# Math 128A, Summer 2019
Lecture 31, Wednesday 8/14/2019

**CLASS ANNOUNCEMENTS:** Final exam tomorrow.

## 1 Review: Final Exam Spring 2019

### 1.1 Problem 5B

We open with a classmate asking about the following problem:

**Problem 5B.** Suppose $A$ is a diagonally dominant $n \times n$ matrx, and a lower-triangular matrix $L$ with diagonal entries equal to 1 and an invertible upper triangular matrix $U$ satisfy:

$$LU = A + F$$

for some $n \times n$ matrix $F$.

(1) Find a lower-triangular matrix $B$ with zero diagonal entries and an upper triangular matrix $C$ such that

$$L(I+B)(I+C)U = A + L\ B\ C\ U.$$

(2) Let $\hat{L} := L(I+B)$ and $\hat{U} := (I+C)U$. Show that the residual

$$\hat{L}\hat{U} - A = O(||F||^2)$$

as $||F|| \to 0$.

Strain opens with a blind shot, eventually getting stuck without using $LU = A + F$ given in the problem. We'll call this 'background' as it is helpful intuition.

### 1.2 Background

Recall that when we solve for the Cholesky factorization, we have many different ways to do this.

$$R^T R = A \implies (R+E)^T(R+E) - A = E^T E$$

In this particular case, we do a weird thing, which is to take the quadratic piece of the LHS and put it on the LHS. We aren't solving for the quadratic anymore but we're solving something more interesting. We're solving for the residual. If the residual is small, then the solution gets more interesting. One way to solve $LU = A$ is to look for corrections such that:

$$(L + \Delta L)(U + \Delta U) = A$$

which will satisfy a coupled quadratic equation (which would be a pain), so we do:

$$(L + \Delta L)(U + \Delta U) = A + \underbrace{\Delta L \cdot \Delta U}$$

via group theory. The residual in the corrected equation which we are trying to solve will be $\Delta L \cdot \Delta U$. However, it is more natural to (instead) take:

$$L(I + \Delta L)(I + \Delta U)U = A$$

Recall that in the related lecture, we had a bunch of 'funny looking stuff' (according to Strain). This was because we did the additive version instead of the multiplicative version (as given this problem).
Now we take (to get a linear equation):

$$L(I + \Delta L)(I + \Delta U)U = A + L \cdot \Delta L \cdot \Delta U \cdot U$$

which gives us an equation to solve:

$$L(\Delta L + \Delta U)U = A - LU$$

Multiplying across, we get:

$$\Delta L + \Delta U = L^{-1}AU^{-1} - I$$

It turns out this gives $n^2$ equations and $n^2$ unknowns. The two matrices $\Delta L$ and $\Delta U$ are orthoginal in that $\Delta L$ is lower triangular and $\Delta U$ is upper triangular. Because they are orthogonal, this suggests that we should mention the Frobenius norm again:

$$||A||_F^2$$

Because we talked about the great things about operator norms before, we'll now talk about the great things about the Frobenius norm.

$$||A||_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2,$$

and another way to think about this is $\langle A, A \rangle$, but notice this is NOT the same thing as $A^T A$. Really what's happening is that we're taking column sums of equared elements, and summing the rows. So we have:

$$||A||_F^2 = \langle A, A \rangle = \sum_{i=1}^{n} \left( \begin{bmatrix} \cdots & \cdots & \cdots \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \vdots & & \\ \vdots & & \\ \vdots & & \end{bmatrix} \right)_{ii} = \mathrm{tr}(A^T A),$$

the trace of $A^T A$. Generally, a good way to define the inner product of two matrices ss :

$$\langle A, B \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ij} = \mathrm{tr}(A^T B).$$

We mention this because in our main problem,

$$\langle \Delta L, \Delta U \rangle = 0.$$

Strain checks again the hypothesis and given information in the problem and starts again.

## 1.3 Solution.

In our problem, we had:

$$LU = A + F,$$

so

$$L(I + \Delta L)(I + \Delta U)U - A = L \cdot \Delta L \cdot \Delta U \cdot U,$$

where the LHS gives us the residual. So this gives:

$$\begin{aligned} LU + L(\Delta L + \Delta U)U - A = 0 \\ L(\Delta L + \Delta U)U = A - LU = -F \\ \Delta L = \text{slt}(-L^{-1}FU^{-1}) \\ \Delta U = \text{upt}(-L^{-1}), \end{aligned}$$

so we have:

$$L \cdot \Delta L \cdot \Delta U \cdot U = L\text{slt}(-L^{-1}FU^{-1})\text{upt}(-L^{-1}FU^{-1})U$$

These matrices slt, upt (strictly lower triangular, upper triangular) are just projections to lower or upper triangular matrices. In an inner product space, projections always decrease the norm. The question is if:

$$||AB||_F \overset{?}{\leq} ||A||_F \; ||B||_F,$$

where we recall $||I||_F = \sqrt{n}$.
It turns out that we need simply need to change one of the norms on the RHS to the 2-norm. That is,

$$||AB||_F \leq ||A||_2 ||B||_F.$$

To see this, consider:

$$||AB||_F^2 = ||Ab_1||_2^2 + \cdots + ||Ab_n||_2^2 \leq ||A||_2^2 ||B||_F^2.$$

So in our original problem, we have that the residual:

$$\text{residual} = R = \hat{L}\hat{U} - A$$

is bounded by:

$$||\hat{L}\hat{U} - A|| \leq \underbrace{\kappa(L)\kappa(U)}\, ||F||^2 \, \underbrace{||U^{-1}|| ||L^{-1}||} = O(||F||^2),$$

where all the other factors are all constants unrelated to $||F||$, so this is our required result (but doesn't look pretty).

## 2 Symmetric Arrowhead Matrices

Now from Spring 2018 Final Exam (as Strain mentions is stolen from Professor Gu), Let

$$A := \begin{bmatrix} d_1 & 0 & \cdots & 0 & r_1 \\ 0 & d_2 & 0 & \cdots & r_2 \\ \vdots & & & & \vdots \\ 0 & 0 & \cdots & d_{n-1} & r_{n-1} \\ r_1 & r_2 & \cdots & r_{n-1} & d_n \end{bmatrix} = D + re_n^T + e_n r^T,$$

where $e_n^T r = 0$, be a symmetric arrowhead matrix. Develop an algorithm for computing the Cholesky factorization fo $A$ in $O(n)$ scalar floating point operations. Use your algorithm to find a condition on the diagonal matrix $D$ and the vector $r$ which determines when the matrix $A$ is positive definite.

### 2.1 Solution.

We call this an arrowhead matrix. We don't want to perform pivoting because that would destroy the nice structure of this matrix. There are a few good ideas, but our best idea here is to try Cholesky factorization.
The question is if in the Cholesky factorization, we have enough degrees of freedom to solve what we need. It turns out that we have enough zeros to 'move' via Gaussian elimination. Our condition on the diagonal is something like:

$$a_{nn} - a_{n1} - a_{n2} - \cdots > 0$$

### 2.2 Algorithm for Computing the Cholesky Factorization of $A$ in $O(n)$

The algorithm would be to take the diagonal entry and 'kill' the last entry in the row. We take:

$$A = D + re_n^T + e_n r^T e_n^T r = 0$$

and we think about a factorization of this. We guess and check $\Lambda$ to be like $D$:

$$R = \Lambda + \rho e_n^T e_n^T \rho = 0$$

so,

$$\begin{aligned} R^T R &= (\Lambda + \rho e_n^T)^T (\Lambda + \rho e_n^T) \\ &= (\Lambda + e_n \rho^T)(\Lambda + \rho e_n^T) \\ &= \Lambda^2 + e_n \rho^T \rho e_n^T + e_n \rho^T \Lambda + \Lambda \rho e_n^T \end{aligned}$$

Strain reads off the result:

$$\Lambda^2 + \rho^T \rho e_n e_n^T = D, \quad r = \Lambda \rho,$$

but we actually have the following for free:

$$\boxed{\lambda_1^2 = d_1 \cdots \lambda_{n-1}^2 = d_{n-1}},$$

and likewise onwards, but the only one we don't know is:

$$\boxed{\lambda_n^2 = d_n - \sum_{j=1}^{n-1} \frac{r_j^2}{d_j}}.$$

To get $\rho$, we use $r = \Lambda \rho \implies \rho = \Lambda^{-1} r$. Our algorithm is actually just to write down the exact expression and evaluate!

```
for i = 1:n-1
  R_ii = sqrt(d_i)

R_nn = sqrt( d_n - sum_{j=1}^{n-1} (r_j)^2/d_j )

for (i = 1:n-1)
  R_in = r_i / R_ii
```

