Thursday, 9/12/13 Math 471, Fall 2013, Section 001 Lecture 3

1 Linear algebra review

Text: section 3.0

2 fundamental problems of linear algebra:

1. Solving the linear system $A\boldsymbol{x} = \boldsymbol{b}$
2. Solving the eigenvalue problem $A\boldsymbol{x} = \lambda\boldsymbol{x}$

1.1 Systems of linear equations

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \quad & \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}\right\} : \begin{array}{l}\text{system of linear equations;} \\ n \text{ equations in } n \text{ unknowns}\end{array}$$

This system can be written in 3 equivalent forms:

1. $$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$
2. $A\boldsymbol{x} = \boldsymbol{b}$
3. $\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i$: row index, j : column index

The basic problem is to find $\boldsymbol{x}$ from a given A and $\boldsymbol{b}$.

- We can't "divide" by a matrix : $A\boldsymbol{x}/A = \boldsymbol{b}/A$ doesn't make sense
- But $\boldsymbol{x} = A\backslash\boldsymbol{b}$ works in Matlab... What's it doing?

Theorem 1 (Fundamental theorem of linear algebra)**.** *The following conditions are equivalent:*

1. *The equation $A\boldsymbol{x} = \boldsymbol{b}$ has a unique solution for any vector $\boldsymbol{b}$*
2. *The matrix A is invertible, i.e., there exists a matrix A^{-1} such that $AA^{-1} = A^{-1}A = I$*
3. $\det A \neq 0$
4. *The equation $A\boldsymbol{x} = 0$ has the unique solution $\boldsymbol{x} = 0$*
5. *The columns of A are linearly independent*

6. *The eigenvalues of A are nonzero*

Notes:

- If A is invertible, then $\boldsymbol{x} = A^{-1}\boldsymbol{b}$ (because $A\boldsymbol{x} = \boldsymbol{b}$ can be solved by left-multiplication with A^{-1}), but this is not the best way to compute $\boldsymbol{x}$ in practice
- There are two methods for solving $A\boldsymbol{x} = \boldsymbol{b}$, direct methods and iterative methods. We begin with direct methods.

2 Gaussian elimination

Tuesday, 9/17/13

Text: section 3.1

First, consider the special case where A is upper triangular.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + \cdots + a_{1,n-1}x_{n-1}x_{n-1} + a_{1n}x_n &= b_1 \\ a_{22}x_2 + a_{23}x_3 + a_{24}x_4 \cdots + a_{2,n-1}x_2 + a_{2n}x_n &= b_2 \\ \ddots & \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\ a_{nn}x_n &= b_n \end{aligned}$$

Idea: Solve for x_n using the nth equation, plug result into $(n-1)$th equation to solve for $x_{n-1}, \ldots$.

$$\begin{aligned} \implies x_n &= \frac{b_n}{a_{nn}} \\ x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}} &= \frac{b_{n-1} - a_{n-1,n}}{a_{n-1,n-1}} \cdot \frac{b_n}{a_{nn}} \\ \vdots & \\ x_1 &= \frac{b_1 - a_{12}x_2 + \cdots + a_{1n}x_n}{a_{11}} \end{aligned}$$

This idea is known as back substitution.

Algorithm 1: Back substitution

Input : Upper-triangular matrix A, matrix size n, right-hand-side vector $\boldsymbol{b}$
Output: $\boldsymbol{x}$ such that $A\boldsymbol{x} = \boldsymbol{b}$

```
1  x_n = b_n / a_nn
2  for i = n − 1 : −1 : 1 do
3  |  % i = row index
4  |  sum = b_i
5  |  for j = i + 1 : n do
6  |  |  % j = column index
7  |  |  sum = sum − a_ij · x_j
8  |  end
9  |  x_i = sum / a_ii
10 end
11 return x
```

Operation Count: Count of the number of multiplications (divisions) and additions (subtractions).
At row n, there is 1 division (line 1).
At row $n-1$ there is 1 multiplication (line 7), one addition (line 7), and one division (line 9).
At row $n-2$ there are 2 multiplications and additions (line 7), and one division (line 9).
At row $n-3$ there are 3 multiplications and additions (line 7), and one division (line 9).
$\Rightarrow$ At row $n-i$ there are i multiplications and additions (line 7) and one division (line 9), and so on...

$$\text{operation count } = \underbrace{\sum_{i=1}^{n-1} i}_{\text{line 7}} + \underbrace{\sum_{i=1}^{n} 1}_{\text{line 9}} = \frac{1}{2}n(n-1) + n \sim O(n^2) \text{ as } n \to \infty$$

Theorem 2 (Sum of consecutive integers)**.** *The sum of the integers from 1 to n is given by*

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

Proof. Draw picture. Can also prove by induction. □

Note : The same considerations hold if A is lower triangular.

Question 1: How do we handle non-triangular matrices?

Idea : Use elementary row operations to reduce A to upper triangular form, then apply back substitution to find $\boldsymbol{x}$.

△

Example 1: $n = 3$.

