Fall 2019

Course Summary

CSCC37 - Introduction to Numerical Algorithms for Computational Mathematics

Computer & Mathematical Sciences
UNIVERSITY OF TORONTO
SCARBOROUGH

Instructor:	**Richard Pancer**
Email:	pancer@utsc.utoronto.ca
Office:	TBA
Office Hours:	Mon 17:10 - 18:30

1 What is Numerical Analysis?

Physical Systems are typically transformed into **Mathematical Models** by engineers, chemists, physicists...
Mathematical Models can then have a closed form solution (rarely!), however most times there will only be a numerical solution which requires **Numerical Analysis**

2 Floating Point Arithmetic

2.1 Representation of non-negative integers

Decimal System (Base 10): $350 = (350)_{10} = 3 \cdot 10^2 + 5 \cdot 10^1 + 0 \cdot 10^0$
Binary System (Base 2): $350 = (101011110)_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + \cdots + 1 \cdot 2^1 + 0 \cdot 2^0$

In general, we have **Base b System**, for $b > 0, b \in \mathbb{N}$
$x = (d_n d_{n-1} \dots d_0)_b = d_n \cdot b^n + d_{n-1} + \cdots + d_0 \cdot d^0$
where $x \geq 0, x \in \mathbb{N}, 0 \leq d_i < b, i \in [1, n]$

Hexadecimal System (Base 16): Symbols $0, 1, \dots, 8, 9, A, B, C, D, E, F$
$(8FA)_{16} = 8 \cdot 16^2 + F \cdot 16^1 + A \cdot 16^0 = 8 \cdot 16^2 + 15 \cdot 16^1 + 10 \cdot 16^0 = (2298)_{10}$

Converting decimal to base b
Example, $(350)_{10}$ in base $b = 2$

Numerator	Denominator	Quotient	Remainder
350	2	175	0
175	2	87	1
87	2	43	1
43	2	21	1
21	2	10	1
10	2	5	0
5	2	2	1
2	2	1	0
1	2	0	1

Once we hit 0 in the **Quotients** column, we simply read the Remainder column from bottom up and we have our conversion. Thus $(350)_{10} = (101011110)_2$.
This algorithm is **safe** (will always terminate). Where **overflow** is the only potential problem.

2.2 Representation of real numbers

if $x \in \mathbb{R}$ then $x = \pm(x_I.x_F)_b = \pm(d_n d_{n-1} \dots d_0.d_{-1} d_{-2} \dots)_b$
Where x_I is the integral part and x_F is the fractional part. The sign ($+$ or $-$) is a single bit 0 or 1
x_I is the non-negative integer talked about above, and x_F can have infinite digits.
Example: $(0.77\dots)_{10} = (0.\bar{7})_{10} = 7 \cdot 10^{-1} + 7 \cdot 10^{-2} + \cdots$
Binary System (Base 2): $(0.77\dots)_{10} = (.000110011\dots)_2 = (0.00\bar{0}\bar{1}1)_2 = 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + \cdots$
In general, we have **Base b System**, for $b > 0, b \in \mathbb{N}$

$$x_F = (.d_{-1} d_{-2} d_{-3} \dots)_b = d_{-1} \cdot b^{-1} + d_{-2} \cdot b^{-2} + \cdots = \sum_{i=1}^{\infty} d_{-i} \cdot b^{-i}$$

We say that a **Binary Fraction** with n digits is terminating if it does not have any cycling and is finite, furthermore it has a terminating decimal representation.

Converting Decimal Fractions to base b
Example, $(0.625)_{10}$ in base $b = 2$

Multiplier	Base	Product	Integral	Fraction
0.625	2	1.25	1	0.25
.25	2	0.5	0	0.5
0.5	2	1	1	0

Once we hit 0 in the **fractions** column, we simply read the Integral column from top down and we have our conversions. Thus $(0.625)_{10} = (.101)_2$
However you may not always hit 0 as this algorithm is **not safe** (won't always terminate)!
For example: What is $(0.1)_{10}$ in binary?

