Matrix Norms: Review

$$\|A\| \geq 0 \quad \text{(0 only for null matrix)}$$

$$\|\alpha A\| = |\alpha| \|A\|$$

$$\|A + B\| \leq \|A\| + \|B\|$$

$$\|AB\| \leq \|A\| \|B\|$$

$$\|Ax\| \leq \|A\| \|x\|$$

$$\|A\|_p = \max_{\|x\|_p = 1} \|Ax\|_p = \max_{\|x\|_p \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

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

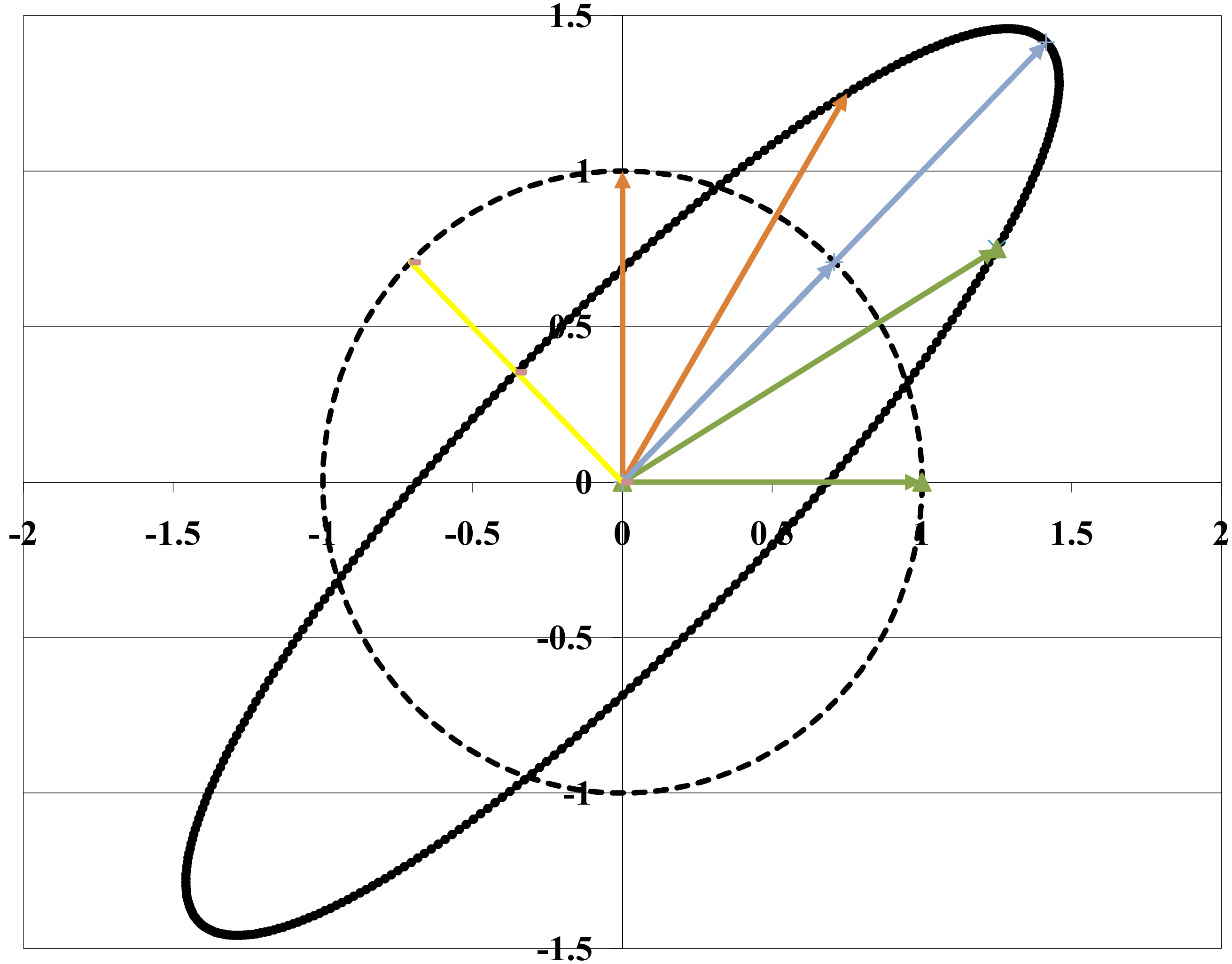
$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{\mathrm{j}=1}^{\mathrm{n}} |a_{ij}|$$