The system

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

has an augmented matrix of

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right).$$

Step 1: Eliminate variable x_1 from equations 2 and 3.

$$m_{21} = \frac{a_{21}}{a_{11}}, \implies a_{21} \to 0 \qquad \% \; m_{21} \text{ is called a multiplier.}$$
$$a_{22} \to a_{22} - m_{21}a_{12}$$
$$a_{23} \to a_{23} - m_{21}a_{13}$$
$$b_2 \to b_2 - m_{21}b_1$$

$$m_{31} = \frac{a_{31}}{a_{11}}, \implies a_{31} \to 0 \qquad \% \; m_{31} \text{ is the row 3 column 1 multiplier.}$$
$$a_{22} \to a_{32} - m_{31}a_{12}$$
$$a_{23} \to a_{33} - m_{31}a_{13}$$
$$b_2 \to b_3 - m_{31}b_1$$

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & a_{32} & a_{33} & b_3 \end{array}\right)$$ these elements have changed

Step 2: Eliminate variable x_2 from equation 3.

$$m_{32} = \frac{a_{32}}{a_{22}}, \implies a_{32} \to 0$$
$$a_{33} \to a_{33} - m_{32}a_{232}$$
$$b_3 \to b_3 - m_{32}b_2$$

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & 0 & a_{33} & b_3 \end{array}\right) \quad : \text{ upper triangular}$$

We can now use back substitution.

△

Example 2:

$$\begin{aligned} 2x_1 - x_2 &= 1 \\ -x_1 + 2x_2 - x_3 &= 0 \\ -x_2 + 2x_3 &= 1 \end{aligned}$$

$$\left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & 1 \end{array}\right) \quad \begin{array}{l} m_{21} = -1/2 \\ m_{31} = 0 \end{array}$$

$$\left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ 0 & 3/2 & -1 & 1/2 \\ 0 & -1 & 2 & 1 \end{array}\right) \quad m_{32} = -1/(3/2) = -2/3$$

$$\left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ 0 & 3/2 & -1 & 1/2 \\ 0 & 0 & 4/3 & 4/3 \end{array}\right)$$

$x_3 = 1\ ,\ x_2 = (\frac{1}{2} - (-1)\cdot 1)/\frac{3}{2} = 1\ ,\ x_1 = (1 - (-1)\cdot 1)/2 = 1$ check : ok

△

Algorithm 2: Reduction to upper triangular form with nonzero pivots

Input : Square matrix A, matrix size n
Output: Square matrix A' so that A' is upper-triangular and row-equivalent to A

```
1  for k = 1 : n − 1 do
2  |  % k = step index
3  |  for i = k + 1 : n do
4  |  |  m_ik = a_ik/a_kk    % assume a_kk ≠ 0 (more later)
5  |  |  for j = k + 1 : n do
6  |  |  |  a_ij = a_ij − m_ik · a_kj
7  |  |  end
8  |  |  b_i = b_i − m_ik · b_k
9  |  end
10 end
11 return A   % Note:  this algorithm overwrites A with A'
```

Note: The element a_{kk} in step k is called a pivot. These are the diagonal elements in the last step. In the previous example, the pivots are 2, $\frac{3}{2}$, and $\frac{4}{3}$.

Operation Count: The leading order term comes from line 6. Each time it executes, there is one multiplication and one addition.

$$\left.\begin{array}{r} k = 1 \Rightarrow 2(n-1)^2 \text{ operations} \\ k = 2 \Rightarrow 2(n-2)^2 \text{ operations} \\ \vdots \\ k = n-2 \Rightarrow 2 \cdot 2^2 \text{ operations} \\ k = n-1 \Rightarrow 2 \cdot 1^2 \text{ operations} \end{array}\right\} \Rightarrow 2 \cdot \sum_{k=1}^{n-1} k^2 = 2 \cdot \frac{1}{6} n(n-1)(2n-1) \text{ operations}$$

Hence the operation count for Gaussian elimination on an $n \times n$ matrix is $\sim \frac{2}{3}n^3$ as $n \to \infty$.

Theorem 3 (Sum of consecutive squares)**.** *The sum of squared integers k from $k = 1$ to $k = n$ is given by*

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

Proof. The proof begins by adding a sequence of clever zeros to n^3. Note that the tautology $n^3 = n^3$ may be rewritten as $0 = n^3 - n^3$. Thus,

$$\begin{aligned} n^3 &= n^3 + (n-1)^3 - (n-1)^3 + (n-2)^3 - (n-2)^2 + \cdots + 2^3 - 2^3 + 1^3 - 1^3 \\ &= \sum_{k=1}^{n} k^3 - \sum_{k=1}^{n-1} k^3 = \sum_{k=1}^{n} \left(k^3 - (k-1)^3\right) \\ &= \sum_{k=1}^{n} \left(k^3 - (k^3 - 3k^2 + 3k - 1)\right) = \sum_{k=1}^{n} \left(3k^2 - 3k + 1\right) \\ &= 3\sum_{k=1}^{n} k^2 - 3\sum_{k=1}^{n} k + \sum_{k=1}^{n} 1 \end{aligned}$$

