M3N10/M4N10/M5N10
Computational Partial Differential Equations

Dr Shahid Mughal (s.mughal@imperial.ac.uk)
30 lectures + Office hours.

Office hours: Tuesday 10 am-11.30am; Friday 11 am: 1.00 pm
Room: Huxley 734.

Recommended books:

1. Finite Difference Computing with PDEs: A Modern Software Approach (Texts in Computational Science and Engineering). Springer 2018 by Langtange H. P. & Linge S.

2. Numerical Solution of Partial Differential Equations: Finite Difference Methods (Oxford Applied Mathematics & Computing Science Series) (Oxford Applied Mathematics and Computing Science Series) Paperback – 16 Jan 1986 by Smith, G. D.

Assessment 100% by Projects:

Project 1: 15 % (Week 2-3), submission date: 7/02/2020
Project 2: 40% (Week 5), submission date: 6/03/2020
Project 3: 45 % (Week 8-9), submission date: 8/04/2020
MSc/MSci: Mastery: 25%), submission date: (to be announced).

These projects should be submitted electronically, via Blackboard.

Programming languages:
Matlab, Python or
You are free to use alternatives: C, C++, Fortran etc.

Prerequisites: None, other than the core modules from years 1 & 2. You will be expected to produce and/or modify programs in (say) Matlab.

Philosophy of the Module: Most of the universe is governed by Partial Differential Equations, but the techniques for solving PDEs analytically are few. Yet nowadays we have incredible computer power, and we can obtain accurate approximations to solutions to most physically relevant problems. An understanding of analytical techniques really should be accompanied by the ability to find numerical solutions when required. This module is important for those considering a future in research, but also for those considering getting a job in a scientific field (including Finance).

It is moderately easy to write inefficient code which solves simple problems adequately. This skill should not be under-related - many problems you will encounter can be dealt with effectively by a "quick and dirty" approach. Yet, if you want to be able to solve important problems quickly, or difficult problems at all, a good understanding of numerical techniques is vital, and is a well-valued talent. This is what this module aims to engender. At its conclusion, you should be able to make a decent stab at previously unsolved Research-level problems.

Assessment: The module will be assessed solely by projects. The plan is as follows: A relatively short project, worth 10%-20% will be set early on, and returned to you with comments. You will not be committed to completing the module at that stage. A second project, worth 30%-40% will be set about week 5 to be handed in in week 8. Submitting the 2nd project will commit you to completing the module. A final project, worth 40% will be set in week 9, for handing in during the Easter vacation. These projects should be submitted electronically, via Blackboard.

During the module, various Matlab codes will be demonstrated in lectures and released on Blackboard. Most of you will choose to modify these codes for the problems set in the projects, but it is quite permissible to write your programs in any language which runs on the IC machines, e.g. Python, C, Maple, FORTRAN.

The projects will involve demonstrating that your codes work to the expected accuracy, obtaining solutions to set problems and discussing the results. The projects may build on one another slightly -- you may be able to use your codes from the first two projects as part of the final project. The final project will be at a high level, the kind of problem which borders on Research level. At the end of the module you should be able to solve difficult problems using the various techniques covered.

More topics will be covered in lectures than will be of direct of use in the Projects, just as some topics do not get assessed in examinations.

General Comments on Project Modules: Most Maths modules are assessed mainly by a summer exam. Some memory is required - in 2-3 hours you demonstrate what you have learned from the lectures and the many hours of revision. In project modules, memory is not a factor, and you can spend a very long time on them if you choose. As a result, average marks on project modules tend to be a little higher. Nevertheless, 100% is not achievable without deep understanding, and you should not expect perfection. In the past, some students have spent too long on their projects and neglected revision of other topics. I can only caution you against this. The Senior Tutor may advise you not to attend too many project modules. Perfection can lead to excessive time spent on the projects, at cost of other courses. Do please bear this in mind.

Collaboration: We do not, of course, wish to discourage discussion between you on any of the mathematical and computational issues underlying this module. However, plagiarism considerations are very important in project modules. The projects really must be produced independently and any help you received MUST be acknowledged in your submissions. You must adhere strictly to the plagiarism guidelines you have been given -- if you are in any doubt about what is permitted you should seek clarification from the Senior Tutor, Dr Ford. The College penalties are exceptionally severe for breaches and this can jeopardise your entire degree. Please do be sensible about this.

Support Classes: Every now and again, problem sessions will occur in lectures. There will also be surgery sessions in office hours. The timings of these will be discussed in lectures. Some limited programming support will be available, but it is not intended that these will be \u201cdebugging sessions.

Module Content:

There is some flexibility as to the exact module content and ordering, but the following should be very close:

- Introduction: How can we solve PDEs on a computer?
- Finite Difference Methods. Basic types of PDEs.
- Well-posedness and the importance of Boundary Conditions.
- Parabolic equations: Explicit & Implicit Schemes. Maximum principle analysis.
- Elliptic equations: Iterative methods: How can they be made faster? Jacobi, Gauss-Seidel, relaxation techniques.
- Multigrid methods, motivation and implementation.
- Hyperbolic equations: characteristics, upwinding, Lex-Wendroff schemes. Non-reflecting boundary conditions, perfectly matched layers.
- Combinations, Extensions and Applications: e.g. advection/diffusion and Navier-Stokes equations. Magnetic Induction and heat transport equations. Domain decomposition, operator splitting.

I hope you enjoy the module!

Equations to ponder over:

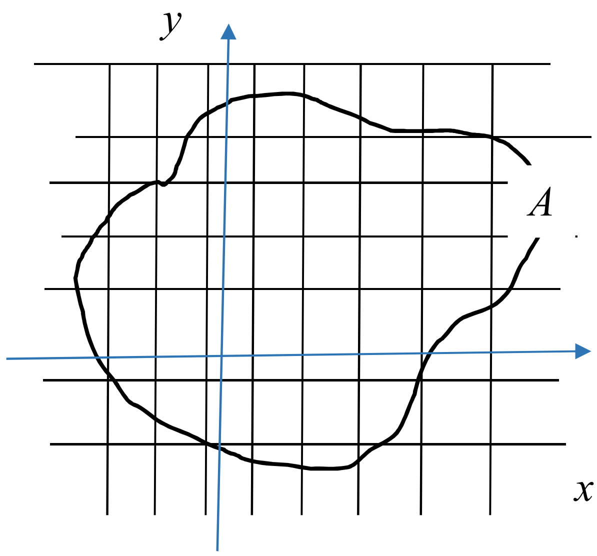
$$\frac{\partial^2\theta}{\partial x^2}+\frac{\partial^2\theta}{\partial y^2}=0;$$

$$\frac{\partial^2\theta}{\partial x^2}+\frac{\partial^2\theta}{\partial y^2}=f(x,y);$$

$$(1-M^2)\frac{\partial^2\theta}{\partial x^2}+\frac{\partial^2\theta}{\partial y^2}=0;\quad \text{Free parameter } M$$

$$\frac{\partial\theta}{\partial t}=\kappa\frac{\partial^2\theta}{\partial x^2};\qquad \text{or}\qquad \frac{\partial\theta}{\partial t}=\kappa\left(\frac{\partial^2\theta}{\partial x^2}+\frac{\partial^2\theta}{\partial y^2}\right);$$

$$\frac{\partial\theta}{\partial t}+U\frac{\partial\theta}{\partial x}=f(x,t)\quad ;$$



+ Appropriate boundary conditions which are to be satisfied at boundaries *(A)*, usually.

θ : field variable, to be determined

t : time variable say

x, y : spatial variables