Matrix Norm – The *2*-norm

- The *2*-norm of the matrix is written as $\|A\|_2 = \max_{\|x\|_2 = 1} \|Ax\|_2$
- If we transform a unit vector, {x}, to another vector, {b}, by multiplying with matrix [A], what is maximum "length" of {b}?
- We could view [A] as operating on vector {x} to generate another vector {b}
- It will, in general, lead to a rotation as well as stretching (or shortening) of the vector {x}
- E.g., consider

$$[A] = \begin{bmatrix} 1.25 & 0.75 \\ 0.75 & 1.25 \end{bmatrix}$$

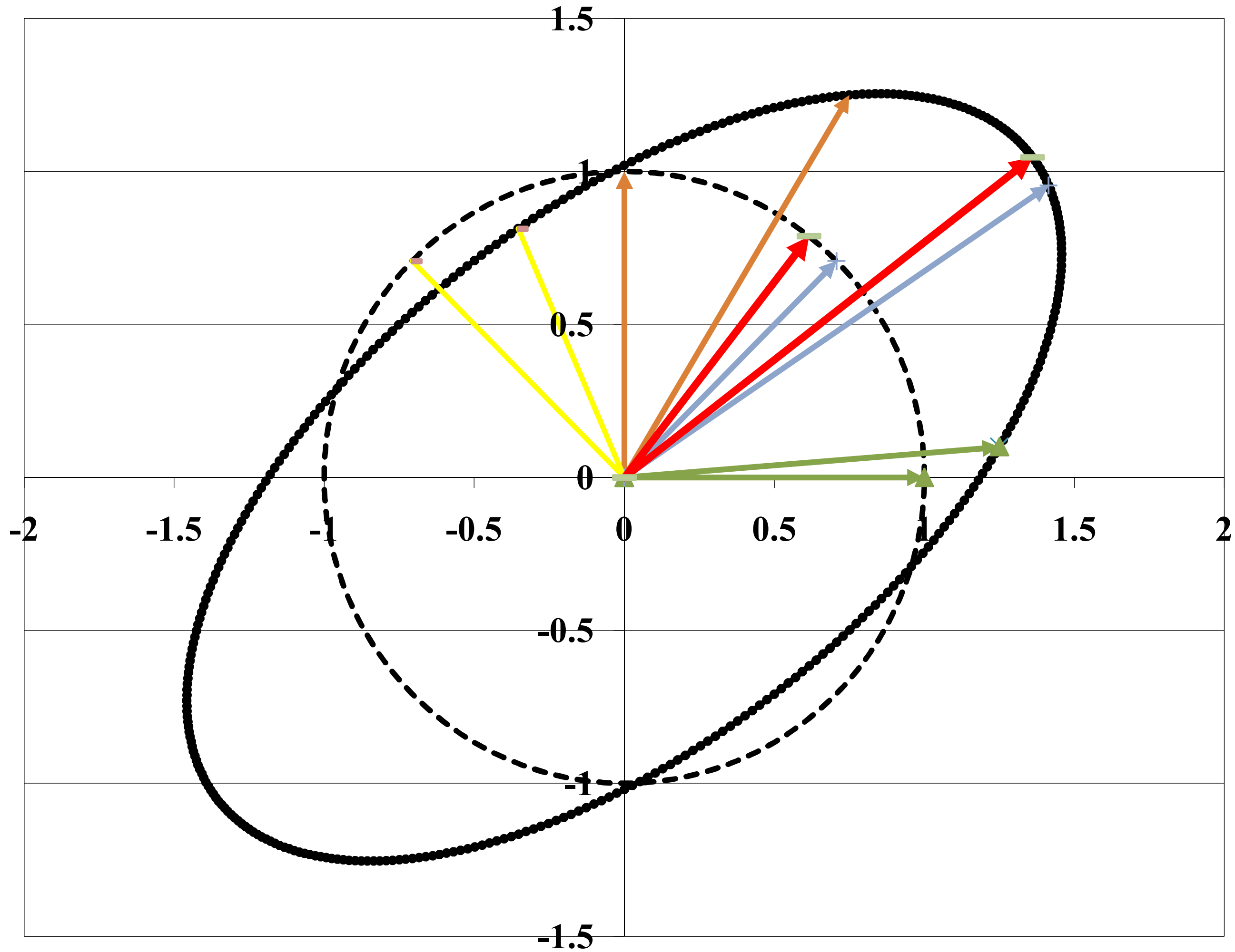
$$[A] = \begin{bmatrix} 1.25 & 0.75 \\ 0.75 & 1.25 \end{bmatrix}$$



