = 0 \implies c = c_0(x - ut),$$

along each line $a = x - ut$ in the (x, t) plane, for a given constant a the value of c is also constant. For $t = 0$, if a is such that $x = a \in [x_0, x_n]$ then the value of c on the line is dependent on the initial condition $c = c_0(x) = c_0(a)$. If however $a < x_0$ then the value is dependent on the upstream boundary condition $c = c_0(x_0 - ut) = c_0(a)$. These lines are called **characteristics**.
To compute from

$$\widetilde{c}_i^{n+1} = \widetilde{c}_i^{n-1} - C(\widetilde{c}_{i+1}^n - \widetilde{c}_{i-1}^n)$$

which was derived above, two initial conditions are needed, the physical value is $\widetilde{c}_i^0$ Using an explicit Euler step from $\widetilde{c}_i^0$ we have

$$\widetilde{c}_i^1 = \widetilde{c}_i^0 - \frac{C}{2}(\widetilde{c}_i^0 - \widetilde{c}_{i-1}^0),$$

which gives the second artificial initial condition.
Mathematically, because of the constancy of solutions along characteristics a boundary condition is required only at the upstream boundary. However, the leapfrog method requires an downstream boundary condition also. In practice an equation consistent with the local discretization at $i = m$, the final cell, is used:

$$\widetilde{c}_m^{n+1} = \widetilde{c}_m^n - C(\widetilde{c}_m^n - \widetilde{c}_{m-1}^n).$$

At the upstream boundary we use the physical condition to get the values of $\widetilde{c}_0^n$. The solution is second order in both x and t other than near the initial condition and the outflow boundary where the order drops by 1. Given the delicacy of this situation, stability analysis is advised for this method.

(11) Stability analysis for the leapfrog method for advective 1D flows:

Let as before using the von Neumann method, substituting a single Fourier mode in the discretized equation we have derived

$$\widetilde{c}_i^{n+1} = \widetilde{c}_i^{n-1} - C(\widetilde{c}_{i+1}^n - \widetilde{c}_{i-1}^n),$$

we get

$$\widetilde{c}_i^n = A\exp(\mathrm{i}(k_x i\Delta x - \omega n\Delta t) \implies \sin(\omega\Delta t) = C\sin(k_x\Delta x).$$

The latter is called the **numerical dispersion relation**. It is a constraint giving a relationship between the wave number and frequency of the mode. For $|C| > 1$ and the wave with $k_x\Delta x = \pi/2$, we get solutions with negative imaginary part for ω, and thus growing amplitude. The method is therefore unstable in this range. If $0 < C \leq 1$ we get two real solutions for ω and thus a stable solution. Hence $|C| \leq 1$ is necessary for stability.
The leapfrog method calculates the value of the unknown function c at point indexed i and time indexed n. This value then depends on values at time $n-1$ and points $i \pm 1$. These in tern require values at time $n-2$ and points $i \pm 2$. We get a triangle of required values in the (x,t) plane. On the other hand the value $\widetilde{c}_i^n$ is completely determined by the unique characteristic $x - ut = x_i - ut^n$. Stability corresponds to this line lying within the triangle of required values. To see this, at the margin for fixed Δx and Δt, for stability the speed $|u|$ needs to be sufficiently small so $|C| \leq 1$. But $1/|u|$ is the slope of the characteristic, so the slope must be sufficiently large, which places the characteristic within the triangle.

