1 MA519 : Introduction

This class consists of an overview of the course. The real mathematics starts in the next lecture.

Module materials, including summaries of many lectures, will be available from
http://www.maths.nuigalway.ie/~niall/MA519

These notes are *not* formal, and are unlikely to make much sense if you are not participating in classes.

1.1 Course content

We are concerned with the construction of computer algorithms for solving problems in linear algebra.

Two main types of problem are of interest:

(i) Solving linear systems ($Ax = b$).

(ii) Solving Eigenvalue problems ($Ax = \lambda x$).

Here A is a matrix: usually it is both real and square (most of what we'll study also applies directly to matrices with complex-entries).

Our primary concern is with *sparse matrices*: these are matrices which have comparatively few non-zero entries. The motivation we'll take will come from such matrices as the arise in numerical solution of differential equations, and in analysis of networks.

Starting fundamental ideas of orthogonality, norms and the Singular Value Decomposition, we introduce QR factorisation, and the notion of the condition number of a matrix.

The main ideas we'll study after that are (though not necessarily in this order)

(i) The direct solution of linear systems using Gaussian Elimination, and its variants, but par (e.g., Cholesky Factorisation, Thomas Algorithm).

(ii) Iterative methods for solving linear systems. Although we'll look at classical techniques, such as Jacobi iteration and SOR, our goal will be to understand more advanced techniques, like conjugate gradients.

(iii) Estimation of eigenvalues and eigenvectors.

We'll also look at the implementation of the methods in (say) Matlab.

1.2 Mathematical Preliminaries

You'll need a solid grounding in linear algebra (so you know the fundamentals of manipulating vectors and matrices), as well as standard ideas in algebra: the fundamental theorem of algebra, linear independence, vectors spaces, orthogonality...

A great place to start is the read (and understand and do all the exercises) in Chapter 1 of Ipsen's *Numerical Matrix Analysis: Linear Systems and Least Squares*.

1.3 Solving linear systems

Problems are of the form: *find the set of (real) numbers* $x_1, x_2, \ldots, x_n$ *such that*

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_n \end{aligned}$$

where the a_{ii} *and* b_i *are real numbers.*

It is natural to rephrase this using the language of linear algebra: *Find the vector* $\mathbf{x} = (x_1, x_2, \ldots x_n)^T$ *such that*

$$A\mathbf{x} = \mathbf{b}, \tag{1.1}$$

where A *is an* $m \times n$ *matrix, and* $\mathbf{b}$ *is a (column) vector with* n *entries.*

In this section, we'll try to find clever ways of solving this system, both directly and iterative.

We are primarily interested in the case where $m = n$ and the matrix A is invertible. But we'll also consider what happens if, say, $m > n$ (over determined system) or $m < n$ (under determined).

1.4 The eigenvalue problem

Our other main problem is, given a square matrix A find a scalar λ and nonzero vector $\mathbf{x}$ such that $A\mathbf{x} = \lambda\mathbf{x}$.

In some case, we will want to find *all* the eigenvalues of a matrix. In other cases, we'll just want estimates for the largest and/or smallest eigenvalues.

And sometimes we'll want to compute the eigenvector associated with a certain known eigenvalue (e.g., if A is a stochastic matrix).

1.5 Other considerations

As well as studying/designing algorithms for solving particular problems, we'll consider

- When does the problem have a solution?
- How *stable* is the computation? That is, if there are errors in the data (maybe due to round-off error) are these magnified?
- How expensive is that algorithm? How much computing time or memory resources are required?