$$n^3 = 3\sum_{k=1}^{n} k^2 - \frac{3}{2}n(n+1) + n$$

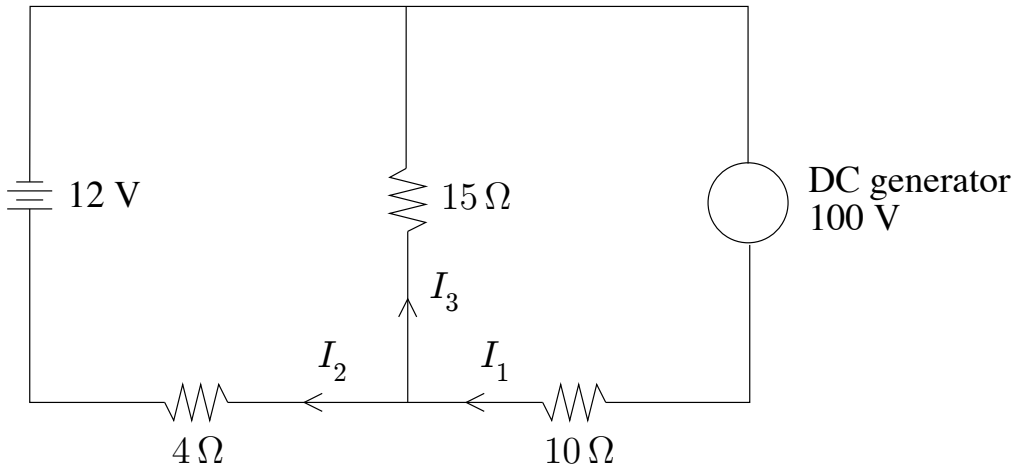
$$\begin{aligned} \implies \sum_{k=1}^{n} k^2 &= \frac{1}{3}\left(n^3 + \frac{3}{2}n(n+1) - n\right) = \frac{1}{6}n\left(2n^2 + 3n + 1\right) \\ &= \frac{1}{6}n(n+1)(2n+1) \end{aligned}$$

□

Thursday, 9/19/13

Explain 5 pt grading system

Example 3: Charging a car battery (p159, problem 13).



Use Kirchoff's law and Ohm's law:

- Net voltage around each loop is zero, and $V = IR$

$$4I_2 + 12 - 15I_3 = 0$$
$$15I_3 - 100 + 10I_1 = 0$$

- The current into a junction is equal to the current flowing out

$$I_1 = I_2 + I_3$$

We have 3 linear equations in 3 unknowns, I_1, I_2, I_3

$$\left.\begin{array}{c} 0I_1 + 4I_2 - 15I_3 = -12 \\ 10I_1 + 0I_2 + 15I_3 = 100 \\ I_1 - I_2 - I_3 = 0 \end{array}\right\} \Rightarrow \begin{bmatrix} 0 & 4 & -15 \\ 10 & 0 & 15 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} -12 \\ 100 \\ 0 \end{bmatrix}$$

and we can't apply Gaussian elimination to the matrix because the first pivot is zero.

△

2.1 Pivoting strategies

Text : section 3.2

The Reduction to Triangular Form algorithm makes use of 2 elementary row operations: scalar multiplication and addition. To implement a pivoting strategy, we employ the third elementary row operation: row exchange.

Partial pivoting
Consider the reduced matrix at the beginning of step k:

$$\left(\begin{array}{ccccccc} a_{11} & \cdots & \cdots & a_{1k} & \cdots & a_{1n} & b_1 \\ & \ddots & & \vdots & & \vdots & \vdots \\ & & \ddots & \vdots & & \vdots & \vdots \\ & & & a_{kk} & \cdots & a_{kn} & b_k \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{nk} & \cdots & a_{nn} & b_n \end{array}\right)$$

If $a_{kk} = 0$, we find index l such that $|a_{lk}| = \max\{|a_{ik}| : k \le i \le n\}$, then we interchange row l and row k and proceed with elimination.

Notes:

- If A is invertible, then partial pivoting does not break down (pf: Math 571).
  - It is possible to construct an invertible matrix A such that Gaussian elimination with partial pivoting loses accuracy – but in 60 years of computing, these types of matrices have not been found in applications; they are only found when you want to break Gaussian elimination. If you chose a billion matrices at random, you would almost certainly not even have found one for which Gaussian elimination is unstable.

- Moving vectors around in computer memory can be very expensive – much more so than simply performing floating point arithmetic operations; it is better to do it implicitly. In practice we use another array, known as an index array, to avoid the expense of actually moving matrix rows.

  Let $\boldsymbol{r}$ be the index array; it is a vector of length n, where n corresponds to the size of the $n \times n$ matrix A. The array is initialized so that $r_i = i$, for $i = 1, \ldots, n$. When partial pivoting triggers a row exchange between rows l and k, we simply interchange the lth and kth elements of $\boldsymbol{r}$ to keep track. Hence, the ith element of $\boldsymbol{r}$ denotes the row used for the pivot at step i, and the actual matrix A remains with its original row-ordering.