(12) Upstream/upwind or donor cell method:

In spite of having some desirable features, the leapfrog method gives poor performance when it comes to accuracy. For example the "top-hat" function with value 1 on a subinterval advects poorly. It seems upstream only information should be used in the numerical method rather than central averaging if we are to model the physical basis of advection. Using an Euler step based on a single time step but calculating the average flow from the grid cell from where the flow comes from gives rise to the **donor cell method**. First we derive the discretization:

$$\widetilde{c}_i^{n+1} = \widetilde{c}_i^n - \frac{\Delta t}{\Delta x}(\widehat{q}_{i+\frac{1}{2}} - \widehat{q}_{i-\frac{1}{2}}), \text{ where } \widehat{q}_{i-\frac{1}{2}} := \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} q_{i-\frac{1}{2}}\, dt \approx u\widetilde{c}_{i-\frac{1}{2}}^n.$$

With the Courant number $C = u\frac{\Delta t}{\Delta x}$ as before, this gives therefore the discrete equation

$$\widetilde{c}_i^{n+1} = \widetilde{c}_i^n - C(\widetilde{c}_i^n - \widetilde{c}_{i-1}^n).$$

Using the von Neumann method or considering the triangle of dependence as we did before we get a necessary condition for stability $0 \leq C \leq 1$. Note that of course $u \geq 0$ in this situation. Note also that there is no need for artificial upwind boundary or for initial conditions. The necessary condition can also be shown to be sufficient.

However, when tested numerically, surprisingly diffusion seems to be inherent in the method. Using Taylor expansions and the original equation, the computed solution satisfies

$$\frac{\partial \widetilde{c}}{\partial t} + u\frac{\partial \widetilde{c}}{\partial x} = \frac{u\Delta x}{2}(1-C)\frac{\partial^2 \widetilde{c}}{\partial x^2} + O(\Delta t^2, \Delta x^2).$$

Hence, unless C happens to equal 1, the computed solution is subject to diffusion. This equation also shows it is only first order in x. A better method would be both second order and reduce the diffusion.

Acknowledgement: Section 6.4, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(13) The Lax-Wendroff method:

The aim of reducing numerical diffusion is achieved with this method. It estimates the flux at a cell boundary by assuming the flux q varies linearly over the cell:

$$\begin{aligned}\widehat{q}_{i-\frac{1}{2}} &:= u\left(\frac{\widetilde{c}_i^n + \widetilde{c}_{i-1}^n}{2} - \frac{C}{2}(\widetilde{c}_i^n - \widetilde{c}_{i-1}^n)\right)\\ &= u\widetilde{c}_{n-1}^n + (1-C)\frac{u\Delta x}{2}\frac{\widetilde{c}_i^n - \widetilde{c}_{i-1}^n}{\Delta x}\\ &\approx u\widetilde{c}_{n-1}^n + (1-C)\frac{u\Delta x}{2}\frac{\partial \widetilde{c}}{\partial x}.\end{aligned}$$

Substituting into the finite volume discretization, namely

$$\widetilde{c}_i^{n+1} = \widetilde{c}_i^n - \frac{\Delta t}{\Delta x}(\widehat{q}_{i+\frac{1}{2}} - \widehat{q}_{i-\frac{1}{2}})$$

applied to c rather than T with some algebra we get:

$$\widetilde{c}_i^{n+1} = \widetilde{c}_i^n - C(\widetilde{c}_i^n - \widetilde{c}_{i-1}^n) - \frac{\Delta t}{\Delta x^2}(1-C)\frac{u\Delta x}{2}(\widetilde{c}_{i+1}^n - 2\widetilde{c}_i^n + \widetilde{c}_{i-1}^n)$$

Note that the new second term in the expression for $\widehat{q}_{i-\frac{1}{2}}$ is aimed at reducing numerical diffusion. The resulting method is second order but subject to dispersion, because we have inadvertently introduced an odd order (third) spacial derivative!

