Exeter College Oxford Summer Programme
Numerical Algorithms for Linear Algebra, Optimization, and Deep Learning

Course Description:
This course will explore modern numerical algorithms through three connected tasks: large scale linear algebra, optimization for data science, and deep learning. The first six lectures will discuss how to approximately solve massive scale linear algebra tasks using techniques not covered in linear algebra courses. Examples include how to improve an estimated solution, and why solving for eigenvalues is easier than you may have been lead to believe. The second six lectures will discuss optimization algorithms with a focus on large data science tasks. Numerical optimization is one of the most useful skills as so many tasks from science to business can be cast as optimization problems. The six seminars will focus on deep learning, which is the key algorithmic advance driving the recent advances in machine learning and artificial intelligence. Deep learning is described mathematically with linear algebra and the learning is conducted through numerical optimization such as those which we will explore. The lectures on numerical linear algebra and optimization will ground this course in well understood numerical algorithms which we can study in detail, while the deep learning seminars will give us the opportunity to explore the excitement driving the AI revolution.

Syllabus Overview: Lectures
1. Solving a linear system through matrix factorizations: LU factorization including stability
2. The QR factorization: The Gram-Schmidt and improvements using Householder rotations
3. Iterative numerical methods 1: first order algorithms Orthomin and Steepest Descent
4. dIterative numerical methods 2: higher order algorithms Orthomin(j), CG, and MINRES
5. Solving for eigenvalues and vectors 1: by diagonalization: Jacobi, Upper-Hessenberg, and Divide and Conquer
6. Solving for eigenvalues and vectors 2: the power method and simultaneous iterations
7. Modelling of data science tasks as optimization problems: sparsity, low-rank, etc…
8. Terminology and background material for optimization
9. The method of steepest descent for continuous optimization
10. Stochastic gradient descent for large scale continuous optimization problems
11. Acceleration of gradient decent methods via momentum
12. Coordinate descent methods

Syllabus Overview: Seminars
1. Introduction to a deep network through a classical image classification convolutional architecture.
2. An overview of some of the recent diverse applications of deep learning.
3. The mathematical case for depth as given by approximation theory.
4. How to initialize a deep network: random initialization and its potential pitfalls.
5. Howa deep network is trained: stochastic gradient descent and controlling gradient

magnitudes through depth.
6. Advanced architectures such as those used in the generative networks making up ChatGPT and similar software.

The course comprises 12 lectures, 6 seminars, and 4 tutorials. Both lectures and seminars require the students to read in advance to gain an understanding of the contents to be discussed. The course will help you to sharpen your analytical skills, improve your abilities to critically interpret primary scientific articles, improve your confidence in academic debate, and develop your presentation skills. This course is suitable for students who have a strong interest in and curiosity about how scientific problems are solved computationally with a focus on modern machine learning. The prerequisites are completion of first or second year calculus (including multivariable) and linear algebra; specific topics include understanding of derivatives in more than one variable, Gaussian elimination, and what eigenvalues and vectors are. Some experience with computer programming is also required, but will not be the focus of the course; such experience can be self taught prior to joining this course as the only tools needed are loops and if/else conditionals.

Teaching Methods and Assessment
• 12 .25hr Lectures 15hrs)
• 6 x 1.25hr Seminars7.5hrs)
• 4 .25hr Tutorials 5hrs)

Twice weekly lectures will present the foundational material which will be assessed in the written examination. Weekly seminars will discuss original research articles which the students are expected to have read prior to the seminars; the essay assessment will concern the deep learning material covered in the seminars.

Assessments: Final assessment: An essay of no more than 3,000 words (30%), a final three-hour written examination (60%), and participation in seminar/ tutorials discussions (10%).

Lecture Schedule:

Lecture 1: Introduction to matrix factorization through Gaussian elimination, LU factorization. Issues of computational efficiency and stability will be explored.

Lecture 2: Algorithms for the orthogonalization of a matrix using the Gram-Schmidt process as well as Householder reflections. These different algorithms compute the same factorization, QR, in exact arithmetic, but for large problems their solution on a computer in finite arithmetic can be substantially different. The different computational costs of the algorithms will be determined.

Lecture 3: For the largest linear algebra systems it is too costly to use the traditional LU and QR factorizations. Here we introduce iterative algorithms which seek to improve an approximate solution at each iteration, with a computational cost that is substantially below that of the direct methods. We begin this topic by considering the algorithms orthomin(1) and steepest descent.

Lecture 4: The algorithms from Lecture 3, orthomin(1) and steepest descent, will be improved to make use of prior iterates. The resulting orthomin(j), MINRES, and conjugate gradient methods are some of the most widely used methods for solving linear systems.

Their remarkable convergence properties will be analyzed.

Lecture 5: Eigenvalue and vectors are typically first introduced for small matrices and the eigenvalues shown to be be the roots of low degree polynomials. Here we consider high-dimensional matrices and show that eigenvalues are actually easier to solve for than are the roots of high-degree polynomials. Instead we introduce a set of algorithms that seek to generate a diagonal matrix whose values are the eigenvalues sought. We will consider the Jacobi algorithm, upper-Hessenberg form, as well as the divide-and-conquer algorithm.

Lecture 6: An alternative perspective on eigenvalue and vector calculation is considered in this lecture; that begin the power method and associated simultaneous iteration.

Lecture 7: Data science tasks can often be case as optimization problems. In this lecture we motivate the study of optimization for data science by discussing numerous examples of such formulations, including sparsity and low-rank regularization.

Lecture 8: The terminology and underlying definitions, such as convexity, used in optimization will be stated and discussed in preparation of studying algorithms for optimization.

Lecture 9: The method of steepest descent will be introduced and shown to converge to a global minima of convex functions, or stationary points otherwise. This is a generalization of the algorithm considered in Lecture 3 from solving a linear system to generic optimization problems.

Lecture 10: Stochastic gradient descent is a now widely used variant of steepest descent which is more efficient for large problems where some form of uncertainty is inherent, such as in deep learning. The algorithm will be shown to have similar convergence properties to gradient descent, with the differences highlighted and methods to improve the limitations discussed.

Lecture 11: Acceleration methods, such as Nesterov and heavy-ball, for gradient and stochastic gradient descent will be introduced. These approaches have inherent similarities to the conjugate gradient method discussed in Lecture 4, but are for much more general optimization tasks.

Lecture 12: The method of coordinate descent will be introduced as a way to further improve the computational efficacy for very large optimization tasks. Convergence properties will be analyzed.

GENERAL READING:

Lecture notes will be provided which will be sufficient for understanding of the material covered in the lectures. Recommended, but not required, reading to accompany the provided lectures notes include:

Applied numerical linear algebra by James Demmel, SIAM 1997.
Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB by Amir Beck, SIAM 2014.