Computational Astrophysics

E. Larrañaga

Observatorio Astronómico Nacional
Universidad Nacional de Colombia

June 5, 2019

Outline

Partial Differential Equations

- A PDE is a relation between the partial derivatives of an unknown function and the independent variables.
- The order of the highest derivative sets the *order* of the PDE. If the highest derivative is a second derivative, we are dealing with a second-order PDE.
- A PDE is *linear* if it is of the first degree in the dependent variable (i.e. the unknown function) and in its partial derivatives.
- If each term of a PDE contains either the dependent variable or one of its partial derivatives, the PDE is called *homogeneous*. Otherwise it is *non-homogeneous*.

Types of Partial Differential Equations

There are three general types of PDEs

- hyperbolic
- parabolic
- elliptic

Not all PDEs fall into one of these three types, but many PDEs used in practice do.
These classes of PDEs model different sorts of phenomena, display different behavior, and require different numerical techniques for their solution.
It is not always straighforward to see (to show and proof) what type of PDE a given PDE one is dealing with is.

Linear Second Order Differential Equation

$$a\partial^2_{xx}u + b\partial^2_{xy}u + c\partial^2_{yy}u + d\partial_x u + e\partial_y u + fu = g \ , \tag{1}$$

This equation is straightforwardly categorized based on the discriminant,

$$b^2 - 4ac \begin{cases} < 0 & \rightarrow & \text{elliptic,} \\ = 0 & \rightarrow & \text{parabolic,} \\ > 0 & \rightarrow & \text{hyperbolic.} \end{cases} \tag{2}$$

The names come from analogy with conic sections in the theory of ellipses.

Hyperbolic PDEs

Hyperbolic equations in physics and astrophysics describe *dynamical* processes and systems that generally start at some initial time $t_0 = 0$ with some initial conditions.
The equations are then integrated in time.

The prototypical linear second-order hyperbolic equation is the homogeneous wave equation,

$$c^2 \partial^2_{xx} u - \partial^2_{tt} u = 0 \ , \tag{3}$$

where c is the wave speed.

Hyperbolic PDEs

Another class of hyperbolic equations are the *first-order hyperbolic systems*. In one space dimension and assuming a linear problem, we have

$$\partial_t u + A\partial_x u = 0 \ , \tag{4}$$

where $u(x, t)$ is the state vector with s components and A is a $s \times s$ matrix.

The problem is called *hyperbolic* if A has only real eigenvalues and is diagonizable, i.e., has a complet set of linearly independent eigenvectors so that one can construct a matrix

$$\Lambda = Q^{-1}AQ \ , \tag{5}$$

where Λ is diagonal and has real numbers on its diagonal.

Hyperbolic PDEs

An example of these equations is the linear *advection equation*:

$$\partial_t u + v \partial_x u = 0 \ , \tag{6}$$

which is first order, linear, and homogeneous.

Hyperbolic PDEs

Other example is given by the non-linear first-order systems. Consider the equation

$$\partial_t u + \partial_x F(u) = 0 \ , \tag{7}$$

where $F(u)$ is the *flux* and may or may not be non-linear in u. We can re-write this PDE in *quasi-linear* form, by introducing the Jacobian

$$\bar{A} = \frac{\partial F}{\partial u} \ , \tag{8}$$

and writing

$$\partial_t u + \bar{A} \partial_x u = 0 \ . \tag{9}$$

This PDE is hyperbolic if $\bar{A}$ has real eigenvalues and is diagonizable.

The *equations of hydrodynamics* are a key example of a non-linear, first-order hyperbolic PDE system.

Hyperbolic PDEs Conditions

Initial Conditions for Hyperbolic Problems
One must specify $u(x, t = 0)$ (at all x) and (order 1) time derivatives.

Boundary Conditions for Hyperbolic Problems
One must specify either von Neumann, Dirichlet, or Robin boundary conditions:

1. **Dirichlet Boundary Conditions**.

$$\begin{aligned} u(x = 0, t) &= \Phi_1(t) \ , \\ u(x = L, t) &= \Phi_2(t) \ . \end{aligned} \tag{10}$$

Hyperbolic PDEs Conditions

2. **von Neumann Boundary Conditions**.

$$\begin{aligned} \partial_x u(x=0,t) &= \Psi_1(t) \ , \\ \partial_x u(x=L,t) &= \Psi_2(t) \ . \end{aligned} \tag{11}$$

Note that in a multi-D problem ∂_x turns into the derivative in the direction of the normal to the boundary.

