

Numerical methods for partial differential equations

by Bernard Knaepen & Yelyzaveta Velizhanina

Contents

1. [Philosophy of the course](#)
2. [Outline](#)
3. [Tools](#)

1 Philosophy of the course

Numerous scientific problems can only be addressed through modeling and computer analysis. This stems from the fact that a lot of these problems are complex and we cannot solve the corresponding models using only pen-and-paper.

Many models appearing in engineering or physical applications are mathematically described by partial differential equations (PDEs) and the aim of this course is to provide a practical introduction to the relevant tools to solve them using computer algorithms. By practical we mean that we will not develop the theory behind PDEs and also avoid complex mathematical derivations of the numerical methods we describe below. Although these are beautiful subjects, they are often characterized by a steep learning curve and our objective is to get our hands on solving real problems as quickly as possible. In some places we nevertheless provide some mathematical background about the techniques we describe because they cannot be properly understood otherwise. In each case, we try to keep things simple and written in such a way that the corresponding section may be skipped at first reading.

2 Outline

By the end of the course, the student should have acquired the necessary skills to solve numerically equations such as the heat equation or Schrödinger's equation:

$$\frac{\partial T(r, t)}{\partial t} = \alpha \Delta T(r, t), \quad \text{Heat equation}$$
$$i\hbar \frac{\partial \Psi(r, t)}{\partial t} = \left[\frac{-\hbar^2}{2m} \Delta + V(r, t) \right] \Psi(r, t), \quad \text{Schrödinger equation}$$

We don't further describing those equations here (it's not a problem if you have never heard about them), but we just emphasises that they both contain the laplacian operator

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

that makes them partial differential equations. Our main objective is then to formulate such operators numerically and to solve the resulting problems with computer algorithms. This also requires to complement the equations with suitable boundary conditions and to implement them consistently with the differential operators.

This course covers the following topics:

- **Integration of ordinary differential equations**

Ordinary differential equations (ODEs), unlike partial differential equations, depend on only one variable. The ability to solve them is essential because we will consider many PDEs that are time dependent and need generalizations of the methods developed for ODEs.

- **Finite difference schemes**

In this chapter we describe the most intuitive way of computing spatial derivatives. It heavily relies on the Taylor's expansion of functions and how we can use the information in the neighborhood of a point to compute derivatives at that point.

- **Resolution of partial differential equations**

This part of the course combines the concepts discussed in the first two chapters and describes through examples how they can be used to solve PDEs.

- **Iterative methods**

Many of the techniques involved in solving ODEs or PDEs require to invert matrices. For low dimensional matrices, this can be done through standard elimination methods. For large matrices, such methods are prohibitively expensive and some approximate inversion techniques are needed. In this chapter we describe some of them belonging to the class of iterative methods.

- **Fourier transform**

In the chapter about finite difference schemes we have described an intuitive way to compute spatial derivatives. Here we introduce a more accurate technique that relies on the expansion of the unknown functions using a basis of functions. We illustrate the concepts introduced to solve problems with periodic boundary conditions.

- **Chebyshev discretisation**

We conclude this course by giving a brief introduction on the Chebyshev spectral method. It also relies on the concept of Fourier transforms but the expansion basis is constructed using Chebyshev polynomials. This method is very accurate and well adapted to problems with physical boundaries as opposed to problems with periodic boundary conditions.

3 Tools

The numerical resolution of PDEs obviously requires the use of a programming language and optionally some previously developed software packages. For this course we have decided to use the Python programming language and some of the SciPy (Scientific computing tools for Python) packages. In the next notebook - 00_01_ToolkitSetup - we say a few more words about this and also provide a complete set of instructions to install all the necessary tools required to get started with the course.

In [1]:

```
from IPython.core.display import HTML
css_file = '../styles/notebookstyle.css'
HTML(open(css_file, 'r').read())
```

Out[1]: