MAT 228A Notes

Sam Fleischer

October 11, 2016

1 Recall

$$u_{xx} = f \qquad u(0) = \alpha \qquad u(1) = \beta \tag{1}$$

discretize this to get

$$A\vec{u} = \vec{b} \tag{2}$$

where

$$A = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \dots & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix} \tag{3}$$

2 How to Solve the Linear System

2.1 Solve by Gaussian Elimination

$$A = LU = (\text{lower triangular matrix with ones on the diagonal}) \cdot (\text{upper triangular matrix}) \tag{4}$$

So,

$$A\vec{u} = \vec{b} \tag{5}$$
$$LU\vec{u} = \vec{b} \tag{6}$$
$$L\vec{v} = \vec{b} \qquad \text{with } \vec{v} = U\vec{u} \tag{7}$$
$$U\vec{u} = L^{-1}\vec{b} = \vec{v} \tag{8}$$

How expensive (computationally) is this? In general, if A is an arbitrary $n \times n$ matrix, the work is $\mathcal{O}(n^3)$ (costly). But the work to solve the triangular system is $\mathcal{O}(n^2)$. But even better, A is a tri-diagonal matrix, so..

2.2 LU decomposition of tridiagonal matrices

$$\begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ -\frac{1}{2} & 1 & & & \\ & -\frac{2}{3} & 1 & & \\ & & \ddots & \ddots & \\ & & & -\frac{n-1}{n} & 1 \end{pmatrix} \begin{pmatrix} -\frac{2}{1} & 1 & & & \\ & -\frac{3}{2} & 1 & & \\ & & \ddots & \ddots & \\ & & & -\frac{n+1}{n} & 1 \end{pmatrix} \tag{9}$$

Work to factor is $\mathcal{O}(n)$. Work to solve is also $\mathcal{O}(n)$. Tridiagonal solvers are *fast* (Thomas algorithm). In Matlab `y = A\b` But make A sparse!

3 Last time we showed

$$A\vec{e} = \vec{\tau} \tag{10}$$
$$\|A\|_2 = \mathcal{O}(1) \tag{11}$$
$$\|\vec{e}\|_2 = \left\|A^{1-}\vec{\tau}\right\|_2 \leq \left\|A^{-1}\right\|_2 \|\vec{\tau}\|_2 = \mathcal{O}\left(h^2\right) \tag{12}$$

We saw an example where

$$\|\vec{e}\|_\infty = \mathcal{O}\left(h^2\right) \tag{13}$$

Is this true in general? Try norm equivalence?

$$\underbrace{c\|\vec{e}\|_\infty \leq \|\vec{e}\|_2}_{\text{try this}} \leq C\|\vec{e}\|_\infty \tag{14}$$

$$\|\vec{e}\|_2 = \sqrt{h}\left(\sum_{j=1}^{n} e_j^2\right)^{\frac{1}{2}} \geq \sqrt{h}\max_j |e_j| = \sqrt{h}\|\vec{e}\|_\infty \tag{15}$$

So,

$$\sqrt{h}\|\vec{e}\|_\infty \leq \|\vec{e}\|_2 \leq Ch^2 \tag{16}$$

So we get a sloppy bound on the error...

$$\|\vec{e}\|_\infty \leq Ch^{\frac{3}{2}} \tag{17}$$

but we can do better.

3.1 Max-norm analysis

Lets solve

$$A\vec{u} = \vec{b} \tag{18}$$

where

$$\vec{b}_i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{19}$$

where j is fixed.

For $i = 1, \ldots, j-1$ we have

$$\frac{u_{i=1} - 2u_i + u_{i+1}}{h^2} = 0 \qquad u_0 = 0 \qquad u_j = U \tag{20}$$

Solve by guessing a linear function of x_i. For $i < j$,

$$u_i = \frac{U}{x_j} x_i \tag{21}$$

For $i > j$,

$$u_i = \frac{U(1 - x_i)}{1 - x_j} \tag{22}$$

At $i = j$,

$$\frac{u_{j-1} - 2u_j + u_{j+1}}{h^2} = 1 \tag{23}$$
$$\frac{Ux_{j-1}}{x_j} - 2U + \frac{U(1 - x_{j+1})}{1 - x_j} = h^2 \tag{24}$$
$$\implies U = h(x_j - 1)x_j \tag{25}$$

The solution is

$$u_i = \begin{cases} h(x_j - 1)x_i & \text{if } i \le j \\ h(x_i - 1)x_j & \text{if } i > j \end{cases} \tag{26}$$

This is the i^{th} element of the j^{th} column of A^{-1}. This is the discrete version of Green's function.

SO!

$$\left\|A^{-1}\right\|_\infty = \max_i \sum_{j=1}^{n} \left|A_{ij}^{-1}\right| \le nh \le 1 \tag{27}$$

This tells us

$$\|\vec{e}\|_\infty \le \left\|A^{-1}\right\|_\infty \|\vec{\tau}\|_\infty = \mathcal{O}\big(h^2\big) \tag{28}$$

$$A^{-1} = B \tag{29}$$

$$\vec{e} = B\vec{\tau} = \sum_{i=1}^{n} \left(\quad b_i \quad \right) \vec{\tau}_i \tag{30}$$