This is a tri-diagonal matrix which can be rapidly solved using Thomas method.

Example program just uses matlab's A\b command. (a bit slower, but for a small matrix it doesn't matter)

Ok, systems of eq'ns are useful but do we always get the right answer?

Sometimes, have significant numerical error. Let's see where it comes from.

First, we look at the algorithm itself suppose we have the problem:

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}$$

Apply elim. (1^{st} step):

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 2.5 \end{pmatrix}$$

36

At the 2^{nd} step we would mult 2^{nd} row by 25 (lge #) and add to third:

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 0 & 155 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 155 \end{pmatrix}$$

This has the sol'n $x_3 = 1$, but the #'s are getting large - can lead to roundoff errors in lge matrices

What if the 2^{nd} coef. is zero??

Avoid this w/ pivoting

Each row corresp. to an equation, so we just reorder them!

flip 2^{nd} & 3^{rd} rows:

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & -0.1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2.5 \\ 6.1 \end{pmatrix}$$

Now do elimination:

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2.5 \\ 6.2 \end{pmatrix}$$

So $x_3 = 1$, $x_2 = -1$, $x_1 = 0$

In general, in $k^{\underline{th}}$ step of GE, interchange rows so that largest element of $k^{\underline{th}}$ column and $i \geq k$ rows is in $k^{\underline{th}}$ row

Then $\Rightarrow$ $a_{ij}^{(k+1)} = a_{ij}^{(k)} - \left(a_{ik}^{(k)} / a_{kk}^{(k)} \right) a_{kj}^{(k)}$

for $i, j > k$

This operation is equivalent to a $\underset{\sim}{P}\underset{\sim}{L}\underset{\sim}{U}$ factorization!

$\underset{\sim}{L}$ is lower triangular matrix

$\underset{\sim}{U}$ is an upper triangular matrix

$\underset{\sim}{P}$ corresp. to pivoting. So:

$$\underbrace{\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix}}_{\underset{\approx}{A}} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{\underset{\approx}{P}} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & -0.04 & 1 \end{pmatrix}}_{\underset{\approx}{L}}$$

$$\cdot \underbrace{\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix}}_{\underset{\approx}{U}}$$

Once we have a PLU decomposition, we can easily get the solution!

We had $\underset{\approx}{A}\,\underset{\sim}{x} = \underset{\sim}{b}$

or: $\underset{\approx}{P}\,\underset{\approx}{L}\,\underset{\approx}{U}\,\underset{\sim}{x} = \underset{\sim}{b}$

Let $\underset{\sim}{z}$ be an array s.t. $\underset{\approx}{P}\,\underset{\sim}{z} = \underset{\sim}{b}$

And $\underset{\sim}{y}$ s.t. $\underset{\approx}{L}\,\underset{\sim}{y} = \underset{\sim}{z}$

Then we have the series of problems:

$$\underset{\approx}{P}\,\underset{\sim}{z} = \underset{\sim}{b} \qquad \text{(rearrange } \underset{\sim}{b} \text{ to get } \underset{\sim}{z}\text{)}$$

$$\underset{\approx}{L}\,\underset{\sim}{y} = \underset{\sim}{z} \qquad \text{(solve via forward substitution)}$$

$$\underset{\approx}{U}\,\underset{\sim}{x} = \underset{\sim}{y} \qquad \text{(solve via back substitution)}$$

You only have to factor $\underset{\approx}{A}$ once, then can solve for many vectors $\underset{\sim}{b}$!

Even w/ pivoting, still sensitive to errors! Look at simple example:

$$\begin{pmatrix} 0.780 & 0.563 \\ 0.457 & 0.330 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

we solve this using GE where we store all #'s to 3 decimal places (mult $\equiv \frac{0.457}{0.780} \approx 0.586$

$\therefore$

$$\begin{pmatrix} 0.780 & 0.563 \\ 0 & 0.0000820 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.217 \\ -0.000162 \end{pmatrix}$$

Solving via back substitution:

$x_2 = -1.98$, $x_1 = 1.71$

Checking the residual, we get:

$\underset{\sim}{r} = \underset{\sim}{b} - \underset{\approx}{A}\,\underset{\sim}{x} = \begin{pmatrix} -0.00206 \\ -0.00107 \end{pmatrix}$ which is about 10^{-2} to 10^{-3} times $\underset{\sim}{b}$, about right for 3 sig digit calc.

But the exact answer was really

$x_1 = 1$, $x_2 = -1$

We were off by 2x! Why? The problem was ill-conditioned.