(14) The Crank-Nicholson method:
This method is implicit, so requires the solution to a set of linear equations at each time step. However it is unconditionally stable, but accuracy is low for higher Courant numbers C. It can be derived using the definition

$$\widehat{q}_{i-\frac{1}{2}} := \alpha u\frac{\widetilde{c}_i^{n+1} + \widetilde{c}_{i-1}^{n+1}}{2} + (1-\alpha)u\frac{\widetilde{c}_i^n + \widetilde{c}_{i-1}^n}{2} \text{ with } \alpha = \frac{1}{2}.$$

Monotonic implies first order - the Godunov theorem:

With this tool box of methods the issue arises of whether stability and/or accuracy might be improved by combining methods. Given the linearity of the original equation, provided the coefficients add to 1 methods can be combined. However the stability is not the sum of the stabilities and needs to be calculated in each case. The same applies to accuracy. As for monotonicity (the norm of the discretized solution does not increase in time), we have the result of Godunov (1959), that a consistent numerical method for the advection equation that is monotonic can be at most first order accurate! Note that only the upwind method can be show to be monotonic.

(15) 1D Advection and diffusion with a source/sink term

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = -Sc + \frac{\partial}{\partial x}\left(\mathscr{A}\frac{\partial c}{\partial x}\right).$$

We combine the second order Lax-Wendroff method with zero diffusion with the second order trapezoidal method for the diffusion term. Let as before $C := u\Delta t/\Delta x$ and set dimensionless $B := S\Delta t$. Then we get

$$\widetilde{c}_i^{n+1} = \widetilde{c}_i^n - \frac{B}{2}(\widetilde{c}_i^n + \widetilde{c}_i^{n+1}) - \frac{C}{2}(\widetilde{c}_{i+1}^n - \widetilde{c}_{i-1}^n) + \frac{C^2}{2}(\widetilde{c}_{i+1}^n - 2\widetilde{c}_i^n + \widetilde{c}_{i-1}^n.$$

Using Taylor expansions one can show that this method is only first order accurate unless $u = 0$ or $S = 0$, i.e. it degenerates to one or other of the component methods. There are also problems with stability and monotonicity when methods are combined. This advisory warning (that the stability of the combined method is not easily derivable from that of the component methods) is given, since the situations faced can be quite varied, given the variety of methods available and the number of different term types in the equations. For example, if we combine Lax-Wendroff ($|C| \leq 1$ with the Explicit Euler method ($0 \leq D \leq \frac{1}{2}$), and the explicit method for the sink term ($S \leq 2$), it can be shown that for stability we must have

$$S + 2C^2 + 4D \leq 2,$$

which is more difficult to satisfy than satisfying each of the individual conditions. These would give at the margin $S + 2C^2 + 4D = 2 + 2 + 2 = 6$.

Acknowledgement: Section 6.5, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(16) 2D advective flows - double 1D discretization:

We consider the equation for $c(t,x,y)$ where c represents the concentration of a "tracer" such as heat or salt:

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} = 0, \quad \text{with } u, v \text{ constant.}$$

If $c_0(x,y)$ is the initial condition then $c_0(x-ut, y-vt)$ is the analytic solution. Using a combination of two 1D methods is unsatisfactory since these take information from nodal points on lines parallel to the axes whereas the flow needs information along the direction (u,v). There are methods which reflect this, for example the so called **Corner Transport upstream method.** To describe this method we need two Courant numbers:

$$C_x := \frac{u\Delta t}{\Delta x},\ C_y := \frac{u\Delta t}{\Delta x}.$$

Using the finite volume approach we next define

$$\begin{aligned}
\widehat{q}_{x,i-\frac{1}{2},j} &:= (1 - C_y/2)\tilde{u}\overline{c}^n_{i-1,j} + (C_y/2)\tilde{u}\overline{c}^n_{i-1,j-1}, \text{ and} \\
\widehat{q}_{y,i,j-\frac{1}{2}} &:= (1 - C_x/2)\tilde{v}\overline{c}^n_{i,j-1} + (C_x/2)\tilde{v}\overline{c}^n_{i-1,j-1}, \text{ leading to} \\
\widetilde{c}^{n+1}_{i,j}\widetilde{c}^n_{i,j} &- C_x(\widetilde{c}^n_{i,j} - \widetilde{c}^n_{i-1,j}) - C_y(\widetilde{c}^n_{i,j} - \widetilde{c}^n_{i,j-1}) \\
&+ C_xC_y(\widetilde{c}^n_{i,j} - \widetilde{c}^n_{i-1,j} - \widetilde{c}^n_{i,j-1} + \widetilde{c}^n_{i-1,j-1} \\
&= (1 - C_x)(1 - C_y)\widetilde{c}^n_{i,j} + (1 - C_y)C_x\widetilde{c}^n_{i-1,j} \\
&+ (1 - C_x)C_y\widetilde{c}^n_{i,j-1} + C_xC_y\widetilde{c}^n_{i-1,j-1}.
\end{aligned}$$

