Module 03: Linear Systems

Starting: Wednesday, January 22

Ax = b, where

- $A = \begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & \ddots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & & a_{mn} \end{bmatrix}$ b = $\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$

- And x = $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$

Solving a lower triangular system

```
% specific example
A = [1 0 0; 1 1 0; 1 1 1];
b = [1;3;6]a;
n = 3;

% general technique
x = zeros(n,1);
x(1) = b(1) / A(1,1);
for k=2:n
  x(k) = (b(k)-A(k,1:k-1)*x(1:k-1))/ A(k,k);
end
```

Formalize GE = LU factorization

- Step 1: Eliminate first column of A –

$$\begin{bmatrix} 1 & 0 & & 0 \\ -m_{21} & 1 & \ddots & 0 \\ \vdots & \vdots & & \vdots \\ -m_{n1} & 0 & & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & \ddots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix}$$

where $m_{j1} = a_{j1}/a_{11}$, which gives

$$\begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ 0 & a_{22}^{(1)} & \ddots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & & a_{nn}^{(1)} \end{bmatrix}$$

Formalize GE = LU factorization

- Step 2: Eliminate second column of $A^{(1)}$ –

$$\begin{bmatrix} 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & -m_{n2} & & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ 0 & a_{22}^{(1)} & & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & & a_{nn}^{(1)} \end{bmatrix}$$

where $m_{j2} = a_{j2}^{(1)} / a_{22}^{(1)}$, which gives

$$\begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ 0 & a_{22}^{(1)} & & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & a_{nn}^{(2)} \end{bmatrix}$$

Formalize GE = LU factorization

- Continue to step n-1

$$\begin{bmatrix} 1 & 0 & & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & -m_{nn-1} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ 0 & a_{22}^{(1)} & \ddots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & a_{nn}^{(n-2)} \end{bmatrix}$$

where $m_{nn-1} = a_{n,n-1}^{(n-2)}/a_{n-1,n-1}^{(n-2)}$, which gives

$$\begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ 0 & a_{22}^{(1)} & \ddots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & a_{nn}^{(n-1)} \end{bmatrix} = U$$

Summary of GE as LU factoring

$(M_{n-1}M_{n-2} \ldots M_2M_1)\, A = U$

$\rightarrow A = (M_{n-1}M_{n-2} \ldots M_2M_1)^{-1}\, U$

$\rightarrow A = (M_1^{-1} M_2^{-1} \ldots M_{n-2}^{-1}M_{n-1}^{-1})\, U = LU$, where L has a very special form:

$$L = \begin{bmatrix} 1 & 0 & & 0 \\ m_{21} & 1 & \ddots & 0 \\ \vdots & \vdots & & \vdots \\ m_{n1} & m_{n2} & & 1 \end{bmatrix}$$

Efficient Storage of LU factorization

After full elimination, arrange data so that $A^{(n-1)}$ holds the following information

$$\begin{bmatrix} a_{11} & a_{21} & & a_{n1} \\ m_{21} & a_{2n}^{(1)} & & a_{2n}^{(1)} \\ m_{31} & m_{32} & \ddots & a_{3n}^{(2)} \\ \vdots & \vdots & & \vdots \\ m_{n-1,1} & m_{n-1,1} & & a_{n-1,n}^{(n-1)} \\ m_{n1} & m_{n2} & & 1 \end{bmatrix}$$

Note: pivot values are stored along diagonal.

To retrieve the individual matrices in Matlab,

`U = triu(A)` and `L = tril(A,-1) + eye(n,n)`

Any drawbacks?

What to do if the pivot is 0?

Consider the system:

$$x_2 = 1$$
$$x_1 + x_2 = 2$$

Switch the order of the equations, and everything is fine – and no GE required. Solution is $x_1=x_2=1$.

What if the pivot is small, but nonzero?

Consider the "nearby" system

$$10^{-4}x_1 + x_2 = 1$$
$$x_1 + x_2 = 2$$

Pivot is nonzero → Proceed as usual to get ...

$$10^{-4}x_1 + x_2 = 1$$
$$-9999x_2 = -9998$$

Solution is:

x_2 = 9998/9999 = 0.99989998…

x_1 = 10000/9999 = 1.00010001…

(Small change to system → Small change to solution. Problem appears to be well-conditioned)

What happens in FL(10,3,1) with rounding?

$$10^{-4}x_1 + x_2 = 1$$
$$fl(-9999)x_2 = fl(-9998)$$

which becomes

$$10^{-4}x_1 + x_2 = 1$$
$$10^{-4}x_2 = 10^{-4}$$

with the solution x_2=1, x_1 = 0

x_2 is close to the actual solution, but x_1 has relative error 100%

Small error in representation → Algorithm fails → Computationally GE is unstable as described

What if we pivoted anyway?

$$10^{-4}x_1 + x_2 = 1$$
$$x_1 + x_2 = 2$$

Switch the rows …

$$x_1 + x_2 = 2$$
$$10^{-4}x_1 + x_2 = 1$$

One step of GE:

$$x_1 + x_2 = 2$$
$$fl(1 - 10^{-4})x_2 = fl(1 - 2 * 10^{-4})$$

or

$$x_2 = 1$$

which gives $x_1=x_2=1$, which is very close to the true solution of the perturbed system.

Why the difference in performance?

- Without switching, pivot was very small relative to other values → multiplier was very large → loss of significant digits
- After switching, pivot was much larger → multiplier was relatively small → may still lose some digits, but not as significant
- Use pivoting even if pivot is nonzero → Gaussian Elimination with Partial Pivoting (GEPP)

How do permutations affect LU?

- $A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$
- Choose 8 as the new pivot. Modify the matrix by pre-multiplying by a permutation matrix
- $\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}$
- $P_1 * A = A^{(1)}$
- Eliminate below the first diagonal (pre-multiply by M_1)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{4} & 0 & 1 & 0 \\ -\frac{3}{4} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ 0 & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix}$$

- Choose a new pivot: 7/4 is largest entry in column 2 (diagonal and below).
- Pre-multiply by permutation matrix P_2 to swap rows 2 and 4.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ 0 & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ 0 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}$$

- Eliminate below the diagonal of column 2 by premultiplying by M_2

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{3}{7} & 1 & 0 \\ 0 & \frac{2}{7} & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ 0 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{2}{7} & \frac{4}{7} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix}$$

- Choose a new pivot: -6/7 is largest entry in column 3 (diagonal and below).
- Pre-multiply by permutation matrix P_3 to swap rows 3 and 4.

$$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}8 & 7 & 9 & 5\\0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4}\\0 & 0 & -\frac{2}{7} & \frac{4}{7}\\0 & 0 & -\frac{6}{7} & -\frac{2}{7}\end{bmatrix}=\begin{bmatrix}8 & 7 & 9 & 5\\0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4}\\0 & 0 & -\frac{6}{7} & -\frac{2}{7}\\0 & 0 & -\frac{2}{7} & \frac{4}{7}\end{bmatrix}$$

- Eliminate below the third diagonal by pre-multiplying by M_3.

******* C O R R E C T I O N *******

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \\ 0 & 0 & -\frac{2}{7} & \frac{4}{7} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \\ 0 & 0 & 0 & \frac{2}{3} \end{bmatrix}$$

So, we now have

$$M_3P_3M_2P_2M_1P_1 * A = U$$

~~Note: $M_3P_3M_2P_2M_1P_1 = M_3M_2M_1P_3P_2P_1$ because of the special nature of the permutation matrices~~

~~So, $P_3P_2P_1\,A = (M_3M_2M_1)^{-1}\,U = LU$, or $PA = LU$~~

***** C O R R E C T I O N *****

In general, GE with Partial Pivoting gives:

$$(M_n P_n \dots M_2 P_2 M_1 P_1) * A = U$$

So, $A = MU$ where $M = (M_n P_n \dots M_2 P_2 M_1 P_1)^{-1}$.

But, M is not unit, lower triangular.