Multiplier	Base	Product	Integral	Fraction
0.1	2	0.2	0	0.2
0.2	2	0.4	0	0.4
0.4	2	0.8	0	0.8
0.8	2	1.6	1	0.6
0.6	2	1.2	1	**0.2**

We see that **0.2** occurs again as a fraction. This means that there must be a cycle! Thus $(0.1)_{10} = (0.00\bar{0}11)_2$

2.3 Machine Representation of Reals

Reals represented in computers as Floating Point numbers (FP for short).
A FP x in base b has the form: $x = (F)_b \cdot b^{(e)_b}$, where F is the fraction and has the form $F = \pm(d_1, d_2 \cdots d_t)_b$ such F is also called the **mantissa**.
and $e = \pm(c_s c_{s-1} \dots c_1)_b$ is called the **exponent**

A FP number is **normalized** if $d_1 \neq 0$ unless $d_1 = d_2 = \cdots = d_t = 0$

Significant Digits: (Of a nonzero FP) are the digits following and including the first nonzero digit. In the case of a normalized FP, all digits of the mantissa is significant (storing useful information).

Absolute value of mantissa is always ≥ 0 and < 1
Exponent is limited: $e \in [-M, M]$ where $M = (a_s a_{s-1} \cdots a_1)_b$ with each $a_i = (b-1)$

Largest FP number in absolute value defined by this system is $(.aa\dots a)_b \cdot b^{(aa\dots a)_b}$ where each $a = (b-1)$
Smallest nonzero normalized FP number in absolute value is $(0.10\dots 0)_b \cdot b^{-(aa\dots a)_b}$, where each $a = (b-1)$
Non-normalized FPs allow us to get very very close to 0, consider $(.00\dots 01)_b \cdot b^{-(aa\dots a)_b}$

$\mathbb{R}_b(t, s)$ denotes the set of all floating point numbers with t digit mantissa and s digit exponent.
And any $\mathbb{R}_b(t, s)$ is finite, while $\mathbb{R}$ is infinite.

Overflow or **Underflow** occurs whenever a nonzero floating point number with absolute value outside these ranges must be stored on the computer.
Note that when the number is too close to zero it's called underflow. not when its largely negative.

Furthermore, $\mathbb{R}$ is compact, where as $\mathbb{R}_b(t, s)$ is not. This is to say that between any two real numbers, there is an infinite number of reals in between. (infinite density).

A real number $x = \pm(x_I.x_F)_b = \pm(d_k d_k{-}1 \dots d_0.d_{-1}d_{-2}\dots)_b$
We say we convert a real x to a fp number with the following notation, $x \in \mathbb{R} \rightarrow Fl(x) \in \mathbb{R}_b(t, s)$

We can represent this in $\mathbb{R}_b(t,s)$ by the following algorithm:

1. Normalizing the mantissa.
   We shift the decimal point, and readjust with the exponent
   $x = \pm(d_k d_{k-1} \dots d_0.d_{-1}d_{-2}\dots)_b = \pm(.D_1D_2\dots)_b \cdot b^{k+1}$
2. chop or round the mantissa
   (a) chopping - chop after digit t of the mantissa.
   (b) rounding - chop after digit t then round D_t depending on whether $D_{t+1} \geq b/2$ or $D_{t+1} < b/2$.
   A more efficient way to round is to add $b/2$ to D_{t+1}, then simply chop.

Example
Consider $Fl(3/2) \in \mathbb{R}_{10}(2,4) = \begin{cases} +0.66 & \text{if chop} \\ +0.67 & \text{if round} \end{cases}$
Round off Error
Precisely, this is the difference between an $x \in \mathbb{R}$ and $FL(x) \in \mathbb{R}_b(t,s)$.
Typically measured relative to x as $\dfrac{x - FL(x)}{x} = \delta$ or $FL(x) = x(1-\delta)$ where δ is the relative round off.
δ can be bound independently of x, $\delta < b^{1-t}$ for chopping normalized FPs.
$|\delta| < \dfrac{1}{2}b^{1-t}$ for rounding normalized FPs.