I can be seen (provided the Courant numbers are not greater than 1 so none of the coefficients are negative) that the method is monotonic. This method has less distortion but still dampens excessively because of numerical diffusion

Acknowledgement: Section 6.6, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(17) Operator splitting methods for 2D advection - Strang splitting:

These methods similar to predictor/corrector and provide easy ways of building on 1D methods. For example let a discretization be written in the form

$$\frac{d\widetilde{c}_i}{dt} + \mathscr{L}_1(\widetilde{c}_i) + \mathscr{L}_2(\widetilde{c}_i) = 0$$

Splitting over time gives an equivalent set of two equations with intermediate values $\widetilde{c}_i^*$:

$$\frac{\widetilde{c}_i^* - \widetilde{c}_i^n}{\Delta t} + \mathscr{L}_1(\widetilde{c}_i^n) = 0,$$
$$\frac{\widetilde{c}_i^{n+1} - \widetilde{c}_i^*}{\Delta t} + \mathscr{L}_2(\widetilde{c}_i^*) = 0.$$

The ordering, x first and then y, for the splitting is unsatisfactory since it introduces a bias into the method. A simple way of obviating this problem is to first split in the x direction and then y and then, for the next time step do the reverse by switching the order of the operators:

$$\frac{\widetilde{c}_i^* - \widetilde{c}_i^n}{\Delta t} + \mathscr{L}_2(\widetilde{c}_i^{n+1}) = 0,$$
$$\frac{\widetilde{c}_i^{n+2} - \widetilde{c}_i^*}{\Delta t} + \mathscr{L}_1(\widetilde{c}_i^*) = 0.$$

For simple examples the method has little diffusion, but some distortion. For more complicated flows, such as those exhibiting shear, the results can be disappointing.

(18) The discrete Poisson equation

This section is a brief summary of well known methods for solving the discrete 2D Poisson equation:

$$\frac{\widetilde{\psi}_{i+1,j} - 2\widetilde{\psi}_{i,j} + \widetilde{\psi}_{i-1,j}}{\Delta x^2} + \frac{\widetilde{\psi}_{i,j+1} - 2\widetilde{\psi}_{i,j} + \widetilde{\psi}_{i,j-1}}{\Delta y^2} = \widetilde{q}_{i,j}.$$

The extension of the methods to 3D is straight forward. It is not the intention of these notes to provide a guide to implementations since these are already extensive in libraries of subroutines, especially those derived from LAPACK and the BLAS. In addition, there are libraries which are optimized for particular architectures, such as GPU, parallel or vector machines. The intention is merely to give an overview of some methods and issues. Later we will describe the Julia package CuArrays.jl, optimized for NVIDIA sourced GPUs.

(18.1) Jacobi method with over relaxation

This is the simplest, and most inefficient, iterative scheme and requires $O(M)$ iterations to obtain a reasonable residual, the difference between the left and right hand sides. We must assign starting values $\widetilde{\psi}^{(0)}_{i,j}$ for the unknowns at each grid point (x_i, y_j) with $k = 0$ and loop on $k \to k+1$ until the residual $\max \epsilon_{i,j}$ is smaller than a preassigned value. The so-called **relaxation parameter** ω is a positive real number, and for stability it turns out we must have $0 < \omega < 2$. Checking for boundary values must be built into the algorithm at every step. Note that we should compute the lines of implemented code in the given order so as not to need store all the residuals.