3. **Robin Boundary Conditions**.

a_1, b_1, a_2, b_2 be real numbers, not $a_i = b_i = 0$.

$$\begin{aligned} a_1 u(x=0,t) + b_1 \partial_x u(x=0,t) &= \Psi_1(t) \ , \\ a_2 u(x=L,t) + b_2 \partial_x u(x=L,t) &= \Psi_2(t) \ . \end{aligned} \tag{12}$$

Dirichlet and von Neuman boundary conditions are recovered if either both a_i or both b_i vanish. Note that in a multi-D problem ∂_x turns into the derivative in the direction of the normal to the boundary.

Parabolic PDEs

Parabolic PDEs describe processes that are slowly changing, such as the slow diffusion of heat in a medium, sediments in ground water, and radiation in an opaque medium. Parabolic PDEs are second order and have generally the shape

$$\partial_t u - k\partial^2_{xx} u = f \ . \tag{13}$$

Parabolic PDEs Conditions

Initial Conditions for Parabolic Problems
One must specify $u(x, t = 0)$ at all x.

Boundary Conditions for Parabolic Problems
Dirichlet, von Neumann or Robin boundary conditions. If the boundary conditions are independent of time, the system will evolve towards a steady state ($\partial_t u = 0$). In this case, one may set $\partial_t u = 0$ for all times and treat Eq. (13) as an elliptic equation.

Elliptic PDEs

Elliptic equations describe systems that are static, in steady state and/or in equilibrium. There is no time dependence. A typical elliptic equation is the Poisson equation:

$$\nabla^2 u = f \ , \tag{14}$$

which one encounters in Newtonian gravity and in electrodynamics. ∇^2 is the Laplace operator, and f is a given scalar function of position. Elliptic problems may be linear (f does not depend on u or its derivatives) or non-linear (f depends on u or its derivatives).

Elliptic PDEs Conditions

Initial Conditions for Elliptic Problems
Do not apply, since there is no time dependence.

Boundary Conditions for Elliptic Problems
Dirichlet, von Neumann or Robin boundary conditions.

Numerical Methods for PDEs

There is no such thing as a general robust method for the solution of generic PDEs. Each type (and each sub-type) of PDE requires a different approach. Real-life PDEs may be of mixed type or may have special properties that require knowledge about the underlying physics for their successful solution.

There are three general classes of approaches to solving PDEs

1. Finite Difference Methods.
The differential operators are approximated using their finite-difference representation on a given grid. A sub-class of finite-difference methods, so-called finite-volume methods, can be used for PDEs arising from conservation laws (e.g., the hydrodynamics equations).
Finite difference/volume methods have polynomial convergence for smooth functions.

2. Finite Element Methods.
The domain is divided into cells ("*elements*"). The solution is represented as a single function (e.g., a polynomial) on each cell and the PDE is transformed to an algebraic problem for the matching conditions of the simple functions at cell interfaces. Finite element methods can have polynomial or exponential convergence for smooth functions.

3. Spectral Methods.
The solution is represented by a linear combination of known functions (e.g. trigonometric functions or special polynomials). The PDE is transformed to a set of algebraic equations (or ODEs) for the amplitudes of the component functions. A sub-class of these methods are the collocation methods. In them, the solution is represented on a grid and the spectral decomposition of the solution in known functions is used to estimate to a high degree of accuracy the partial derivatives of the solution on the grid points.

Linear Advection Equation

Linear Advection Equation

Consider the linear advection equation,

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0 \ . \tag{15}$$

This is the simplest example of a hyperbolic equation and its exact solution is simply given by

$$u(t, x) = u(t = 0, x - vt) \ , \tag{16}$$

hence, the advection equation does (as one might expect) just translate the given data along the x-axis with constant advection velocity v.

FTCS Discretization

First-order in Time, Centered in Space (FTCS) Discretization gives

$$u_j^{(n+1)} = u_j^{(n)} - \frac{v\Delta t}{2\Delta x}\left(u_{j+1}^{(n)} - u_{j-1}^{(n)}\right) \ . \qquad (17)$$

FTCS Discretization

Introducing $u(x, t^n) = e^{ikx}$ into the discretized equation,

$$\begin{aligned} u_j^{(n+1)} &= e^{ik\Delta x j} - \frac{v\Delta t}{2\Delta x}\left(e^{ik\Delta x(j+1)} - e^{ik\Delta x(j-1)}\right) \;, \\ &= \left(1 - \frac{v\Delta t}{2\Delta x}\left(e^{ik\Delta x} - e^{-ik\Delta x}\right)\right) e^{ik\Delta x j} \;, \\ &= \underbrace{\left(1 - \frac{v\Delta t}{\Delta x} i \sin(k\Delta x)\right)}_{=\xi} e^{ik\Delta x j} \;, \end{aligned} \tag{18}$$

and

$$|\xi| = \sqrt{\xi\xi^*} = \sqrt{1 + \left(\frac{v\Delta t}{\Delta x}\sin(k\Delta x)\right)^2} > 1 \;. \tag{19}$$

Hence, the FTCS method is *unconditionally unstable* for the advection equation!