The condition for our matrix to be positive definite is:

$$\partial_n - \sum_{j=1}^{n-1} \frac{r_j^2}{d_j} > 0$$

# 3 Numerical Integration Rules

We want a numerical integration rule where we characterize good points and weights for:

$$\int_0^1 f(t)\ dt = \sum_{j=1}^{n} w_i f(t_i)$$

What's going on here is that the quality of our quadrature rule is to separate these two things.
1) $|f(t) - p(t)| \leq \epsilon$ Find a class of polynomials that achieves this bound for a given $f$.

$$\begin{aligned}\int_0^1 f(t)\ dt &= \int_0^1 f(t) - p(t)\ dt + \underbrace{\int_0^1 p(t)\ dt} \\ &= \int_0^1 \left(f(t) - p(t)\right)\ dt + \sum_{i=1}^{n} w_i \left(p(t_i) - f(t_i) + f(t_i)\right)\end{aligned}$$

so we move a term over and use an infinite number of triangle inequalities to get:

$$\begin{aligned}\left|\int_0^1 f(t)\ dt - \sum_{i=1}^{n} w_i f(t_i)\right| &\leq \int_0^1 |f(t) - p(t)|\ dt + \sum_{i=1}^{n} |w_i|\, \underbrace{|p(t_i) - f(t_i)|}_{\epsilon} \\ &\leq \underbrace{\left(1 + \sum_{i=1}^{n} |w_i|\right)}_{\text{cond. num for quad rule}} \epsilon\end{aligned}$$

As an aside, we can get the total sum of the weights by the condition set upon the integration rule. In our present case we usually test $f = 1$ to get weight sums of 1 or 2 because our quadrature rule makes integration exact for polynomials of degree less than or equal to $n$. That is,

$$0 < \int_{-1}^{1} L_j(t)^2\ dt = \sum_{i=1}^{n} L_j(t_i)^2 w_i = w_j$$

## 3.1 Example: Integration Weights for Exponentials

Suppose $\lambda_j > 0$ for $1 \leq j \leq n$, so:

$$\frac{1}{\lambda_j} = \int_0^\infty e^{-\lambda_j x}\ dx = \int_{i=1}^{n} e^{\lambda_j x_i} w_i,$$

which gives an $n \times n$ linear system we can solve for weights. Doing so gives:

$$\int_0^\infty f(x)\ dx = \int_{i=1}^{n} w_i f(x_i),$$

whenever $f(x)$ is some linear combination of decaying exponentials:

$$f(x) = \sum_{i=1}^{n} f_j e^{-\lambda_j x}$$

### 3.2 Example: ODE, Linear Stability

Let's say we're solving some explicit scheme (so we could see how bad it is).

$$u_{n+1} = u_n + h\left(\frac{3}{2}f_n - \frac{1}{2}f_{n-1}\right),$$

and consider $f(y) = \lambda y$. Then:

$$u_{n+1} = u_n + h\lambda\left(\frac{3}{2}u_n - \frac{1}{2}u_{n-1}\right)$$
$$u_{n+1} = \left(1 + \frac{3}{2}h\lambda\right)u_n - \frac{1}{2}h\lambda u_{n-1},$$

and this looks like a 2-term recurrence relation. Let $z := h\lambda$, and we check the characteristic roots:

$$r^2 - \left(1 + \frac{3}{2}z\right)r + \frac{1}{2}z = 0,$$

and we have a problem (we don't need to look any further). If we have some $r_1, r_2$ roots satisfying this,

$$(r - r_1)(r - r_2) = 0$$

then

$$r_1 r_2 = \frac{1}{2}z,$$

and $z$ will vary and explode if one of the roots is outside the unit circle. This two-step explicit Adams (Bashforth) would not be good for stiff equations.

## 4 Iterative Refinement

The compute bound is that multiplying matrices (inputting $n^2$ inputs costs $n^3$ time), but numerical linear algebraists.
(This got interrupted to answer more questions above about past exam problems and related hypotheticals).

Lecture ends here.

This was the last lecture; good luck on the final exam (lecture/class 32), and have a nice summer!