Lecture 19

Floating point and backwards stability
How to tell when an algorithm computes something correct for a slightly wrong input.
Examples where this occurs
- Variance
- Hurwitz zeta.
- PageRank

Then we'll see as much of the backwards stability analysis of LU as we have time for.

I'll outline the results. The rest will largely be tedious algebra. (Triple indices!)

The notes on this on very good, so I encourage you to read them for the algebra.

Next up, more advanced problems!
- Sequences of linear systems
- Generalized eigenvalue problems
- Matrix functions

If $i > j$, then

$$B_{ij}^{(2)} = B_{ij}^{(1)} - \tilde{L}_{i,1}B_{1,j}^{(1)} + \mu_{ij}^{(2)}$$
$$B_{ij}^{(3)} = B_{ij}^{(2)} - \tilde{L}_{i,2}B_{2,j}^{(2)} + \mu_{ij}^{(3)}$$
$$\cdots$$
$$B_{ij}^{(j)} = B_{ij}^{(j-1)} - \tilde{L}_{i,j-1}B_{j-1,j}^{(j-1)} + \mu_{ij}^{(j)}.$$

This is, of course, the same.[3] But we also have:

$$0 = B_{ij}^{(j)} - \tilde{L}_{i,j}B_{jj}^{(j)} + \mu_{ij}^{(j+1)}$$

because B_{ij} becomes o in the $(j+1)$st step. After a similar cancellation of terms, we get:

$$B_{ij}^{(1)} = 0 + \sum_{k=1}^{j} \tilde{L}_{i,k}B_{k,j}^{(k)} + E_{ij}$$

where $E_{ij} = \sum_{k=2}^{j+1} \mu_{ij}^{(k)}$.

Thus, what we compute on the computer is:

$$\tilde{L}\tilde{U} = \begin{bmatrix} 1 & & & & \\ \tilde{L}_{2,1} & 1 & & & \\ \tilde{L}_{3,1} & \tilde{L}_{3,2} & 1 & & \\ \vdots & \vdots & & \ddots & \\ \tilde{L}_{n,1} & \tilde{L}_{n,2} & \cdots & \cdots & 1 \end{bmatrix} \begin{bmatrix} B_{1,1}^{(1)} & B_{1,2}^{(1)} & B_{1,3}^{(1)} & \cdots & B_{1,n}^{(1)} \\ & B_{2,2}^{(2)} & B_{2,3}^{(2)} & \cdots & B_{2,n}^{(2)} \\ & & B_{3,3}^{(3)} & & \vdots \\ & & & \ddots & \vdots \\ & & & & B_{n,n}^{(n)} \end{bmatrix}.$$

Backward Stability

$y = f(x)$ our math target

$\hat{y} = \hat{f}(x)$ our code.

We ideally want

$$\frac{\|y - \hat{y}\|}{\|y\|} \text{ to be small.}$$

But this is hard if f is ill-conditioned.

For ill-cond f, forward Error is not helpful to pick an ALG.

We want algs that guarantee

$$\boxed{\hat{y} = \hat{f}(x) = f(x + \delta x) \text{ for } \delta x \text{ small. } \|\delta x\| / \|x\| \le C_n \underset{\text{machine } \varepsilon}{u}}$$

Defn of Backwards Stable ALG.

$$x \oplus y = (x+y)(1+\delta)$$
$$|\delta| \le u = \text{eps}(1.0)$$
$$= x(1+\delta) + y(1+\delta)$$
$$= x + y + \delta x + \delta y$$

Now, a seq. of float pt. ops $\not\Rightarrow$ Backward Stable ☹

Floating pt gives "16-digits of accuracy"

Is this a lot? or Not?

150 m km to the Sun.

$\approx 1 \times 10^{?}$ m.

A human hair is $\approx$ 20 μm. $\approx 10^{-6}$ m.

$\approx 10^{15}$ hairs from Earth to Sun.

Where did float pt Get me?

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$x =$

$ax^2 + bx + c = 0 \Rightarrow$

$$\frac{-b \pm b\sqrt{1 - 4ac/b^2}}{2a}$$

PageRank.

Given a graph G, the PageRank vector is related to a "random Surfer process"

- At ~~page~~ node i, w/ prob α, we'll go to a random neighbor of i.
- w/ prob $(1-\alpha)$, jump to a random Page.
- If there are no links. → random page

Let X_t be the state of the Surfer at time t.

$$x_i = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \begin{cases} 1 & X_t = i \\ 0 & \text{else.} \end{cases}$$

$$x_i = \alpha \sum_{j \to i} \frac{x_j}{\deg_j} + (1-\alpha) \frac{\overset{1}{\cancel{\sum_{j=1}^{n} x_j}}}{n} + \alpha \sum_{j \,|\, \deg(j)=0} x_j / n.$$

$$\underline{x} = \alpha P^T \underline{x} + (1-\alpha) \frac{ee^T}{n} \underline{x}$$

Where

$$P_{ij} = \begin{cases} 1/\deg(i) & \text{if } i \to j \in E \\ 1/n & \text{if } \deg(i) = 0 \\ 0 & \text{else.} \end{cases}$$

$\Leftrightarrow$

$$(I - \alpha P^T)\underline{x} = (1-\alpha)\frac{e}{n}.$$

Alg:

$$\underline{x}_{k+1} = \alpha P^T \underline{x}_k + (1-\alpha)\frac{e}{n}$$

Let $\underline{x}$ be the true Soln.

$$\underline{x}_{k+1} - \underline{x} = \alpha P^T \underline{x}_k + (1-\alpha)\frac{e}{n} - (\alpha P^T x + (1-\alpha) e/n)$$
$$= \alpha P^T(\underline{x}_k - \underline{x})$$

$$\|x_{k+1} - x\|_1 = \|\alpha P^T(x_k - x)\|_1$$
$$\le \|\alpha P^T\|_1 \cdot \|x_k - x\|_1$$
$$= \alpha \|x_k - x\|_1 = \alpha^k \|x_0 - x\|_1 \le 2\alpha^k$$

If we run this for $\dfrac{2\log(\varepsilon)}{\log(\alpha)}$ steps, we get $\|\text{Error}\|_1 \le \varepsilon$.