$$\left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}\right)\epsilon^{(k)}_{i,j} = \frac{\widetilde{\psi}^{(k)}_{i+1,j} - 2\widetilde{\psi}^{(k)}_{i,j} + \widetilde{\psi}^{(k)}_{i-1,j}}{\Delta x^2} + \frac{\widetilde{\psi}^{(k)}_{i,j+1} - 2\widetilde{\psi}^{(k)}_{i,j} + \widetilde{\psi}^{(k)}_{i,j-1}}{\Delta y^2} - \widetilde{q}_{i,j}$$
$$\widetilde{\psi}^{(k+1)}_{i,j} = \widetilde{\psi}^{(k)}_{i,j} + \omega\epsilon^{(k)}_{i,j}$$

(18.2) Gauss-Sidel method with successive over-relaxation (SOR)

If we arrange the algorithm in the Jacobi method to loop across the domain with increasing i, j, computing each line successively at each k value step in the given order, then we could use the updated neighbourhood values of the unknowns immediately. This ensures more rapid convergence, indeed from $O(M)$ to $O(\sqrt{M})$, but the optimal value of ω depends on the shape of the boundary, making a generic algorithm impossible to supply in a library version of SOR.

$$\widetilde{\psi}^{(k+1)}_{i,j} = \widetilde{\psi}^{(k)}_{i,j} + \omega\epsilon^{(k)}_{i,j}$$

$$\left(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2}\right)\epsilon_{i,j}^{(k)}=\frac{\widetilde{\psi}_{i+1,j}^{(k)}-2\widetilde{\psi}_{i,j}^{(k)}+\widetilde{\psi}_{i-1,j}^{(k+1)}}{\Delta x^2}+\frac{\widetilde{\psi}_{i,j+1}^{(k)}-2\widetilde{\psi}_{i,j}^{(k)}+\widetilde{\psi}_{i,j-1}^{(k+1)}}{\Delta y^2}-\widetilde{q}_{i,j}$$
$$\widetilde{\psi}_{i,j}^{(k+1)}=\widetilde{\psi}_{i,j}^{(k)}+\omega\epsilon_{i,j}^{(k)}$$

(18.3) Red-black methods

Linear algebra routines are ideally suited for parallel or vector processing since the same operations are performed on many variables. SOR is not well adapted for parallel processing. So-called **red-black methods** overcome this feature by dividing the domain (if its 2D) into two sets of interlaced nodes, called red nodes and black nodes. SOR is then applied at step k to each set independently (in parallel) using the neighbouring values of the unknowns from the previous step with the alternative colour. Thus we have two Jacobi iterations performed independently and in parallel on two interlaced grids. This results in a significant speedup on suitable hardware.

(18.4) The Steepest descent method

Noting that the left hand side of the discretized Poisson equation is symmetric and positive definite when written in the matrix form $\mathbf{Ax}=\mathbf{b}$, leads to writing the inversion problem as a minimization problem. Define a real number J and its gradient $\nabla_{\mathbf{x}}J$ by

$$J=\tfrac{1}{2}\mathbf{x}'\mathbf{Ax}-\mathbf{x}'\mathbf{b},$$
$$\nabla_{\mathbf{x}}J=\mathbf{Ax}-\mathbf{b},$$

where $\mathbf{x}'$ represents the transpose of the column vector $\mathbf{x}$.
The problem $\mathbf{Ax}=\mathbf{b}$ can be solved for $\mathbf{x}$ by finding the unique minimum of J, which occurs when the gradient vanishes. Typically, $\mathbf{x}$ is a long vector with one dimension for every node in the domain, and finding the exact minimum is expensive and unnecessary. At step k, if $\mathbf{r}$ is the kth residual, let $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}-\alpha r$, where α is a real number chosen to minimize the $k+1$th residual. Indeed, it can be shown that $\alpha=\mathbf{r}'\mathbf{r}/\mathbf{r}'\mathbf{Ar}$. Then pseudo-code for the steepest descent algorithm takes the form:

Initialize:
$\mathbf{x^{(0)}}=\mathbf{x_0}$
loop on k from $k=0$ until the residual $\mathbf{r}$ is sufficiently small

$\mathbf{r} = \mathbf{A}\mathbf{x}^{(k)} - b$

$\alpha = \dfrac{\mathbf{r}'\mathbf{r}}{\mathbf{r}'\mathbf{A}\mathbf{r}}$

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha r$

end of kth loop

However, convergence of the steepest descent algorithm is slow, giving rise to the following so-called conjugate gradient method.

(18.5) Conjugate-gradient method

This is a generalization of the steepest descent method. Instead of minimizing over one real number we consider vectors $\mathbf{x}$ with the shape

$$\mathbf{x} = \mathbf{x}_0 - \alpha_1\mathbf{e}_1 - \cdots - \alpha_M\mathbf{e}_M,$$

where the vectors $\mathbf{e}_j$ are chosen to satisfy $\mathbf{e}_i'\mathbf{A}\mathbf{e}_j = 0,\ i \neq j$. With this choice With an initial value of $k = 0$

Initialize:

$\mathbf{x^{(0)}} = \mathbf{x_0},\ \mathbf{r_0} = \mathbf{A}\mathbf{x_0} - \mathbf{b},\ \mathbf{e_1} = \mathbf{r_0},\ \mathbf{s_0} = \|\mathbf{r_0}\|^{\mathbf{2}}$

Then loop on k from $k = 0$ until the residual $\mathbf{r}$ is sufficiently small

$\alpha_k = \dfrac{\mathbf{e}_k'\mathbf{r}_{k-1}}{\mathbf{e}_k'\mathbf{A}\mathbf{e}_k}$

$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \alpha_k\mathbf{e}_k$

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{A}\mathbf{e}_k$

$s_k = \|\mathbf{r}_k\|^2$

$\mathbf{e}_{k+1} = \mathbf{r}_k + \dfrac{s_k}{s_{k-1}}\mathbf{e}_k$

end of the kth loop

Convergence is normally obtained with this algorithm with $O(M^{3/2}$ iterations.

(18.6) multigrid methods

This approach can give the most efficient Poisson equation solvers. They begin with a coarse grid and find a solution which gives a rough idea of what is happening with the given equations. This is then interpolated across a finer grid to get a new initial $\mathbf{x}^{(0)}$, which is improved with several iterations and so on. Clearly, some experimentation is needed to determine a good rate at which the grid should be refined and the number of steps used for each grid. Apparently, it's possible to get convergence in $O(M)$ steps.

Acknowledgement: Section 7.8, “Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.

(19) Jacobian evaluations

In two dimensions for invicid and non-turbulent flow we can write the equation for quasi-geostrophic flow as

$$\frac{\partial q}{\partial t} + J(\psi, q) = 0 \text{ where } q = \nabla^2\psi + \beta_0 y,$$

and where $\psi(x, y, t)$ is called the pressure stream function and $q(x, y, t)$ the potential vorticity.
We can rewrite the Jacobian determinant in three distinct ways, the first being the normal definition:

$$\begin{aligned} J(\psi, q) &= \frac{\partial \psi}{\partial x}\frac{\partial q}{\partial y} - \frac{\partial \psi}{\partial y}\frac{\partial q}{\partial x} && \text{(a)}, \\ &= \frac{\partial}{\partial x}\left(\psi\frac{\partial q}{\partial y}\right) - \frac{\partial}{\partial y}\left(\psi\frac{\partial q}{\partial x}\right) && \text{(b)}, \\ &= \frac{\partial}{\partial y}\left(q\frac{\partial \psi}{\partial x}\right) - \frac{\partial}{\partial x}\left(q\frac{\partial \psi}{\partial y}\right) && \text{(c)}. \end{aligned}$$

