Partial Differential Equations IV: Beyond the Laplace and Finite Difference Methods

Sam Sinayoko

Computational Methods 7

Contents

1 Learning Outcomes

- List the three classical PDE categories and give one example of PDE for each category.
- Explain in your own words one key caracteristics of the solutions in each category.

- List two ways of speeding up the solution of linear system of equations.
- List the two main alternatives to the Finite Difference Method and discuss their strength and weaknesses.

2 Elliptical equations

Solutions to the Laplace equation depend entirely on the boundary conditions. This is a bit like having a drum. If you change the boundaries of the drum, the automatically adopts a different shape. Likewise, changing the temperature of the boundaries instantaneously changes the Laplace equation solution, i.e. the steady state solution temperature throughout the domain. Equations exhibiting this type of behaviour are called *elliptical* equations. Two other common examples:

1. The *Poisson* equation,

$$\nabla^2 T = -\frac{q}{k}, \tag{1}$$

where q is the heat flux and k the thermal conductivity. The Poisson equation is also common in electomagnetism and fluid mechanics. It gives the Electric field for a particular distribution of charges, or the velocity potential given a particular distribution of fluid sinks or sources.

1. The *Helmholtz* equation:

$$\nabla^2 P' + k^2 P' = 0, \tag{2}$$

where P' denotes pressure fluctuations in the frequency domain, and k the wavenumber. The Helmholtz equation governs the propagation of sound or light at a particular frequency through a uniform medium.

3 Parabolic equations

The prototype of parabolic equation is the heat equation

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T, \tag{3}$$

where α is the thermal diffusivity. Parabolic equations are more complex than elliptical equations. Information propagates at constant speed from a particular hot spot, by diffusing slowly through the domain. The diffusion

speed is controlled by α. Diffusion phenomena occur in fluid mechanics (e.g. diffusion of a pollutant in the ocean or in air) through viscosity. They are related to mixing.

This time knowing the information at the boundary isn't enough to solve the equation. As we discussed in the first lecture on PDEs, provided we can discretize the right hand side in the above equation and therefore write it as a function $F(t, T)$, we can treat the resulting equation as an ODE and integrate it using for example Runge-Kutta methods. This is called the *method of lines*. For each time step, we would have to apply our Laplace operator, for example by multiplying A with U using the notations from the previous lecture. If we use an explicit Runge-Kutta method, such as RK4, there is no need to solve a linear system of equations.

4 Hyperbolic equations

The prototype of hyperbolic equations is the wave equation:

$$\nabla^2 P' + \frac{1}{c_0^2} \frac{\partial^2}{\partial t^2} P' = 0, \tag{4}$$

where c_0 denotes the speed of sound. This equation describes how sound pressure fluctuations propagate in the time domain through a quiescent medium. Hyperbolic equations are hence related to wave phenomena.

More generally, hyperbolic equations describe how a particular quantity is transported through a domain. Thus, a typical hyperbolic equation is the conservation equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{5}$$

where, for example, ρ denotes density and $\mathbf{u}$ velocity. This equation describes how mass is conserved in a fluid. Thus, if density increases at a particular location (for example because of an increase in pressure), we can track how this high density region will be *transported* through the flow as it is *convected* by flow velocity field.

It is common to use non-central schemes, such as the upwind scheme in order to discretize spatial equations in this case. The idea behind the upwind scheme is to improve the stability of the scheme by going against the flow direction in the calculation of the partial derivatives in space.

Note that if the flow (or convection) speed is higher than the speed of sound, which is the maximum speed at which information propagates through the flow, then the pressure field can not balance itself quickly enough

and abrupt discontinuities appear in the flow. These are called shocks and are challenging to solve numerically and you need specialized methods such as Riemann solvers.

5 Advanced Partial Differential Equations

In general, partial differential equations can not simply be characterized as elliptic, parabolic or hyperbolic, but tend to exhibit characteristics belonging to two or three of these categories. For example, the Navier-Stokes equations exhibit both hyperbolic and parabolic characters, via the conservation of mass and momentum on the one hand, and diffusion effects due to viscosity on the other hand. It is however useful to recognise some of these characteristics in the equation you are looking at, because this has implications on how best to solve this problem.

6 Order of methods for solving linear systems of equations equations

Elliptic equations and implicit schemes for parabolic and hyperbolic equations require the solution of linear system of equations $AX = B$. There are several ways of speeding up the solution. The first one is to use a more efficient algorithm. The order of a solver can dramatically change the time to a solution – indeed as much or much more than just buying a bigger computer (or waiting for Moore's Law). A comparison of the algorithmic complexity of a variety of linear solvers is given in table 1.

Table 1: Complexity of a variety of algorithms for solving linear system of equations.