2.4 Machine Arithmetic

Let $x, y \in \mathbb{R}$ and $FL(x), FL(y) \in \mathbb{R}_b(t,s)$

Consider operations $\circ \in \{+,-,\times,/\}$, the computer gives $x \circ y \approx FL(FL(x) \circ FL(y))$
Example in $\mathbb{R}_{10}(2,4)$
let $x = 2, y = 0.0000058 \in \mathbb{R}$ we have $x + y = 2.0000058$
then $FL(x) = (+0.20 \cdot 10^1)_{10}$ and $FL(y) = (+0.58 \cdot 10^{-5})_{10}$ Note that no RROs in $FL(x), FL(y)$
then $FL(x) + FL(y) = 0.20 \cdot 10^1 + 0.58 \cdot 10^{-5} = 0.20000058 \cdot 10^1$ stored temporarily.
finally, $FL(FL(x) + FL(y)) = (+0.20 \cdot 10^1)_{10}$

2.5 Machine Precision

Definition: The smallest **non-normalized** FP number *eps* such that $1 + eps > 1$. This number (eps) is referred to as machine epsilon.
This is important as this number $eps = \begin{cases} b^{1-t} & \text{chopping} \\ \frac{1}{2}b^{1-t} & \text{rounding} \end{cases}$
recall $\delta = \dfrac{x - FL(x)}{x}$ is the **RRO** where δ can be bound independently.
$0 \leq \delta \leq eps$ for chopping, and $|\delta| \leq eps$ for rounding.

How exactly is error propagated in each of $\circ \in \{+,-,\times,/\}$?
Let $x, y \in \mathbb{R}$, then $Fl(x) = x(1-\delta), Fl(y) = y(1-\delta)$
Multiplication: $x \cdot y$, where computer gives $FL(FL(x) \cdot FL(y))$
We have $[x(x-\delta_x) \cdot y(1-\delta_y)](1-\delta_{xy}) = x \cdot y(1-\delta_x)(1-\delta_y)(1-\delta_{xy}) \approx x \cdot y(1 - \delta_x - \delta_y - \delta_{xy}) = x \cdot y(1-\delta_.)$
We say this is a good approximation as each δ is a small fraction bound by machine epsilon, there's a good chance that products of δs will be so small that they are lower than such machine epsilon.
Where $|\delta| \leq 3 \cdot eps$, this is the worst case scenario.

Addition: $x+y$, where computer gives $FL(FL(x)+FL(y))$
We have $(x(1-\delta_x)+y(1-\delta_y))(1-\delta_{xy}) = x(1-\delta_x)(1-\delta_{xy})+y(1-\delta_y)(1-\delta_{xy}) \approx x(1-\delta_x-\delta_{xy})+y(1-\delta_y-\delta_{xy})$
$= (x+y)\left[1-\frac{x(1-\delta_x-\delta_{xy})}{x+y}+\frac{y(1-\delta_y-\delta_{xy})}{x+y}\right] = (x+y)[1-\delta_+]$
Where $|\delta_+| \le \left|\frac{x}{x+y}\right| 2\cdot eps + \left|\frac{y}{x+y}\right| 2\cdot eps = \frac{|x|+|y|}{|x+y|}2\cdot eps$

i.e. $|\delta_+| \le \begin{cases} 2\cdot eps & \text{x and y are the same sign} \\ 2\cdot eps\frac{|x-y|}{|x+y|} & \text{x and y are different signs} \end{cases}$
We see that when $x \approx -y$, the error bound could approach infinity. This is called subtractive cancellation.

Example of Subtractive Cancellation in $\mathbb{R}_{10}(3,1)$ with rounding.
Compute $a^2-2ab+b^2$ with $a = 15.6, b = 15.7 \in \mathbb{R}$
first we have $fl(a) = +(0.156\cdot 10^2)$ and $fl(b) = +(0.157\cdot 10^2)$ with no initial round off.

In $\mathbb{R}_{10}(3,1), fl(a^2) \to fl(243.36) = +(0.243\cdot 10^3)$
$fl(2ab) \to fl(489.84) = +(0.490\cdot 10^3)$
$fl(b^2) \to fl(246.49) = +(0.246\cdot 10^3)$
$fl(a^2-2ab+b^2) \to fl(243-490+246) = -(0.100.10^1)$
But this is a problem as $a^2-2ab+b^2 = (a-b)^2$ which is positive.