- Partial pivoting gets its name because it only performs exchanges by rows; another strategy might exchange both rows and columns to find the largest element. This is known as complete pivoting, and it rarely used because it's too expensive.

  Consider the reduced matrix from step k, above. A complete pivoting algorithm would search the remaining $n - k$ rows **and** the remaining $n - k + 1$ columns to find the largest element, and use that as the pivot. This results in having to check $O((n-k)^2)$ as $n \to \infty$ elements per step, which, over the n steps required to find A^{-1}, is another $O(n^3)$ expense in an already expensive procedure.

  By contrast, partial pivoting only has to check $n - k$ rows per step, which is much less work. In practice, partial pivoting provides nearly the same amount of accuracy, so it is the method we use.

- **Beware!** Recall that, because of roundoff error, a number that should be equal to zero might have a nonzero actual value that is on the order of machine epsilon. Hence, in practice, pivoting is not only used when a pivot element is equal to zero, but also if a pivot element is **near** zero.

Example 4: Consider the 2×2 matrix $A = \begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix}$ and right-hand side vector $\boldsymbol{b} = \begin{pmatrix} 1+\epsilon \\ 2 \end{pmatrix}$ where $0 < \epsilon \ll 1$.

1. The exact solution is given by the usual process of elimination.

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1+\epsilon \\ 1 & 1 & 2 \end{array}\right) \to \left(\begin{array}{cc|c} \epsilon & 1 & 1+\epsilon \\ 0 & 1-\frac{1}{\epsilon} & 1-\frac{1}{\epsilon} \end{array}\right) \Rightarrow \left.\begin{array}{l} x_1 = \frac{1+\epsilon-1}{\epsilon} = 1 \\ x_2 = \frac{1-1/\epsilon}{1-1/\epsilon} = 1 \end{array}\right\} : \text{exact solution}$$

$$m_{21} = \tfrac{1}{\epsilon}$$

2. In the presence of roundoff error, $1 - \dfrac{1}{\epsilon} \approx -\dfrac{1}{\epsilon}$ and $1 + \epsilon \approx 1$, and we get

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 0 & -\frac{1}{\epsilon} & -\frac{1}{\epsilon} \end{array}\right) \Rightarrow \left.\begin{array}{l} \tilde{x}_1 = \frac{1-1}{\epsilon} = 0 \\ \tilde{x}_2 = \frac{-1/\epsilon}{-1/\epsilon} = 1 \end{array}\right\} : \text{computed solution, inaccurate}$$

3. Now apply partial pivoting in the presence of roundoff error.

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ \epsilon & 1 & 1 \end{array}\right) \to \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right) \Rightarrow \left.\begin{array}{l} \tilde{x}_1 = 1 \\ \tilde{x}_2 = 1 \end{array}\right\} : \text{computed solution, accurate}$$

$$m_{21} = \tfrac{\epsilon}{1} = \epsilon$$

This is an issue of stability, which we will return to later.

△

3 Vector and matrix norms

Text: section 3.3

Definition 4. A vector norm is a function $\boldsymbol{x} \mapsto \|\boldsymbol{x}\|$ that satisfies the following properties:

1. $\|\boldsymbol{x}\| \geq 0$ and $\|\boldsymbol{x}\| = 0 \Leftrightarrow \boldsymbol{x} = 0$
2. $\|\alpha\boldsymbol{x}\| = |\alpha| \cdot \|\boldsymbol{x}\|$, where α is a scalar
3. $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$: triangle inequality

Vector norms are used to measure the "size" of a vector in various contexts.

Example 5: The following two examples define norms for vectors $\boldsymbol{x} \in \mathbb{R}^n$.

1. $\|\boldsymbol{x}\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$: Euclidean length
2. $\|\boldsymbol{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$

Proof. (hw)

Consider $\boldsymbol{x} = [1, 2]^T$. Then $\|\boldsymbol{x}\|_2 = \sqrt{5}$ and $\|\boldsymbol{x}\|_\infty = 2$.

△

In linear algebra, where a matrix A can represent the action of a specific linear transformation T with respect to a particular set of basis vectors, we often regard vectors $\boldsymbol{x}$ as *input* and vectors $A\boldsymbol{x}$ as *output*.
With this interpretation, the quantity $\dfrac{\|A\boldsymbol{x}\|}{\|\boldsymbol{x}\|}$ is the amplification factor for a given input vector $\boldsymbol{x}$.