Method	Algorithmic Complexity
Jacobi type	$O(n^2)$
Gauss-Seidel	$O(n^2)$ (can be better)
(Optimal) Successive Over-relaxation	$O(n^{1.5})$
Conjugate Gradient	$O(n^{1.5})$ (depending on pre-conditioner)
Sparse LU	$O(n^{1.5})$
Fast Fourier Transform	$O(n \log n)$
Multigrid	$O(n)$

Furthemore, many PDEs produce *sparse matrices*, filled mostly with zeros. There are very efficient sparse solvers available in SciPy's `linalg.sparse`

module, or directly from the GNU Scientifc Library (in C), or the lower level sparse BLAS library (in Fortran).

7 Comparison with other methods

Two other popular methods for discretizing PDEs are teh Finite Volume Method (FVM) and the Finite Element Method (FEM). A comparison of the FDM, FVM and FEM methods is shown in table 2.

Strong, integral and weak forms for the mass conservation equation:

- Strong form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{6}$$

- Integral form:

$$\frac{\partial}{\partial t} \int_V \rho \mathrm{d}V + \int_A \rho \mathbf{u} \cdot \mathbf{n} \mathrm{d}A = 0, \tag{7}$$

which can be obtained from the weak form by integrating over a closed volume and using Gauss's theorem to turn the divergence term into a surface integral. By solving the above integral equation, the FVM is ensures that the conservation law will be satisfied for each cell in the mesh. This is not true for the finite difference method.

- Weak form:

$$\int_V \frac{\partial \rho}{\partial t} w(\mathbf{x}) - \rho \mathbf{u} \cdot \nabla w(\mathbf{x}) \mathrm{d}V + \int_A \rho \mathbf{u} \cdot \mathbf{n} w(\mathbf{x}) \mathrm{d}A = 0, \tag{8}$$

which can be obtained by multiplying the strong form by a test function $w(\mathbf{x})$, and by using integration by Gauss's theorem. The possibility of choosing the test functions is what makes the FEM so powerful and flexible.

8 Further methods

There are other alternatives, such as:

- Spectral methods [7]: http://www.nektar.info/
- Meshgree or meshless methods
- Particle-based methods

- Monte Carlo methods
- ... <insert the new ground break method you may well invent and research>

9 Conclusions

- The three classical PDE categories are
  1. Elliptic equations (e.g. Laplace equation)
  2. Parabolic equations (e.g. Heat equation)
  3. Hyperbolic equation (e.g. Conservation equation or Wave Equation)
- Linear systems can be sped up by using:
  1. A more efficient algorithm
  2. Sparse matrices

and, if possible, both.

- Alternative to the Finite Difference Method (simple and fast but hard to deal with complex geometries), include
  - the Finite Volume Method (physically intuitive and can deal with complex geometries, but makes it hard to implement higher order methods)
  - the Finite Element Method (complex but powerful and flexible, can deal with

10 References

1. LeVeque, Randall J. Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems. SIAM, 2007.
2. G. Smith, Numerical Solution of Partial Differential Equations: Finite Diference Methods, Oxford University Press, 1985.
3. H. Versteeg and W. Malalasekera, An Introduction to Computational Fluid Dynamics. The Finite Volume Method, Longman Scientific & Technical, 1995.

4. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

5. T. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publishers, 2000.

6. O.Zienkiewicz and R.Taylor,The Finite Element Method:The Basis, vol.1, Butterworth and Heinemann, 2000.

7. G.E. Karniadakis and S. Sherwin, Spectral/hp Element Methods for CFD, Oxford University Press, 1999.

Table 2: The second simplest kind of table.

	FDM: Finite Difference Method [1, 2]	**FVM**: Finite Volume Method [3, 4]	**FEM**: Finite Element Method [5, 6]
Formulation	Strong form: the usual PDE (Eq. 6)	Integral form: strong form integrated over a control volume (Eq. 7)	Weak form: integral form with weighting functions (Eq. 8)
Mesh	Structured	Un-structured or structured	Unstructured or structured
Pros	- Simple and Efficient on structured grids; - Easy to implement high order methods; - Good for mathematical analyses on stability, dispersion and dissipation.	- Physical and intuitive: conserves pysical quantities in each cell; - Can deal with complex geometries easily; - Relatively simple and leads to finite difference equation.	- Very flexible and powerful; - Can deal with complex geometries easily. - Can deal with high order methods easily.
Cons	Hard to mesh complex geometries with structured meshes.	Hard to implement higher order methods	- Mathematially sophisticated, less physically intuitive; - Harder to get started and implement efficiently.
Applications	Fluid Mechanics (CFD), Wave Phenomena (e.g. Acoustics, Electromagnetism)	Fluid Mechanics (CFD)	Structural Engineering, Solid Mechanics
Codes	- PencilCode: High order FDM code for turbulence and magneto-hydrdynamics (F90) - Mostly proprietary codes, written in Fortran, C or C++ - I'm working on an Open Source High Level framework with Hans and we're recruiting!	- OpenFOAM : CFD code, especially good for incompressible flows (C++) - ClawPack: especially good for hyperbolic problems (F77 and F90, Python) - FiPy: open source, (Python + compiled solvers from Trilinos or SciPy)	Two outstanding frameworks written in C++ and Python: - Fenics - Firedrake