Lec-9

Algorithm Complexity.

Gaussian elimination w/ backsubstitution is an algorithm to solve the system of eq^{n}s.

$$a_{11}x_1 + \cdots + a_{1n}x_n = b_1$$

$$\vdots$$

$$a_{n1}x_1 + \cdots + a_{nn}x_n = b_n.$$

We want to know how the time taken to complete the algo depends upon the system size, n.

We are particularly interested in large n ($n \gg 1$), which is the regime in which practical applications lie.

Complexity of Gaussian elimination:

Last time we established that G.E. is defined by the following pseudo-code (ignore row exchanges):

"comment"

$i = 1, \dots, n-1$ # choose pivot row
    $j = i+1, \dots, n$ # rows below pivot row
        $m_{ji} = a_{ji} / a_{ii}$ # multiplier
        $R_j \leftarrow R_j - m_{ji} R_i$

How many ops?

$(n+1)-i$ cols (excl. pivot col.)

$$\begin{array}{lcccccc} R_i & 0 \cdots 0 & a_{ii} & \cdots & a_{in} & b_i & \\ & & \text{PIVOT} & & & & \\ R_j & 0 \cdots 0 & a_{ji} & \cdots & a_{jn} & b_j & \Big\} \; n-i \text{ rows} \\ & \vdots & \vdots & & \vdots & \vdots & \end{array}$$

$$\#\text{muls}|\text{divs} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} [\underbrace{1}_{\text{compute } m_{ji}} + \underbrace{1(n+1-i)}_{\text{compute } m_{ji} R_i}]$$

Thus

$$\#\text{muls}|\text{divs} = \sum_{i=1}^{n-1} (n-i)\,\underbrace{[n-i+2]}_{\#\text{ terms in } \sum_{j=i+1}^{n}}$$

$$= \underbrace{\sum_{i=1}^{n-1} (n-i)^2}_{} + \underbrace{2\sum_{i=1}^{n-1} (n-i)}_{}$$

$$(n-1)^2 + (n-2)^2 + \cdots + 1^2$$

$$= 1^2 + \cdots + (n-2)^2 + (n-1)^2$$

$$= \sum_{i=1}^{n-1} i^2$$

$$(n-1) + (n-2) + \cdots + 1$$

$$= 1 + \cdots + (n-2) + (n-1)$$

$$= \sum_{i=1}^{n-1} i$$

Thus

$$\#\text{muls}\,|\,\text{divs} = \sum_{i=1}^{n-1} i^2 + 2\sum_{i=1}^{n-1} i$$

But:

$$\sum_{k=1}^{N} k = \frac{N(N+1)}{2}$$

$$\sum_{k=1}^{N} k^2 = \frac{N(N+1)(2N+1)}{6}$$

Thus

$$\#\text{muls/divs} = \frac{(n-1)(n)(2(n-1)+1)}{6} + 2 \cdot \frac{(n-1)n}{2}$$

$$= \frac{2n^3 + 3n^2 - 5n}{6}$$

when $n \gg 1$, $2n^3 \gg 3n^2 \gg -5n$, thus:

$$\#\text{muls/divs} \simeq n^3/3.$$

eg. double size of system $(n \to 2n) \Rightarrow$ computational cost increases by a factor of $2^3 = 8$.

Similar calculation yields:

$$\#\text{ adds/subs} = \frac{n^3 - n}{3} \approx \frac{n^3}{3}$$

(refers to $n \gg 1$.)

(we determine # mults/divs and #adds/subs seperately because the former take longer to perform than the latter).

Complexity of Back-substitution

Recall:

$i = n, n-1, \ldots, 2, 1$

$$x_i \leftarrow \frac{1}{a_{ii}} \left[b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right]$$

$$\#\text{mults/divs} = \sum_{i=1}^{n} \left[\underbrace{\sum_{j=i+1}^{n} 1}_{\substack{(n-i) \\ \text{terms}}} + 1 \right]$$

$$= \sum_{i=1}^{n} [(n-i)+1]$$

$$= \sum_{i=1}^{n} (n-i) + \sum_{i=1}^{n} 1$$

$$= \sum_{i=1}^{n-1} i + n$$

$$= \frac{(n-1)(n)}{2} + n$$

$$= \frac{n^2+n}{2} \simeq n^2/2.$$

Similarly,

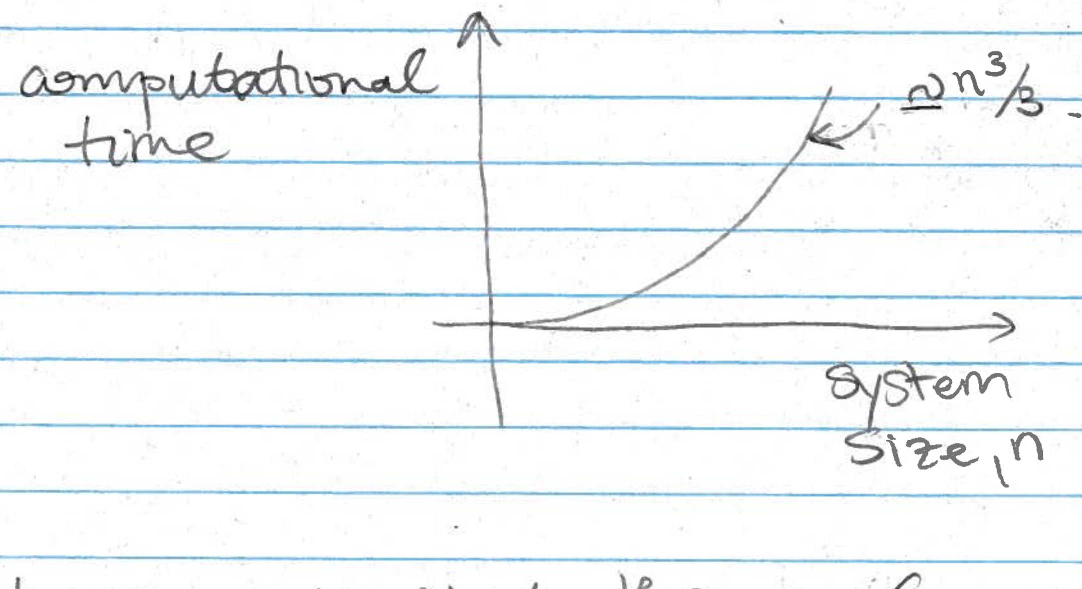
$$\# \text{adds} | \text{subs} = \frac{n^2-n}{2} \simeq \frac{n^2}{2}.$$

Total "time complexity" of Gaussian elimination with back substitution is therefore:

$$\# \text{mults} | \text{divs} \simeq \underbrace{\boxed{\frac{n^3}{3}}}_{\text{G.E.}} + \underbrace{\boxed{\frac{n^2}{2}}}_{\text{B.S.}} \simeq \frac{n^3}{3}$$

$$\# \text{adds} | \text{subs} \simeq \frac{n^3}{3} + \frac{n^2}{2} \simeq \frac{n^3}{3}.$$

Thus:



It turns out that this is (marginally) better than solving the system by matrix inversion [which $\cong \alpha n^3$, $\alpha > 1/3$, cf Ex12(g) §6.3] but (much) worse than iterative techniques (which $\sim \beta n^2$), as we'll see later in the course.