Let $P = P_n P_{n-1} \dots P_2 P_1$, then $PA = PMU = LU$, where $L = PM$ is unit lower triangular matrix.

So, $PA = LU$ is our new factorization.

Solve Ax=b when PA=LU

Ax = b

→PAx = Pb

→LUx = Pb

→Solve for y: Ly=Pb

→Then, solve for x: Ux = y

Running time of Gaussian Elimination

- For each stage of process, count the number of floating point operations (additions/subtractions/multiplications/divisions)
- Step 1: Calculate LU decomposition: A = LU
- Step 2: Solve for y: Ly = b, forward substitution
- Step 3: Solve for x: Ux = y, backward substitution

How does the introduction of partial pivoting affect the running time?

Important summations

$$\sum_{k=1}^{n} 1 = n$$

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

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

Pseudocode for Gaussian Elimination

```
L = diag(n)
U = A
for p = 1:n-1
   for r = p+1:n
      m = -U(r,p)/U(p,p)
      U(r,p) = 0
      for c = p+1:n
         U(r,c) = U(r,c) + m*U(p,c)
      end
      L(r,p) = -m
   end
end
```

Solving the Triangular Systems

Forward Substitution

```
y = b
for r = 2:n
 for c = 1:r-1
  y(r) = y(r) -
         L(r,c)*y(c)
  end
end
```

Backward Substitution

```
x = y
for r = n:-1:1
 for c = r+1:n
  x(r) = x(r) -
         U(r,c)*x(c)
 end
 x(r) = x(r) /
         U(r,r)
end
```

Consider another small system in FL(10,3,1) with rounding

$$2x_1 + 20000x_2 = 20000$$
$$x_1 + x_2 = 2$$

Exact soln is: x_1=10000/9999, x_2 = 9998/9999

No row exchanges needed →

$$2x_1 + 20000x_2 = 20000$$
$$fl(1-10^4)x_2 = fl(1-2*10^4)$$
$$\text{or } -10^4x_2 = -10^4$$
$$\rightarrow x1 = 0,\ x2 = 1$$

→Partial pivoting did not help here. Algorithm unstable.

A different approach

Change the order of the columns!

$$20000x_2 + 2x_1 = 20000$$

$$x_2 + x_1 = 2$$

Now, eliminate x_2 from second equation →

$$fl(1-2/20000)\ x_1 = fl(2-20000/20000) \rightarrow$$

$$x_1 = 1 \rightarrow x2 = 19998/20000$$

We chose the largest entry in the coefficient matrix to be our pivot → GE with complete pivoting

More on complete pivoting

- At step k, partial pivoting looks for largest value in column k, for row k, k+1, …n
- At step k, complete pivoting looks for the largest value in the submatrix containing rows k, k+1, … n and columns k,k+1, …, n.
- To bring new pivot into row k, col k, we need to permute rows and columns: equivalent to matrix multiplication $P_k A(k-1) Q_k$ for row (P_k) and column (Q_k) permutation matrices.

"Complete" Pivoting

- Leads to: M P A Q = U, or P A Q = LU, defining L and U as before, and P, Q are permutation matrices.
- How to solve Ax = b using this decomposition?
- Complete pivoting is rarely worth the extra work involved. Partial pivoting is usually sufficient.

We have so far assumed

- Unique soln to Ax = b, and
- Well-conditioned system.

It can be shown (deSterck) that the conditioning of solving Ax=b depends primarily on properties of A.

For example, consider (A) (x+ Δx) = b+Δb, what can we say about $\dfrac{\|\Delta x\| / \|x\|}{\|\Delta b\| / \|b\|}$

Define: Condition number of a nonsingular matrix A relative to a natural p-norm $\|\cdot\|_p$ is:

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

- If κ(A) is small (e.g. < 10), then GE with partial pivoting should find a reasonable solution
- If κ(A) is large (e.g. > 100), then even GE with complete pivoting will have issues.