Upwind Method

We consider now a first-order in space, first-order in time method to test if it could be stable.

The spatial derivative, to first order, has two possiblities:

$$\begin{aligned} \frac{\partial u}{\partial x} &\approx \frac{1}{\Delta x}(u_j - u_{j-1}) \qquad && \text{upwind finite difference,} \\ \frac{\partial u}{\partial x} &\approx \frac{1}{\Delta x}(u_{j+1} - u_j) \qquad && \text{downwind finite difference.} \end{aligned} \tag{20}$$

These are both so-called one-sided approximations, because they use data only from one side or the other of point x_j.

Upwind Method

Coupling one of these equations with forward Euler time integration yields

$$
\begin{aligned}
u_j^{(n+1)} &= u_j^{(n)} - \frac{v\Delta t}{\Delta x}\left(u_j^{(n)} - u_{j-1}^{(n)}\right) \qquad && \text{upwind,} \\
u_j^{(n+1)} &= u_j^{(n)} - \frac{v\Delta t}{\Delta x}\left(u_{j+1}^{(n)} - u_j^{(n)}\right) \qquad && \text{downwind .}
\end{aligned} \tag{21}
$$

Upwind Method

Stability analysis shows that the upwind method is stable for

$$0 \leq \frac{v\Delta t}{\Delta x} \leq 1 \ , \tag{22}$$

while the downwind method is stable for

$$-1 \leq \frac{v\Delta t}{\Delta x} \leq 0 \ , \tag{23}$$

which just confirms that for $v > 0$ one should use the upwind method and for $v < 0$ the downwind method.

Upwind Method

The condition

$$\alpha = \left| \frac{v\Delta t}{\Delta x} \right| \leq 1 \tag{24}$$

is a mathematical realization of the physical causality principle: the propagation of information (via advection) in one time step Δt must not jump ahead more than one grid interval of size Δx.
The demand that $\alpha \leq 1$ is usually referred to as the *Courant-Friedrics-Lewy (CFL) condition*. The CFL condition is used to determine the allowable time step for a spatial grid size Δx for a certain accuracy and stability,

$$\Delta t = c_{\mathrm{CFL}} \frac{\Delta x}{|v|} \ , \tag{25}$$

where $c_{\mathrm{CFL}} \leq 1$ is the CFL factor.

Lax-Friedrich Method

The Lax-Friedrichs method, like FTCS, is first order in time, but second order in space. It is given by

$$u_j^{(n+1)} = \frac{1}{2}\left(u_{j+1}^{(n)} + u_{j-1}^{(n)}\right) - \frac{v\Delta t}{2\Delta x}\left(u_{j+1}^{(n)} - u_{j-1}^{(n)}\right) \ , \qquad (26)$$

and, compared to FTCS, has been made stable for $\alpha \leq 1$ (Eq. 24) by using the average of the old result at points $j+1$ and $j-1$ to compute the update at point j. This is equivalent to adding a dissipative term to the equations (which damps the instability of the FTCS method) and leads to rather poor accuracy.

Leapfrog Method

A fully second-order method is the *Leapfrog* Method (also called the midpoint method) given by

$$u_j^{(n+1)} = u_j^{(n-1)} - \frac{v\Delta t}{\Delta x}\left(u_{j+1}^{(n)} - u_{j-1}^{(n)}\right) \ . \qquad (27)$$

One can show that it is stable for $\alpha < 1$ (Eq. 24). A special feature of this method is that it is *non-dissipative*. This means that any initial condition simply translates unchanged. All modes (in a decomposition picture) travel unchanged, but they do not generally travel at the correct speed, which can lead to high-frequency oscillations that cannot damp (since there is no numerical viscosity).

Lax-Wendroff Method

The *Lax-Wendroff* method is an extension of the Lax-Friedrich method to second order in space and time. It is given by

$$u_j^{(n+1)} = u_j^n - \frac{v\Delta t}{2\Delta x}\left(u_{j+1}^{(n)} - u_{j-1}^{(n)}\right) + \frac{v^2(\Delta t)^2}{2(\Delta x)^2}\left(u_{j-1}^{(n)} - 2u_j^{(n)} + u_{j+1}^{(n)}\right) \quad . \tag{28}$$

It has considerably better accuracy than the Lax-Friedrich method and is stable for $\alpha \leq 1$ (Eq. 24).

Next Class

Poisson Equation