
Summer School in Numerical Mathematics

Numerical Analysis, Image Processing and Surface Computing

Numerical Analysis is the study of algorithms for solving the problems of continuous science. By “algorithm” we mean a sequence of calculations, typically performed on a computer. This is a broad and exciting subject. This course will introduce (or review, depending on your background) some basic areas of numerical analysis. We will then explore, computationally, several application areas which intersect with cutting-edge Oxford Mathematics research.

Websites

We will have various material online. Start at people.maths.ox.ac.uk/macdonald/CBL and github.com/cbm755/summer-school

Lecturers

- Prof. Colin Macdonald, Associate Professor, Tutorial Fellow at Oriel College, University of Oxford. email: macdonald@maths.ox.ac.uk
- Dr Martin Robinson, Research Software Engineer in Computational Biology, University of Oxford.

Also includes material from Dr Yujia Chen, Dr Ingrid von Glehn, Dr Thomas März and others.

Midterms, projects and presentations

- There will be a midterm exam on Friday 7th August.
- Students will present the results of one or more projects towards the end of the course. More details on projects to follow.

Week 1: Intro to Numerical Analysis

Course description: this first week of the course covers background material in Numerical Analysis. It will consist of lectures and computer demonstrations.

Prerequisites: calculus, basic linear algebra, basic knowledge about differential equations. It will be helpful to have either Matlab or GNU Octave installed on your computer.

Contents: finite differences and interpolation; quadrature; interpolation in barycentric form; numerical solution of differential equations (ODEs and PDEs); software considerations.

Examining: the first week of material will be assessed by a midterm examination.

References: not required, but for additional reading:

LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, SIAM 2007.

Süli and Mayers, An Introduction to Numerical Analysis, CUP 2003.

(And other references, papers and lecture notes to be noted during the course.)

Week 2: Image processing

Course description: Image processing uses mathematics to manipulate digital images (e.g., from a camera or a medical scanning device). The aim of this course is to give an introduction to diffusion PDEs as a means for image processing. Diffusion processes are used to remove noise while preserving or enhancing features such as edges which play an important role in the human perception of an image. In particular we will discuss edge-stopping and edge-enhancing filters based on diffusion and transport models.

The course will be a combination of lectures and practical computing work.

Prerequisites: Multivariable calculus, basic linear algebra, basic knowledge about differential equations.

Contents: The 1D & 2D heat equation, frequency response (low-pass filter), fundamental solution with Fourier Transform, numerical solution with finite difference schemes, MATLAB implementation, unsharp-mask filter. Anisotropic and non-linear diffusion for edge preservation, transport equation and the Osher and Rudin shock filter for edge enhancement.

References:

J. Weickert, “Anisotropic diffusion in image processing”, Teubner Stuttgart, 1998.

T. F. Chan, J. J. Shen, “Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods”, SIAM, 2005.

G. Gilboa, N. Sochen, Y.Y. Zeevi, “Regularized Shock Filters and Complex Diffusion”, ECCV 2002, LNCS 2350, pp. 399-313, Springer-Verlag 2002.

Week 3: Surface computing

Course description: This course introduces numerical solutions of Partial Differential Equations (PDEs) on surfaces using the Closest Point Method.

Surface PDEs arise from many applications in physics, biology, and engineering. Among various numerical techniques for solving surface PDEs, the Closest Point Method is easy to implement and it works for a wide range of PDEs on surfaces with complex geometries.

This course will cover basic theories, numerics and MATLAB implementations related to the method.

Prerequisites: Multivariable calculus, basic linear algebra, basic knowledge about differential equations.

Contents: Applications of surface PDEs, surface differential operators and examples, introduction to the Closest Point Method, matrix formulation of spatial discretizations, diffusion equation on a unit circle, reaction-diffusion on a unit sphere, MATLAB implementations and demos. Closest Point Method: Method of Lines approach, coupled surface-bulk problems

References:

Steven J. Ruuth and Barry Merriman. A simple embedding method for solving partial differential equations on surfaces. *J. Comput. Phys.*, 2008.

Colin B. Macdonald and Steven J. Ruuth. Level set equations on surfaces via the Closest Point Method. *J. Sci. Comput.*, 2008.

Colin B. Macdonald and Steven J. Ruuth. The implicit Closest Point Method for the numerical solution of partial differential equations on surfaces. *SIAM J. Sci. Comput.*, 2009.

Computations and Software

The projects and lectures will involve computation and programming.

Matlab: we recommend Matlab if you already have it.

Octave: alternatively, you can use GNU Octave which is Free/Open Source Software. Octave is included in the package managers for most popular GNU/Linux distributions. For Windows, you should be able to download version 4.0 or later from www.octave.org. Mac users consult internet.

Tablets/phones: one possibility is to use Octave within cloud.sagemath.com. There are also Android apps for Octave and Matlab.

Other software

Python: The Python packages SciPy and NumPy are another possibility (although we will focus on Matlab/Octave for this course). Most of the things we're doing could be easily done in Python as well.

Revision control: like many research groups, we use *Git* for revision control. This is a very useful skill to learn for graduate school or for almost any career involving software. On Windows/Mac you might try the GUI "SourceTree". We may use this software during the course.

L^AT_EX: you might consider using the Beamer class for your presentation. Also a necessary skill for graduate school.