3 Linear Systems

$A\vec{x} = \vec{b}, A \in \mathbb{R}^{n\times n}, \vec{x}, \vec{b} \in \mathbb{R}^n$, Given $A, \vec{b}$, calculate $\vec{x}$
General Solution Technique:

1. Reduce the problem to an equivalent one that is easier to solve.
2. Solve the reduced problem.

Ex, given $\begin{matrix} 3x_1-5x_2 = 1 \\ 6x_1-7x_2 = 5 \end{matrix} \Leftrightarrow \begin{bmatrix} 3 & -5 \\ 6 & -7 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & -5 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \Rightarrow x_1 = 1, x_1 = 2$
Generalize this to n equations in n unknowns.
Let matrix $A = [a_{ij}]$ where a_{11} refers to the top left element.

We have:
Equation 1: $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$
Equation 2: $a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$
$\vdots$
Equation n: $a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$
Reduce system to triangular form.

1. Assume that $a_11 \ne 0$
   multiply Equation 1 by $\frac{a_{21}}{a_{11}}$ and subtract that from Equation 2.
   multiply Equation 1 by $\frac{a_{31}}{a_{11}}$ and subtract that from Equation 3.
   $\vdots$
   multiply Equation 1 by $\frac{a_{n1}}{a_{11}}$ and subtract that from Equation n.
   This results in a equivalent system, where x_1 has been eliminated from Equations 2 to n.

Now we have:

Equation 1: $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$

Equation $\hat{2}$: $0 + \hat{a}_{22}x_2 + \cdots + \hat{a}_{2n}x_n = \hat{b}_2$

$\vdots$

Equation $\hat{n}$: $0 + \hat{a}_{n2}x_2 + \cdots + \hat{a}_{nn}x_n = \hat{b}_n$

2. Assume that $\hat{a}_{22} \neq 0$

   multiply Equation $\hat{2}$ by $\frac{\hat{a}_{32}}{\hat{a}_{22}}$ and subtract that from Equation $\hat{3}$.

   multiply Equation $\hat{2}$ by $\frac{\hat{a}_{n2}}{\hat{a}_{22}}$ and subtract that from Equation $\hat{n}$.

$\vdots$

finally, at step $n-1$, we assume $\tilde{a}_{n-1,n-1} \neq 0$, multiply equation $n \tilde{-} 1$ by $\frac{\tilde{a}_{n,n-1}}{\tilde{a}_{n-1.n-1}}$ and subtract from Equation $\tilde{n}$
this will result in an upper triangular system.
Equation 1: $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$
Equation $\hat{2}$: $0 + \hat{a}_{22}x_2 + \cdots + \hat{a}_{2n}x_n = \hat{b}_2$

$\vdots$

Equation $\tilde{n}$: $0 + 0 + \cdots + \tilde{a}_{nn}x_n = \tilde{b}_n$

Another way of looking at this problem is using matrix vector notation, looking to solve $A\vec{x} = \vec{b}$
where $A \in \mathbb{R}^{n\times n}, \vec{b}, \vec{x} \in \mathbb{R}^n$
Triangulating such A requires the following steps:

1. Eliminate the first column of A below a_{11}

$$L_1 A\vec{x} = L_1\vec{b}, \text{ where } L_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -a_{21}/a_{11} & 1 & \dots & 0 \\ -a_{31}/a_{11} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -a_{n1}/a_{11} & 0 & \dots & 1 \end{bmatrix}$$

   This is very similar to the identity matrix, except the first column is filled with the multipliers used in the first step.

2. Eliminate the second column of L_1A below $\hat{a}_{22}$

$$L_2(L_1A)\vec{x} = L_2(L_1\vec{b}) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & \hat{a}_{22} & \dots & \hat{a}_{2n} \\ 0 & 0 & \hat{\hat{a}}_{33} & \hat{\hat{a}}_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \hat{\hat{a}}_{n3} & \hat{\hat{a}}_{nn} \end{bmatrix}, \text{ where } L_2 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & -\hat{a}_{32}/\hat{a}_{22} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & -\hat{a}_{n2}/\hat{a}_{22} & \dots & 1 \end{bmatrix}$$

$\vdots$

We continue until we have $L_{n-1}L_{n-2}\dots L_1A\vec{x} = L_{n-1}L_{n-2}\dots L_1\vec{b}$
let $L_{n-1}L_{n-2}\dots L_1A = U$, where U is an upper triangular matrix. This becomes very easy to solve.