Definition 5. We define a matrix norm to be the maximum amplification factor over all nonzero input vectors,

$$\|A\| = \max_{\boldsymbol{x} \neq 0} \frac{\|A\boldsymbol{x}\|}{\|\boldsymbol{x}\|}.$$

The matrix norm satisfies the following properties:

1. $\|A\| \geq 0$ and $\|A\| = 0 \Leftrightarrow A = 0$
2. $\|\alpha A\| = |\alpha| \cdot \|A\|$
3. $\|A + B\| \leq \|A\| + \|B\|$
4. $\|A\boldsymbol{x}\| \leq \|A\| \cdot \|\boldsymbol{x}\|$
5. $\|AB\| \leq \|A\| \cdot \|B\|$

Property 5.

$$\|AB\| = \underbrace{\max_{\boldsymbol{x}\neq 0} \frac{\|AB\boldsymbol{x}\|}{\|\boldsymbol{x}\|}}_{\text{def.}} \leq \underbrace{\max_{\boldsymbol{x}\neq 0} \frac{\|A\| \cdot \|B\boldsymbol{x}\|}{\|\boldsymbol{x}\|}}_{\text{prop. 4}} \leq \underbrace{\max_{\boldsymbol{x}\neq 0} \frac{\|A\| \cdot \|B\| \cdot \|\boldsymbol{x}\|}{\|\boldsymbol{x}\|}}_{\text{prop. 4}} = \|A\| \cdot \|B\|$$

□

Notes:

- Matrix norms that are derived from associated vector norms in this way are known as *natural* or *associated* norms
- Computing $\|A\|$ using the definition is difficult, but there are more convenient formulas that may be used...

Theorem 6. *The induced matrix norm $\|A\|_\infty$, associated with the l_∞ vector norm, $\|\cdot\|_\infty$, is equivalent to the maximum row sum,*

$$\|A\|_\infty = \max_{\boldsymbol{x}\neq 0} \frac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \max_i \sum_j |a_{ij}| .$$

Proof.

$$\|A\boldsymbol{x}\|_\infty = \max_i |(A\boldsymbol{x})_i| = \max_i \left| \sum_j a_{ij} x_j \right| \leq \max_i \sum_j |a_{ij}|\,|x_j| \leq \max_i \sum_j |a_{ij}| \cdot \|\boldsymbol{x}\|_\infty$$

$$\Rightarrow \frac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} \leq \frac{\max_i \sum_j |a_{ij}| \cdot \|\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \max_i \sum_j |a_{ij}| = \sum_j |a_{lj}| \text{ for some index } l \in [1, n]$$

Define $\boldsymbol{y} = [\text{sign}(a_{l1}), \text{sign}(a_{l2}), \ldots, \text{sign}(a_{ln})]^T$. Then $\|\boldsymbol{y}\|_\infty = 1$.
By construction of $\boldsymbol{y}$,

$$\sum_j |a_{lj}| = \sum_j a_{lj} y_j = (A\boldsymbol{y})_l \leq \|A\boldsymbol{y}\|_\infty .$$

$$\Rightarrow \|A\|_\infty = \max_{\boldsymbol{x}\neq 0} \frac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} \leq \sum_j |a_{lj}| \leq \frac{\|A\boldsymbol{y}\|_\infty}{\|\boldsymbol{y}\|_\infty} \leq \max_{\boldsymbol{x}\neq 0} \frac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \|A\|_\infty$$

□

Theorem 7. *The induced matrix norm $\|A\|_2$ associated with the l_2 or Euclidean vector norm $\|\cdot\|_2$ is the square root of the maximum eigenvalue of the matrix $A^T A$,*

$$\|A\|_2 = \max_{\boldsymbol{x}\neq 0} \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \max\{\sqrt{\lambda}\},$$

where $\{\lambda\}$ is the set of all eigenvalues of $A^T A$.

Proof. Math 571.

Tuesday, 9/24/13

Example 6: $A = \begin{bmatrix} 3 & -4 \\ 1 & 0 \end{bmatrix} \Rightarrow \|A\|_\infty = \max\{|3| + |-4|, |1| + |0|\} = 7.$

$\boldsymbol{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow A\boldsymbol{x} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \Rightarrow \dfrac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \dfrac{3}{1} = 3$

$\boldsymbol{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow A\boldsymbol{x} = \begin{bmatrix} -4 \\ 0 \end{bmatrix} \Rightarrow \dfrac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \dfrac{4}{1} = 4$

$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow A\boldsymbol{x} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \Rightarrow \dfrac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \dfrac{1}{1} = 1$

$\boldsymbol{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \Rightarrow A\boldsymbol{x} = \begin{bmatrix} 7 \\ 1 \end{bmatrix} \Rightarrow \dfrac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \dfrac{7}{1} = 7$: max amplification factor according to theorem

△

Example 7: $A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} \Rightarrow A^T A = \begin{bmatrix} 1 & 0 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix} \Rightarrow \|A\|_2 = 2.9208$ (Matlab)

$\boldsymbol{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow A\boldsymbol{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow \dfrac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \dfrac{1}{1} = 1$

$\boldsymbol{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow A\boldsymbol{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \Rightarrow \dfrac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \dfrac{\sqrt{8}}{1} = 2.8284$

$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow A\boldsymbol{x} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \Rightarrow \dfrac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \dfrac{\sqrt{13}}{\sqrt{2}} = 2.5495$

△

4 Error analysis

Text : section 3.4

We are still considering one of the fundamental linear algebra problems, $A\boldsymbol{x} = \boldsymbol{b}$, where A is nonsingular.
Let $\boldsymbol{x}$ denote the exact solution and $\tilde{\boldsymbol{x}}$ represent an approximate solution.

