Honors Numerical Analysis

$\boxed{\text{Lecture 8}}$

Numerical Linear Algebra

In infinite precision - exact - arithmetic, solving linear systems comes down to a question of whether or not the system is consistent (or not).

On the computer, however, the situation is different due to round-off error and finite precision calculations.

First, a few comments:

Block matrices: $$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & \cdots \\ \vdots & & \end{pmatrix}$$

is notation where each A_{ij} is a matrix.

$$A = (\vec{a}_1 \ \vec{a}_2 \cdots \vec{a}_n)$$

↳ vector.

In Julia, $\vec{a}_j$ can be accessed by A[:,j].

A sub-matrix can be extracted as A[i_1:i_2, j_1:j_2].

```
using LinearAlgebra
a = rand(5,4)
opnorm(a,p)
```
(induced p norm)

also p=Inf is option

Note: at the julia> prompt, type "?" for help, search functions

Gaussian Elimination

We all know how Gaussian elimination works –

Recall: Let $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$

$A\vec{x} = \vec{b}$ can be solved by eliminating a_{21}:

$$\left(\begin{array}{cc|c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{array}\right) \sim \left(\begin{array}{cc|c} a_{11} & a_{12} & b_1 \\ 0 & a_{22} - \frac{a_{12}a_{21}}{a_{11}} & b_2 - \frac{a_{21}a_{12}}{a_{11}} \end{array}\right)$$

This implies $x_2 = \dfrac{b_2 - \frac{a_{21}a_{12}}{a_{11}}}{a_{22} - \frac{a_{21}a_{12}}{a_{11}}}$ and then

$$x_1 = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2\right)$$

This is <u>back-substitution</u>.

But – the process can easily fail: Ex: $a_{11} = 0$ or

$$a_{22} - \frac{a_{21}a_{12}}{a_{11}} = 0$$

A successful algorithm would need to involve pivoting rows.

(will return to this later.)

How expensive is G.E.?

Count flops: "floating point operations" - a single $+, -, \times, \div$

To put A in echelon form:
$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} & \cdots \\ 0 & 0 & \cdots & \end{pmatrix}$$

Consider the algorithm:

Loop over columns $j = 1, \ldots, n-1$

Loop over rows $i = j+1, \ldots, n$

① Compute a_{ij}/a_{jj} (1 flop)

② Compute row i - $\frac{a_{ij}}{a_{jj}}$ row j ($2(n-j)$ flops)

③ Compute $b_i - \frac{a_{ij}}{a_{jj}} b_j$ (2 flops)

The total is obtained by summing:

$$\sum_{j=1}^{n-1} \sum_{i=j+1}^{n} \left(2(n-j) + 3\right) = \sum_{j=1}^{n-1} (n-j)(2(n-j)+3)$$

$$\approx O(n^3) \text{ flops}$$ (work out details at home.)

(see Driscoll & Braun for asymptotic notation, flop counting.)

Another way to think about Gaussian Elimination:

LU Factorization

Each row operation corresponds to multiplication by an "elementary" lower triangular matrix on the left.

Ex: $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{pmatrix}$

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -7 & 0 & 1 \end{pmatrix}}_{L_1} \underbrace{\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{pmatrix}}_{A} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -21 \end{pmatrix}$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}}_{L_2} L_1 A = \underbrace{\begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & -9 \end{pmatrix}}_{U}$$

So $L_2 L_1 A = U$

$\Rightarrow A = \underbrace{L_1^{-1} L_2^{-1}}_{\text{also lower triangular}} U$

Then to solve $\underbrace{A}_{LU}\vec{x} = \vec{b}$

$LU\vec{x} = \vec{b}$

Solve $L\vec{y} = \vec{b}$ FWD substitution (*)

then $U\vec{x} = \vec{y}$ BKWD substitution (**)

(*) and (**) only cost $\mathcal{O}(n^2)$ flops

Pivoting

Of course G.E. or LU may fail if there is a zero or small number in a pivot position.

Solution is to interchange rows, resulting in a factorization of the form $PA = LU$

$\uparrow$ permutation matrix.

Example Let ϵ be a small number, e.g. $\epsilon < \epsilon_{MACH}$

$\uparrow$ machine precision

Then consider $\begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

Exact solution is: $x_1 = 1 + \dfrac{\epsilon}{\epsilon - 1} \simeq 1$

$$x_2 = 1 - \frac{\epsilon}{\epsilon + 1} \simeq 1$$

Without pivoting, in floating point arithmetic, G.E. gives us:

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \sim \left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 0 & -1/\epsilon & -1/\epsilon \end{array}\right)$$

Can be alleviated totally by pivoting, for next time...

$$\Rightarrow \quad x_2 = 1$$

$$\underbrace{x_1 = \frac{1 - 1}{\epsilon} = 0}_{\text{The wrong answer!}} \quad \left(\text{or maybe } \frac{\epsilon_{MACH}}{\epsilon} \right)$$