3.1 LU Factorization

We have $L_{n-1}L_{n-2}\dots L_1A = U \Leftrightarrow A = L_1^{-1}L_2^{-1}\dots L_{n-1}^{-1}U$
Lemma 1: If L_i is a Gauss Transform. Then L_i^{-1} exists and is also a Gauss Transform.
Lemma 2: If L_i and L_j are Gauss Transforms, and $i < j$, $L_iL_j = L_i + L_j - I$
Then $A = L_1^{-1}L_2^{-1}\dots L_{n-1}^{-1}U = LU$

Using $A = LU$ to solve $A\vec{x} = \vec{b}$

$$A\vec{x} = \vec{b} \Leftrightarrow LU\vec{x} = \vec{b}\text{ , let } \vec{d} = \vec{b} \text{ for a lower triangular } L\vec{d}$$
$$\text{Then } L\vec{d} = \vec{b} \Leftrightarrow U\vec{x} = \vec{d} \text{ for an upper triangular } U\vec{x}$$

We use LU factorization because its far cheaper, and if we have several linear systems with the same coefficient matrix A, but different right hand side, i.e. $A\vec{x} = \vec{b} = \vec{c} = \vec{d}\dots$, then we can use the same LU.

Example of LU factorization

let $A = \begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -2 & 4 & 1 \end{bmatrix}$

$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, L_1A = \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 1 \\ 0 & 3 & 2 \end{bmatrix}$

$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, L_2L_1A = \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{bmatrix}, L_2L_1A = U \Leftrightarrow A = L_1^{-1}L_2^{-1}U$

$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, L = L_1^{-1}L_2^{-1} = L_1^{-1} + L_2^{-1} - I = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}$

3.2 Gaussian Elimination with Pivoting

From last lecture, we have $L_2P_2L_1P_1A = U$
$L_2P_2L_1P_1A \equiv L_2P_2L_1P_2P_2P_1A$, as $P_2P_2 \equiv I$
Then $L_2P_2L_1P_2P_2P_1A \equiv L_2(P_2L_1P_2)P_2P_1A$

Let $\tilde{L}_1 = P_2L_1P_2$
Claim: $\tilde{L}_1$ is a Gauss Transform.
Example $P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, L_1 = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & 0 & 1 \end{bmatrix}$, then $P_2L_1P_2 = \begin{bmatrix} 1 & 0 & 0 \\ l_{31} & 1 & 0 \\ l_{21} & 0 & 1 \end{bmatrix}$ (swapped multipliers)

Them $L_2\tilde{L}_1P_2P_1A = U \Leftrightarrow P_2P_1A = \tilde{L}_1^{-1}L_2^{-1}U \Leftrightarrow PA = LU$

Now, to solve $A\vec{x} = \vec{b}$, given $PA = LU$
$A\vec{x} = \vec{b} \Leftrightarrow PA\vec{x} = P\vec{b} \Leftrightarrow LU\vec{x} = P\vec{b}$, note that $P\vec{b}$ is $\vec{b}$ with 2 elements swapped.
Then $LU\vec{x} = \vec{\tilde{b}}$, let $\vec{d} = U\vec{x}$
$L\vec{d} = \vec{b}$ is easy to solve because L is lower triangular
$U\vec{x} = \vec{d}$ is easy to solve because U is upper triangular.

What happens if at some k-th stage, the diagonal and everything below it is 0?
For example $L_{k-1}\dots L_2L_1A$, we cannot divide by the k-th row, k-th column element because its 0, furthermore every element below it is 0.
Remember our goal originally is to make every element in the k-th column under k zero, we just continue.

This will result in a matrix U with a 0 along one of the diagonal entries.

Then we have a singular matrix U, but the factorization still stays.
While its possible for U to be singular, it's not for the matrix L.
As a combination of gauss transforms, they must be invertible and thus non-singular.

In the last step, we have $U\vec{x} = \vec{d}$, but U is singular. We could have no solution, or infinitely many solutions.