Definition 8. Absolute error and relative error are defined as

$$E_A = \tilde{x} - x, \qquad E_R = \frac{\tilde{x} - x}{x}, \quad x \neq 0.$$

Interpretation:
In some texts, you may see error defined as the opposite sign of the above, which is fine; what matters is consistency.
Using our definition, E_A is the error, for example, and $-E_A = x - \tilde{x}$ is the correction which should be added to get rid of the error.

Definition 9. The residual is defined as

$$\boldsymbol{r} = A\tilde{\boldsymbol{x}} - \boldsymbol{b}.$$

It measures the amount by which an approximation fails to satisfy the equations.

Notes:

- $A\boldsymbol{E_A} = \boldsymbol{r}$, pf: $AE_A = A(\tilde{\boldsymbol{x}} - \boldsymbol{x}) = A\tilde{\boldsymbol{x}} - A\boldsymbol{x} = A\tilde{\boldsymbol{x}} - \boldsymbol{b} = \boldsymbol{r}$ ok
- Then $E_A = 0$ if and only if $\boldsymbol{r} = 0$ (why?)
- $\|\boldsymbol{r}\| \ll 1$ does not mean that $\|E_A\| \ll 1$.

Example 8: Consider $A = \begin{bmatrix} 1.01 & 0.99 \\ 0.99 & 1.01 \end{bmatrix}$ with $\boldsymbol{b} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$. The exact solution is $\boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

Suppose we've come up with $\tilde{\boldsymbol{x}}_1 = \begin{bmatrix} 1.01 \\ 1.01 \end{bmatrix}$ as an approximate solution. Then

$$e_1 = \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix} \Rightarrow \|e_1\|_\infty = 0.01, \qquad \boldsymbol{r}_1 = A\tilde{\boldsymbol{x}}_1 - \boldsymbol{b} = \begin{bmatrix} 0.02 \\ 0.02 \end{bmatrix} \Rightarrow \|\boldsymbol{r}_1\|_\infty = 0.02$$

Suppose another method gives $\tilde{\boldsymbol{x}}_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ as a different approximate solution. Then

$$e_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \Rightarrow \|e_2\|_\infty = 1, \qquad \boldsymbol{r}_2 = A\tilde{\boldsymbol{x}}_2 - \boldsymbol{b} = \begin{bmatrix} 0.02 \\ -0.02 \end{bmatrix} \Rightarrow \|\boldsymbol{r}_2\|_\infty = 0.02$$

In both cases, the residual's maximum norm is the same, but in case 2, the error is 100 times greater than case 1.

△

Question 2: How large can $\|e\|$ be?

Definition 10. The condition number of a matrix A is defined as

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

and depends on the chosen matrix norm.

Theorem 11. *For nonzero $\boldsymbol{x}$ and $\boldsymbol{b}$, the following holds*

$$\frac{\|e\|}{\|\boldsymbol{x}\|} \leq \kappa(A) \frac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|}.$$

Proof.

$$\|\boldsymbol{b}\| = \|A\boldsymbol{x}\| \leq \|A\| \cdot \|\boldsymbol{x}\| \Rightarrow \|\boldsymbol{x}\| \geq \frac{\|\boldsymbol{b}\|}{\|A\|}$$

$$Ae = \boldsymbol{r} \Rightarrow e = A^{-1}\boldsymbol{r} \Rightarrow \|e\| = \left\|A^{-1}\boldsymbol{r}\right\| \leq \left\|A^{-1}\right\| \cdot \|\boldsymbol{r}\|$$

$$\frac{\|e\|}{\|\boldsymbol{x}\|} \leq \frac{\left\|A^{-1}\right\| \cdot \|\boldsymbol{r}\|}{\|\boldsymbol{b}\| \,/\, \|A\|} = \kappa(A) \cdot \frac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|}$$

□

Notes:

- For $A = \begin{pmatrix} 1.01 & 0.99 \\ 0.99 & 1.01 \end{pmatrix}$, we have

  $\|A\|_\infty = 2, \quad A^{-1} = \frac{1}{0.04}\begin{pmatrix} 1.01 & -0.99 \\ -0.99 & 1.01 \end{pmatrix} \Rightarrow \left\|A^{-1}\right\|_\infty = 50$

  $\kappa_\infty(A) = \|A\|_\infty \cdot \left\|A^{-1}\right\|_\infty = 2 \cdot 50 = 100$

- Perturbations in right-hand-side, $\boldsymbol{b}$.

  $$\left.\begin{array}{l} Ax = b \\ A\tilde{x} = \tilde{b} \end{array}\right\} \Rightarrow \frac{\|x - \tilde{x}\|}{\|x\|} \leq \kappa(A) \frac{\left\|b - \tilde{b}\right\|}{\|b\|}$$

- Perturbations in matrix (hw)

  $$\left.\begin{array}{l} Ax = b \\ \tilde{A}\tilde{x} = b \end{array}\right\} \Rightarrow \frac{\|x - \tilde{x}\|}{\|\tilde{x}\|} \leq \kappa(A) \frac{\left\|A - \tilde{A}\right\|}{\|A\|}$$

- Hence the matrix condition number $\kappa(A)$ controls the error in the solution due to perturbations in either A or b.