Recall: Matrix Norms

Matrix norms induced from vector norms include:

- The 1-norm (maximal absolute column sum):

$$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$

- The 2-norm:

$$\|A\|_2 = \sqrt{\lambda_{max}(A^T A)}$$

- The ∞-norm (maximal absolute row sum):

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix} \rightarrow A^{-1} = \begin{bmatrix} -{}^{5}/_{12} & {}^{1}/_{3} & {}^{1}/_{4} \\ {}^{7}/_{12} & -{}^{2}/_{3} & {}^{1}/_{4} \\ {}^{1}/_{12} & {}^{1}/_{3} & -{}^{1}/_{4} \end{bmatrix}$$

- $\|A\|_1 = 7,\ \|A^{-1}\|_1 = {}^{4}/_{3}$, κ_1(A) = 9 ${}^{1}/_{3}$

- $\|A\|_\infty = 6, \|A^{-1}\|_\infty = 1\ {}^{1}/_{2}$, κ_{∞}(A) = 9

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix} \rightarrow A^{-1} = \begin{bmatrix} -{}^{5}/_{12} & {}^{1}/_{3} & {}^{1}/_{4} \\ {}^{7}/_{12} & -{}^{2}/_{3} & {}^{1}/_{4} \\ {}^{1}/_{12} & {}^{1}/_{3} & -{}^{1}/_{4} \end{bmatrix}$$

$$\|A\|_2 = 6.06 \text{ since } A^T A = \begin{bmatrix} 14 & 10 & 12 \\ 10 & 9 & 11 \\ 12 & 11 & 19 \end{bmatrix}$$

$$\Rightarrow \lambda(A^T A) = \begin{bmatrix} 0.89 \\ 4.39 \\ 36.71 \end{bmatrix}$$

$$\|A^{-1}\|_2 = 1/\sqrt{\lambda_{min}(A^T A)} = \sqrt{{}^{1}/_{0.89}} = 1.06$$

$\Rightarrow \kappa_2(A) = 6.41$

$\Rightarrow$ Ax = b is a well-conditioned problem

- But consider:

- $H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$, $\kappa_2(H) = 15513.74$

→ Hx=b is generally an ill-conditioned problem

What solution is better?

- Let x be the true solution, $\hat{x}$ be the computed solution
- Residual error = b - A $\hat{x}$
- Relative error = ${}^{\|x-\hat{x}\|}/_{\|x\|}$
- Consider $\begin{bmatrix} .780 & .563 \\ .913 & .659 \end{bmatrix} x = \begin{bmatrix} .217 \\ .254 \end{bmatrix}$

  $\hat{x}_1 = \begin{bmatrix} .341 \\ -0.087 \end{bmatrix}$ and $\hat{x}_2 = \begin{bmatrix} .999 \\ -1.00 \end{bmatrix}$ and $x = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$
- Residual errors:

  $\|b - Ax_1\| = 1e - 6,\qquad \|b - Ax_2\| = 1.2e - 3$
- Relative errors:

  ${}^{\|x_1-\hat{x}\|}/_{\|x_1\|} = 0.8,\qquad {}^{\|x_2-\hat{x}\|}/_{\|x_2\|} = 7.1e - 4$

Iterative Techniques

- Idea: start with an initial guess $x^{(0)}$
- Use $x^{(k)}$ to generate a new guess $x^{(k+1)}$
- Repeat until a "good" solution found

- Do we need iterative methods?
  - For small systems – probably not
  - Perform well for larger, sparse systems

General Iterative Approach

For i=1:n, $\sum_{j=1}^{n} a_{ij}x_j = b_i$, or

$$\sum_{j=1}^{i-1} a_{ij}x_j + a_{ii}x_i + \sum_{j=i+1}^{n} a_{ij}x_j = b_i$$

Rewrite to isolate x_i

$$x_i = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_j)/a_{ii}$$

Jacobi Method for Ax=b

$$x_i^{(k+1)}$$
$$= (b_i - \sum_{j=1}^{i=1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}) / a_{ii}$$

Note: assumes $a_{ii} \neq 0$.