Example: $U\vec{x} = \vec{d} \rightarrow \begin{bmatrix} 2 & 5 & 4 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \Rightarrow 2x_3 = b_3 \text{ and } x_2 = b_2$

Now if $b_3 = 2b_2$, we have a free parameter x_2. Otherwise, there are no solutions.
Now suppose during a numerical factorization of FPS, if all elements below the diagonal in column k and $[a_{kk}]$ at the k-th stage and are of magnitude $\leq eps \cdot max_{j\in[1,k]}|u_{jj}|$, we call this **Numerical Singularity** (Near Singularity).

3.3 Complexity of Gaussian Elimination

Let matrix A be of order $n \times n$

We will count multiplication/addition pairs, i.e. $mx + b$ as a **Flop** (Floating Point Operation).
Lets begin with factorization. Computing the LU factorization,

1st stage is the zeroing out the first column, we have $(n-1)^2$ flops as we are adding multiples of row 1 to rows 2 to n.
2nd stage is the zeroing out the first column, we have $(n-2)^2$ flops as we are adding multiples of row 2 to rows 3 to n.
$\vdots$
$(n-1)$-th stage is the zeroing out the first column, we have 1 flop.
Finally, we have $(n-1)^2 + (n-2)^2 + \cdots + 1$ total flops.

$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$, then $(n-1)^2 + (n-2)^2 + \cdots + 1 = \frac{n(n-1)(2n)}{6} = \frac{n^3}{3} + O(n^2)$

Computing the forward solve, $L\vec{d} = \vec{b}$ where L is a lower triangular matrix.

$\begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \text{o}ts & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \Leftrightarrow d_1 = b_1, d_2 = b_2 - l_{21}d_1, d_3 = b_3 - l_{31}d_1 - l_{32}d_2 \dots$

Total $= 0 + 1 + 2 + \cdots + (n-1) = \frac{n^2}{2} + O(n)$ Flops

Back-solving is similar, resulting in another $\frac{n^2}{2} + O(n)$ Flops
Total Forward/Backward solve: $n^2 + O(n)$ Flops

3.4 Round Off Error of Gaussian Elimination

Recall we have a factorization $PA = LU$ computed in a floating point system.
Because of machine round off error, we actually get $\hat{P}(A+E) = \hat{L}\hat{U}$, where $\hat{P}, \hat{L}, \hat{U}$ are the computed factors.

Then solving $A\vec{x} = \vec{b}$ becomes solving $(A+E)\hat{\vec{x}} = \vec{b}$, where $\hat{\vec{x}}$ is the computed solution.
Equivalently, let $E\hat{\vec{x}} = \vec{r}$, then $(A+E)\hat{\vec{x}} = \vec{b}$ becomes $A\hat{\vec{x}} + \vec{r} = \vec{b}$, or $\vec{r} = \vec{b} - A\hat{\vec{x}}$
Where $\vec{r}$ is the residual. (We would like this to be $\vec{0}$).

We can show that if row partial pivoting is used,

$\|E\| \lessapprox k \cdot eps \cdot \|A\|$ where k is not too large and depends on n.

And that $\|\vec{r}\| \lessapprox k \cdot eps \cdot \|b\| \Leftrightarrow \dfrac{\|r\|}{\|b\|} \lessapprox k \cdot eps$

However, this does not mean that $\|\vec{x} - \vec{\hat{x}}\|$ or $\dfrac{\|\vec{x} - \vec{\hat{x}}\|}{\|x\|}$ is small.

Example: $\begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.656 \end{bmatrix} \vec{x} = \begin{bmatrix} 0.217 \\ 0.254 \end{bmatrix}$, we know true solution is $\vec{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Consider two computed solutions, $\hat{x}_1 = \begin{bmatrix} 0.999 \\ -1.001 \end{bmatrix}, \hat{x}_2 = \begin{bmatrix} 0.341 \\ -0.087 \end{bmatrix}$

$\vec{r}_1 = \vec{b} - A\hat{x}_1 = \begin{bmatrix} -0.001243 \\ -0.001572 \end{bmatrix}, \vec{r}_2 = \vec{b} - A\hat{x}_2 = \begin{bmatrix} -0.000001 \\ 0 \end{bmatrix}$

We see that $\dfrac{\|\vec{r}_2\|}{\|\vec{b}\|}$ is much smaller than $\dfrac{\|\vec{r}_1\|}{\vec{b}}$, yet we see that $\hat{x}_2$ is a terrible solution.