Example 9: Recall the 2×2 matrix $A = \begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix}$ and right-hand side vector $\boldsymbol{b} = \begin{pmatrix} 1+\epsilon \\ 2 \end{pmatrix}$ where $0 < \epsilon \ll 1$.

1. The exact solution is given by the usual process of elimination.

   $$\left(\begin{array}{cc|c} \epsilon & 1 & 1+\epsilon \\ 1 & 1 & 2 \end{array}\right) \rightarrow \left(\begin{array}{cc|c} \epsilon & 1 & 1+\epsilon \\ 0 & 1-\frac{1}{\epsilon} & 1-\frac{1}{\epsilon} \end{array}\right) \Rightarrow \left.\begin{array}{l} x_1 = \frac{1+\epsilon-1}{\epsilon} = 1 \\ x_2 = \frac{1-1/\epsilon}{1-1/\epsilon} = 1 \end{array}\right\} : \text{exact solution}$$

2. In the presence of roundoff error, $1 - \dfrac{1}{\epsilon} \approx -\dfrac{1}{\epsilon}$ and $1 + \epsilon \approx 1$, and we get

   $$\left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 0 & -\frac{1}{\epsilon} & -\frac{1}{\epsilon} \end{array}\right) \Rightarrow \left.\begin{array}{l} \tilde{x}_1 = 0 \\ \tilde{x}_2 = 1 \end{array}\right\} : \text{computed solution, inaccurate}$$

   Explanation: $A = \begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix}, \quad A^{-1} = \dfrac{1}{1-\epsilon}\begin{pmatrix} 1 & -1 \\ -1 & \epsilon \end{pmatrix} \Rightarrow \kappa_\infty(A) = 2 \cdot \dfrac{2}{1-\epsilon} \approx 4$

   The condition number is not too large; however, Gaussian elimination reduces A to a row-equivalent upper triangular matrix U,

   $U = \begin{pmatrix} \epsilon & 1 \\ 0 & 1-\frac{1}{\epsilon} \end{pmatrix}, \quad U^{-1} = \dfrac{1}{\epsilon - 1}\begin{pmatrix} 1-\frac{1}{\epsilon} & -1 \\ 0 & \epsilon \end{pmatrix}$

   $\Rightarrow \kappa_\infty(U) = \left|1 - \frac{1}{\epsilon}\right| \cdot \frac{1}{\epsilon - 1}\left(\left|1 - \frac{1}{\epsilon}\right| + 1\right) \approx \frac{1}{\epsilon^2}$,

   thus $\kappa_\infty(U)$ can be larger than $\kappa_\infty(A)$. Hence a small change in the entries of the matrix or the RHS of the reduced system (e.g., due to roundoff error) can produce a large change in the computed solution, as we have seen in this example.

   This means that Gaussian elimination is an unstable algorithm for solving $Ax = b$ because it can replace a well-conditioned matrix A with an ill-conditioned matrix U.

3. Gaussian elimination combined with partial pivoting produces a different result.

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ \epsilon & 1 & 1 \end{array}\right) \to \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1-\epsilon & 1-\epsilon \end{array}\right) \Rightarrow \left.\begin{array}{l} \tilde{x}_1 = 1 \\ \tilde{x}_2 = 1 \end{array}\right\} : \text{exact solution}$$

$$U = \begin{pmatrix} 1 & 1 \\ 0 & 1-\epsilon \end{pmatrix}, \quad U^{-1} = \frac{1}{1-\epsilon}\begin{pmatrix} 1-\epsilon & -1 \\ 0 & 1 \end{pmatrix}$$

$\Rightarrow \kappa_\infty(U) \approx 2 \cdot 2 = 4 \approx \kappa_\infty(A)$

In fact, for most cases Gaussian elimination + partial pivoting + IEEE arithmetic is stable in most cases.

△

5 LU Factorization

Text : section 3.5

The matrix form of Gaussian elimination is known as LU factorization. The goal is to replace the matrix A with two triangular factors, a lower triangular matrix L and an upper triangular matrix U such that $A = LU$.

Question 3: Why?
Suppose that there exist matrices L and U such that $A = LU$, and that we want to solve the equation $Ax = b$ for a variety of right-hand side vectors b, some of which we don't know yet (for example, in a time-dependent problem).

Algorithm 3: Solving $Ax = b$ with LU factorization

Input : Lower triangular matrix L and upper triangular matrix U such that $A = LU$, right hand side vector b
Output: Vector x such that $Ax = b$.

1 Solve $Ly = b$ by forward substitution. `% Note:` $O(n^2)$ `operations`
2 Solve $Ux = y$ by backward substition. `% Note:` $O(n^2)$ `operations`
3 **return** x

We know that Gaussian elimination requires $O(n^3)$ operations; by using LU factorization we only have to incur that cost once. Subsequent solutions of $Ax = b$ only require $O(n^2)$ operations once the factors L and U are known.