Gauss-Seidel Method for Ax=b

Note, when setting x_i in Jacobi, x_k ($k<i$) have already been updated. Use them in new estimate of x_i:

$$x_i^{(k+1)}$$
$$= (b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}) / a_{ii}$$

Note: assumes $a_{ii} \neq 0$.

When to stop iterating?

- Set number of iterations performed
- Residual satisfies $\left\| b - Ax^{(k)} \right\| \leq tol$
- Consecutive guesses satisfy $\left\| x^{(k+1)} - x^{(k)} \right\| \leq tol$

Will the iterates converge?

Defn: A square matrix A is strictly diagonally dominant if for all i=1:n,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} \left\lceil a_{ij} \right\rceil$$

Thm: If A is strictly diagonally dominant, then the sequence $x^{(k)}$ generated from $x^{(0)}$ using either Jacobi or Gauss-Seidel will converge to the unique solution of Ax-b.

What if the system is overdetermined? (More equations than unknowns)

- $A = \begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & & a_{mn} \end{bmatrix}$ b = $\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$
- And x = $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ where m > n.

How many solutions are there?

- No solutions

$$x_1 + x_2 = 2, \quad x_1 - x_2 = 0, \quad x_1 + 2x_2 = 5$$

- One solution

$$x_1 + x_2 = 2, \quad x_1 - x_2 = 0, \quad x_1 + 2x_2 = 3$$

- Infinite number of solutions

$$x_1 + x_2 = 2, \quad 2x_1 + 2x_2 = 4, \quad 3x_1 + 3x_2 = 6$$

This holds for any $m > n$.

When there is not a unique solution …

Consider choosing *x* to minimize the residual errors in the system *Ax = b*, i.e.

$$\min_{x} \|b - Ax\|_2$$

(Linear Least Squares Problem)

Solution to our problem satisfies …

$$A^TAx = A^Tb$$

(called the Normal Equations)

If A has full column rank, A^TA is positive definite, and the solution of the Normal Equations is a minimum, i.e. our least squares solution.

Example

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 0 \\ 2 & -2 & -1 \\ -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 5 \\ 0 \\ 3 \end{bmatrix}$$

Normal equations reduce to solving

$$\begin{bmatrix} 19 & 4 & 1 \\ 4 & 11 & 7 \\ 1 & 7 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 25 \\ 22 \\ 19 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10747 \\ 0.8448 \\ 1.2011 \end{bmatrix}$$

Solving the Least Squares Problem

First Approach:

Use GE to solve the Normal Equations

- Calculate $M=A^TA$
- Use GE, to find P, L, U such that PM = LU
- Solve for y: $Ly = PA^Tb$
- Solve for x: $Ux = y$

Solving the Least Squares Problem

A second approach:

- A^TA is symmetric
- If positive definite –> Cholesky decomposition
- Find L such that $A^TA = LL^T$
- Solve for y: $Ly = A^Tb$
- Solve for x: $L^Tx = y$

Cholesky Decomposition of Symmetrix Positive Definite matrix M

$$\begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} & \dots & l_{n1} \\ 0 & l_{22} & l_{32} & \dots & l_{n2} \\ 0 & 0 & l_{33} & \cdots & l_{n3} \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} m_{11} & m_{21} & m_{31} & \dots & m_{n1} \\ m_{21} & m_{22} & m_{32} & \dots & m_{n2} \\ m_{31} & m_{32} & m_{33} & \cdots & m_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & m_{n3} & \cdots & m_{nn} \end{bmatrix}$$

Cholesky

For k=1:n

$$l_{kk} = \sqrt{m_{kk} - \sum_{j=1}^{k-1} {l_{kj}}^2}$$

for i=k+1:n

$$l_{ik} = \frac{1}{l_{kk}}\left(m_{ik} - \sum_{j=1}^{k-1} l_{ij} l_{kj}\right)$$

Solving the Least Squares Problem

A third approach:

- Factor A = QR, where
  - Q is mxn orthogonal ($Q^TQ = I$)
  - R is nxn upper triangular with $r_{ii}>0$
- Use factorization in Normal Equations