But why is $\dfrac{\|\vec{x} - \vec{\hat{x}}_1\|}{\|x\|}$ so much smaller than $\dfrac{\|\vec{x} - \vec{\hat{x}}_2\|}{\|x\|}$?

We need the relationship between relative error and relative residual.
We have $A\hat{x} = \vec{b} - \vec{r}$ and $A\vec{x} = \vec{b}$, subtract the top from the bottom yields:
$A(\vec{x} - \hat{x}) = \vec{r} \Leftrightarrow \vec{x} - \hat{x} = A^{-1}\vec{r}$
Taking the norm of both sides yields: $\|\vec{x} - \hat{x}\| = \|A^{-1}\vec{r}\| \leq \|A^{-1}\|\|\vec{r}\|$ (1)
We can take the original $\vec{b} = A\vec{x} \Leftrightarrow \|\vec{b}\| = \|A\vec{x}\| \leq \|A\|\|\vec{x}\|$ (2)

Combining (1) and (2): $\dfrac{\|\vec{x} - \hat{x}\|}{\|A\|\|\vec{x}\|} \leq \dfrac{\|A^{-1}\|\|\vec{r}\|}{\|\vec{b}\|} \Leftrightarrow \dfrac{\|\vec{x} - \hat{x}\|}{\|\vec{x}\|} \leq \dfrac{\|A\|\|A^{-1}\|\|\vec{r}\|}{\|\vec{b}\|}$

Where $\|A\|\|A^{-1}\| = cond(A)$

We can try to derive a lower bound for this, again we have equations $A\vec{x} = \vec{b}$ and $A\hat{x} = \vec{b} - \vec{r}$,
Subtracting the two yields $A(\vec{x} - \hat{x}) = \vec{r} \Leftrightarrow \|A(\vec{x} - \hat{x})\| = \|\vec{r}\|$

By triangular inequality, $\|A\|\|\vec{x} - \hat{x}\| \geq \|\vec{r}\|$ (1)
We take the original $A\vec{x} = \vec{b}$, rearrange: $\vec{x} = A^{-1}\vec{b} \Rightarrow \vec{x} \leq \|A^{-1}\|\|\vec{b}\|$ (2)

Combining equations (1) and (2) yields $\dfrac{\|\vec{x} - \hat{x}\|}{\|\vec{x}\|} \geq \dfrac{\|\vec{r}\|}{\|\vec{b}\|\|A\|\|A^{-1}\|}$

Finally, combining two bounds, we have $\dfrac{\|\vec{r}\|}{\|\vec{b}\|cond(A)} \leq \dfrac{\|\vec{x} - \hat{x}\|}{\|\vec{x}\|} \leq cond(A)\dfrac{\|\vec{r}\|}{\|\vec{b}\|}$

If $cond(A)$ is very large, problem is poorly conditioned. Small relative residual does not mean small relative error.
If the $cond(A)$ is not too large, the problem is well conditioned and the small relative residual is a reliable indicator of small relative error.
Conditioning is a continuous spectrum, how large is "very large" depends on context.
Recall in the previous example: $A = \begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.656 \end{bmatrix}, A^{-1} = 10^6 \begin{bmatrix} 0.656 & -0.565 \\ -0.913 & 0.780 \end{bmatrix}$, where $10^6 = \dfrac{1}{det(A)}$

Determinant formula: $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, A^{-1} = \dfrac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ where $det(A) = 0 \Rightarrow A^{-1}$ DNE.

$\|A\|_\infty = 1.572, \|A^{-1}\| = 1.693 \cdot 10^6 \Rightarrow cond_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty = 2.66 \cdot 10^6$

$\frac{\|\vec{x} - \hat{x}\|}{\|\vec{x}\|} \leq 2.66 \cdot 10^6 \frac{\|r\|}{\|b\|}$, i.e. relative error in x could be as big as $2.66 \cdot 10^6$ times the relative residual.
Thus this A is a poorly conditioned matrix. and relative residual is not a reliable indicator of relative error.