Example 10: We consider the $n = 3$ case, but the general $n \times n$ case is similar.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

step 1: eliminate variable x_1 from equations 2 and 3.

$$m_{21} = \frac{a_{21}}{a_{11}}, \qquad m_{31} = \frac{a_{31}}{a_{11}}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \boxed{a_{22}} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix}$$

step 2: eliminate x_2 from equation 3.

$m_{32} = \dfrac{a_{32}}{a_{22}}$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & \boxed{a_{33}} \end{pmatrix} = U, \text{ upper triangular}$$

$$\Rightarrow L_2 L_1 A = U, \text{ where } L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix}$$

$\Rightarrow A = L_1^{-1} L_2^{-1} U$

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix}, \quad M_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{pmatrix}, \text{ check : } M_1 M_1^{-1} = I$$

$$M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix}, \quad M_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{pmatrix}$$

Then

$$M_2^{-1} M_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix} = L, \text{ lower triangular}$$

Final result: $A = LU$.

Notes:

- The LU factorization algorithm does not require additional storage to output the result; the matrix L can be stored in the below-diagonal elements of the input matrix A (we know the diagonal of L is unity), and the elements of U are written to the diagonal and above-diagonal elements of A

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \mapsto \underbrace{\begin{pmatrix} u_{11} & u_{12} & u_{13} \\ m_{21} & u_{22} & u_{23} \\ m_{31} & m_{32} & u_{33} \end{pmatrix}}_{L} U$$

Thursday, 9/26/13

- Collect homework 2
  - discuss results of problem 6
- Distribute homework 3
- Distribute computing project 1: don't wait on this one.

- Correct typo in matrix perturbation.
- Rename intermediate lower triangular factors as $M_1, M_2, \ldots$ to match textbook.

Example 11: $A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \mapsto \begin{pmatrix} 2 & -1 & 0 \\ 0 & 3/2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \mapsto \begin{pmatrix} 2 & -1 & 0 \\ 0 & 3/2 & -1 \\ 0 & 0 & 4/3 \end{pmatrix}$

$m_{21} = -\frac{1}{2}, m_{31} = 0 \qquad m_{32} = -\frac{1}{3/2} = -\frac{2}{3}$

$L = \begin{pmatrix} 0 & 1 & 0 \\ -1/2 & 1 & 0 \\ 0 & -2/3 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 2 & -1 & 0 \\ 0 & 3/2 & -1 \\ 0 & 0 & 4/3 \end{pmatrix} \quad LU = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} = A$

Now we use L and U to solve $Ax = b$ with $b = [1, 0, 1]^T$ (previously, we used Gaussian elimination).

$Ly = b \Rightarrow \begin{pmatrix} 0 & 1 & 0 \\ -1/2 & 1 & 0 \\ 0 & -2/3 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \Rightarrow y = \begin{pmatrix} 1 \\ 1/2 \\ 4/3 \end{pmatrix}$

$Ux = y \Rightarrow \begin{pmatrix} 2 & -1 & 0 \\ 0 & 3/2 & -1 \\ 0 & 0 & 4/3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1/2 \\ 4/3 \end{pmatrix} \Rightarrow x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

△

Question 4: What about partial pivoting?

The LU decomposition represents Gaussian elimination as a set of matrix operations that are equivalent to row-reduction; to exchange rows (i.e., to implement a pivoting strategy), we use permutation matrices. Instead of $A = LU$, the final result is $PA = LU$, where P is a permutation matrix.

Example 12: $A = \begin{pmatrix} 0 & 4 & -15 \\ 10 & 0 & 15 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -12 \\ 100 \\ 0 \end{pmatrix}$

We want to interchange rows 1 and 2.

$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 4 & -15 \\ 10 & 0 & 15 \\ 1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 1 & -1 & -1 \end{pmatrix}, \quad P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

$\begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 1 & -1 & -1 \end{pmatrix} \to \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 0 & -1 & -2.5 \end{pmatrix} \to \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 0 & 0 & -6.25 \end{pmatrix} = U$

$m_{21} = \frac{0}{10} = 0$
$m_{31} = \frac{1}{10} = 0.1$
$m_{32} = \frac{-1}{4} = -0.25 \qquad \Rightarrow L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.1 & -0.25 & 1 \end{pmatrix}$

check : $PA = LU$
Then $Ax = b \Rightarrow PAx = Pb \Rightarrow LUx = Pb$, and we apply forward and backward substitution to find x.

$$Ly = Pb \Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.1 & -0.25 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 100 \\ -12 \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 100 \\ -12 \\ -13 \end{pmatrix}$$

$$Ux = y \Rightarrow \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 0 & 0 & -6.25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 100 \\ -12 \\ -13 \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6.88 \\ 4.80 \\ 2.08 \end{pmatrix}$$

Notes:

- If pivoting is required in more than one step, we proceed as follows:
  $M_2P_2M_1P_1A = U$, but it can be shown that $P_2M_1 = \tilde{M}_1P_2$ (hw).
  $\Rightarrow M_2\tilde{M}_1P_2P_1A = U \quad \Rightarrow PA = LU$, where $P = P_2P_1$, and $L = \tilde{M}_1^{-1}M_2^{-1}$

△