$\underset{\approx}{A}$ was close to being singular

we always get:

size residual $\sim$ size sol'n $\times$ size $(A) \times \varepsilon_{mach}$

but

size error ~ size sol'n × cond(**A**) × ε_{mach}

What is cond(**A**)? It is a measure of how close **A** is to singularity

What happens if **A** is singular?

First, det(**A**) = 0

Second, for some b's there will be no sol'n for **x**, while for others there will be an inf. #

Ex: $$\begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

eq'ns are linearly dependent

any sol'n $x_2 = 1 - x_1$ will work!

and:

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

has no sol'n (requires 1 = 0)

To understand cond $(\underset{\sim}{A})$ we must look at <u>norms</u>

3 common types

<u>Euclidean Norm</u> (2-norm)

$$\|\underset{\sim}{x}\|_2 = \left[\sum_{i=1}^{N} (x_i^2)\right]^{1/2}$$

This is the usual physical vector length

<u>Infinity Norm</u> (∞-norm)

$$\|\underset{\sim}{x}\|_\infty = \max_i |x_i|$$

Gives a bound on the size of the elements of $\underset{\sim}{x}$

<u>Manhattan Norm</u> (1-norm)

$$\|\underset{\sim}{x}\| = \sum_{i=1}^{N} |x_i|$$

We will use this one!

All of these are specific examples of the general norm:

$$\|\underset{\sim}{x}\|_n = \left(\sum_{i=1}^{N} (x_i)^n \right)^{1/n}$$

Only $n = 1, 2, \infty$ are in common usage

What does this have to do with the condition ※?

We can form the scalar:

$$\frac{\|\underset{\approx}{A}\,\underset{\sim}{x}\|}{\|\underset{\sim}{x}\|} \leftarrow \text{1-norm!}$$

This depends on $\underset{\sim}{x}$!

Let $\displaystyle M = \max_{\underset{\sim}{x}} \frac{\|\underset{\approx}{A}\,\underset{\sim}{x}\|}{\|\underset{\sim}{x}\|}$

and

$$m = \min_{\underset{\sim}{x}} \frac{\|\underset{\approx}{A}\,\underset{\sim}{x}\|}{\|\underset{\sim}{x}\|}$$

If $\underset{\approx}{A}$ is <u>singular</u>, $m = 0$!

The condition number of $\underset{\approx}{A}$ is defined as:

$$\text{cond}(\underset{\approx}{A}) \equiv \frac{M}{m} \quad !$$

Ok, now what do we do with it??

Let $\underset{\approx}{A}\,\underset{\sim}{x} = \underset{\sim}{b}$

and $\underset{\approx}{A}\,(\underset{\sim}{x} + \Delta \underset{\sim}{x}) = (\underset{\sim}{b} + \Delta \underset{\sim}{b})$

we want to see how much $\underset{\sim}{x}$ changes with a small change in $\underset{\sim}{b}$, e.g. the rel. between $\Delta \underset{\sim}{x}$ and $\Delta \underset{\sim}{b}$.

We know that:

$$\underset{\approx}{A}\,\Delta \underset{\sim}{x} = \Delta \underset{\sim}{b} \qquad \text{(subtract off original eq'n)}$$

We know that

$$\|b\| \le M \|\underset{\sim}{x}\|$$

since $M \ge \dfrac{\|\underset{\sim}{b}\|}{\|\underset{\sim}{x}\|}$ by def$^{\underline{n}}$

Similarly,

$$\|\Delta \underset{\sim}{b}\| \geq m \, \|\Delta \underset{\sim}{x}\|$$

$\therefore$ for all $m \neq 0$ (non-singular $\underset{\approx}{A}$):

$$\frac{\|\Delta \underset{\sim}{x}\|}{\|\underset{\sim}{x}\|} \leq \operatorname{cond}(\underset{\approx}{A}) \frac{\|\Delta \underset{\sim}{b}\|}{\|\underset{\sim}{b}\|}$$

How do we calculate $\operatorname{cond}(\underset{\approx}{A})$?

First, we <u>define</u> $\|\underset{\approx}{A}\| \equiv M = \max\limits_{\underset{\sim}{x} \neq 0} \dfrac{\|\underset{\approx}{A}\underset{\sim}{x}\|}{\|\underset{\sim}{x}\|}$

It turns out that:

$$\|\underset{\approx}{A}\| = \max_j \|\underset{\sim}{a}_j\| \quad \text{where}$$

$\underset{\sim}{a}_j$ are the <u>columns</u> of $\underset{\approx}{A}$

The min m is:

$$m = \frac{1}{\|\underset{\approx}{A}^{-1}\|}$$

so: $\operatorname{cond}(\underset{\approx}{A}) = \dfrac{M}{m} = \|\underset{\approx}{A}\| \, \|\underset{\approx}{A}^{-1}\|$