3.5 Iterative Refinement/Iterative Improvement

Can we improve $\hat{x}$? One easy way is to improve mantissa length.
Want to solve $A\vec{x} = \vec{b}$, having already solved $(A + E)\hat{x} = \vec{b}$
Subtracting the two equations yields $A(\vec{x} - \hat{x}) = \vec{r}$, let $\vec{z} = \vec{x} - \hat{x}$, and we solve $A\vec{z} = \vec{r}$
Then $\vec{x} = \hat{x} + \vec{z}$, however this is a fallacy since we can not solve for $\vec{z}$ exactly. thus we get $\hat{z}$, and not $\vec{z}$.
But if we were to get 2 leading digits correct in $\hat{x}$, then we will get 2 leading digits correct in $\hat{z}$, and then you will have 4 leading digits correct in $\hat{x} + \hat{z}$

The Algorithm

```
Compute $\hat{x}^{(0)}$ by solving $A\vec{x} = \vec{b}$ in a floating point system.
For $i = 0, 1, 2, \dots$ until solution is good enough.
    Compute the residual: $r^{(i)} = b - a\hat{x}^{(i)}$
    Solve $A\vec{z}^{(i)} = r^{(i)}$ For some $\hat{z}^{(i)}$
    Update $\hat{x}^{(i+1)} = \hat{x}^{(i)} + \hat{z}^{(i)}$
```

4 Non Linear Equations

Problem: $F : \mathbb{R} \to \mathbb{R}$, solving for $F(\hat{x}) = 0$, where $\hat{x}$ is the root.
Linear vs non-Linear
Examples: $F(x) = 2x + 7$ is Linear, $F(x) = 2x^2 + 7$ non-Linear, $F(x) = x - e^{-x}$ non-Linear
Typically, we cannot find a "closed form" (analytic) solution to non-Linear problems. Can find iterative methods, which generate good approximate solutions $\hat{x_k}, k = 0, 1, 2, \dots$ such that they converge. $\lim_{k\to\infty} \hat{x_k} \to \hat{x}$

Consider $F(x) = x - e^{-x} = 0$, One way to get the number of roots is to roughly graph to approximate.

4.1 Fixed Point Methods

Root Finding Problem $\Leftrightarrow$ Fixed Point Problem
$F(\tilde{x}) = 0 \Leftrightarrow \tilde{x} = g(\tilde{x})$
Example: $F(x) = x - e^{-x} = 0 \Leftrightarrow x = e^{-x} = g(x)$
Can always use $g(x) = x - F(x)$ (First Form) or $g(x) = x - h(x)F(x)$ (Second Form)

First form: $F(\tilde{x}) = 0 \Leftrightarrow \tilde{x} = g(\tilde{x})$
Second Form: $F(\tilde{x}) = 0 \Leftrightarrow \tilde{x} = g(\tilde{x})$ but we could have a fixed point $\tilde{x} = g(\tilde{x})$ without $F(\tilde{x}) = 0$, see $h(\tilde{x}) = 0, F(\tilde{x}) \neq 0$

4.2 Fixed Point Iteration

Start with an approximate solution $\hat{x_0}$ then iterate:
$\quad \hat{x_{k+1}} = g(\hat{x_k}), k = 0, 1, 2, \dots$ until convergence.

Example of fix point iteration: $F(x) = x^2 + 2x - 3$ we know roots are $\hat{x} = 1, \hat{x} = -3$
Consider the fixed point iteration $x_{k+1} = x_k + \frac{(x_k)^2 + 2(x_k) - 3}{(x_k)^2 - 5}$, for $x = g(x), g(x) = x + (x^2 + 2x - 3)/(x^2 - 5)$

This is the second form, where $h(x) = -\frac{1}{x^2 - 5}$
This auxillary function cannot disappear as it is nonzero.

4.3 Fixed Point Theorem

If there is an interval $[a, b]$ such that $g(x) \in [a, b]$ for all $x \in [a, b]$
And if $\|g'(x)\| \leq L < 1$, $\forall x \in [a, b]$
Then $g(x)$ has a unique fixed point in $[a, b]$

4.4 Rate of Convergence

If $\lim\limits_{x_k \to \tilde{x}} \frac{\|\tilde{x} - x_{k+1}\|}{\|\tilde{x} - x_k\|} = c \neq 0$, then we have p-th order convergence to find fixed point $\tilde{x}$