Using a uniform rectilinear grid and introducing the notation

$$\begin{aligned} \widetilde{\psi}_0 &= \widetilde{\psi}(x_i, y_j, t), \\ \widetilde{\psi}_1 &= \widetilde{\psi}(x_{i-1}, y_{j-1}, t), \\ \widetilde{\psi}_2 &= \widetilde{\psi}(x_i, y_{j-1}, t), \\ \widetilde{\psi}_3 &= \widetilde{\psi}(x_{i+1}, y_{j-1}, t), \\ \widetilde{\psi}_4 &= \widetilde{\psi}(x_{i+1}, y_j, t), \\ \widetilde{\psi}_5 &= \widetilde{\psi}(x_{i+1}, y_{j+1}, t), \\ \widetilde{\psi}_6 &= \widetilde{\psi}(x_i, y_{j+1}, t), \\ \widetilde{\psi}_7 &= \widetilde{\psi}(x_{i-1}, y_{j+1}, t), \\ \widetilde{\psi}_8 &= \widetilde{\psi}(x_{i-1}, y_j, t), \end{aligned}$$

we can write second order approximations to the three forms of the Jacobian determinant:

$$\begin{aligned} J^a &= \frac{(\widetilde{\psi}_4 - \widetilde{\psi}_8)(\widetilde{q}_6 - \widetilde{q}_2) - (\widetilde{\psi}_6 - \widetilde{\psi}_2)(\widetilde{q}_4 - \widetilde{q}_8)}{4\Delta x \Delta y}, \\ J^b &= \frac{\Big(\widetilde{\psi}_4(\widetilde{q}_5 - \widetilde{q}_3) - \widetilde{\psi}_8(\widetilde{q}_7 - \widetilde{q} - 1)\Big) - \Big(\widetilde{\psi}_6(\widetilde{q}_5 - \widetilde{q}_7) - \widetilde{\psi}_2(\widetilde{q}_3 - \widetilde{q} - 1)\Big)}{4\Delta x \Delta y}, \end{aligned}$$

$$J^c = \frac{\left(\widetilde{q}_6(\widetilde{\psi}_5 - \widetilde{\psi}_7) - \widetilde{q}_2(\widetilde{\psi}_3 - \widetilde{\psi}_1)\right) - \left(\widetilde{q}_4(\widetilde{\psi}_5 - \widetilde{\psi}_3) - \widetilde{q}_8(\widetilde{\psi}_7 - \widetilde{\psi}_1)\right)}{4\Delta x \Delta y}.$$

To exploit these different forms Arakawa derived a linear combination which (thus) had this second order truncation error, but also satisfied numerically some nice anti-symmetry and conservation laws provided ψ was uniform along boundaries of the 2D domain S or had periodic boundaries, namely

$$\begin{aligned} J(\psi, q) &= -J(q, \psi), \\ \int_S J(\psi, q)\ dS &= 0, \\ \int_S qJ(\psi, q)\ dS &= 0, \\ \int_S \psi J(\psi, q)\ dS &= 0. \end{aligned}$$

Arakawa's jacobian has the form

$$J := \frac{J^a + J^b + J^c}{3}.$$

(19.2) Cross derivatives 2D second order approximation

$$\left| . \frac{\partial^2 u}{\partial x \partial y} \right|_{i,j} \approx \frac{u_{i+1,j+1} - u_{i+1,j-1} + u_{i-1,j-1} - u_{i-1,j+1}}{4\Delta x \Delta y} + O(\Delta x^2 + \Delta y^2).$$

Acknowledgement: Chapter 16 Section 16.7, Appendix C.4, "Introduction to geophysical fluid dynamics: physical and numerical aspects" 2nd edition, by Benoit Cushman-Roisin and Jean-Marie Beckers, AP, Elsevier, 2011.