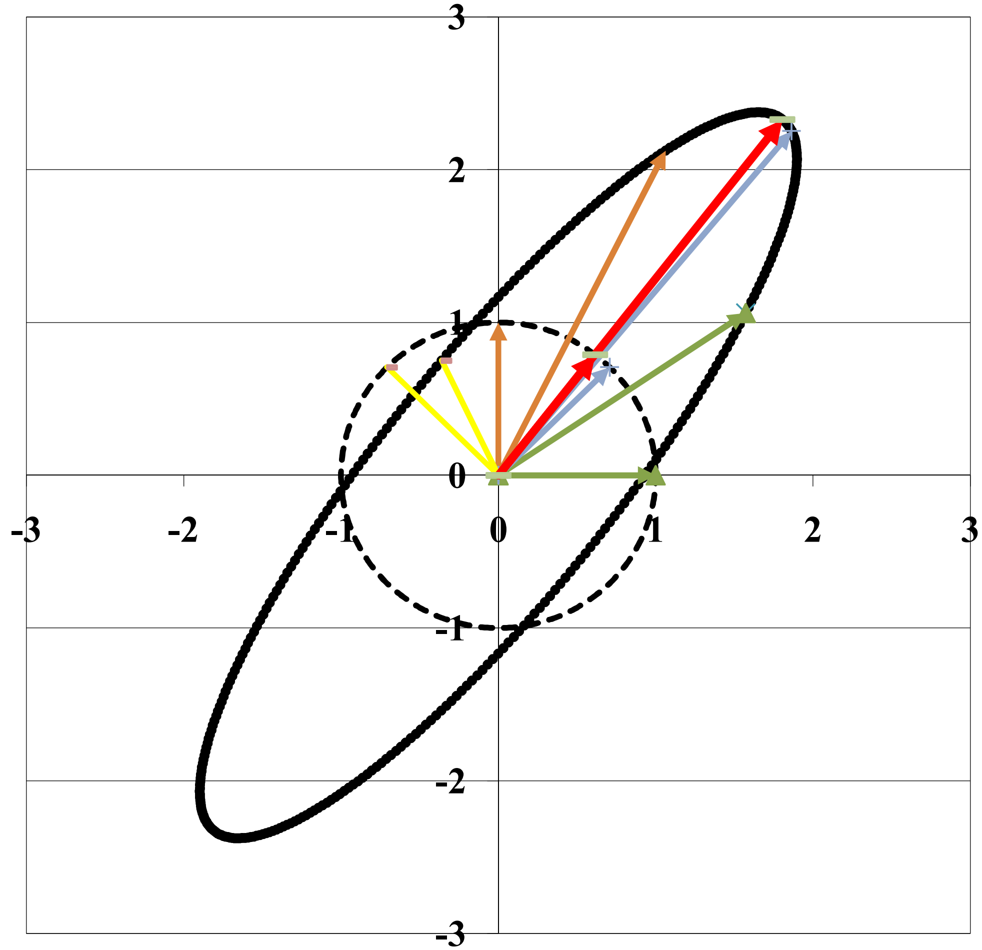
Eigenvalue

- If $Ax = \lambda x$, there is no rotation. λ is called an Eigenvalue of [A] and {x} is the corresponding Eigenvector
- For symmetric matrices, the maximum stretching and/or shortening will occur along its Eigenvectors, all of which are mutually orthogonal
- For others, it will occur along some other direction (we will see later that it is eigenvector of $A^T A$).
- How to find these will be considered after we look at the methods of solving the linear system.

$$[A]=\begin{bmatrix}1.25 & 0.75\\ 0.1 & 1.25\end{bmatrix}$$



$$[A]^T[A] = \begin{bmatrix} 1.5725 & 1.0625 \\ 1.0625 & 2.125 \end{bmatrix}$$



Matrix Norm – The *2*-norm

- The *2*-norm of the matrix is written as $\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2$
- If we transform a unit vector, {x}, to another vector, {b}, by multiplying with matrix [A], what is maximum "length" of {b}?
- This may be posed as a constrained optimization problem: Maximize $\{x\}^T[A]^T[A]\{x\}$ subject to $\{x\}^T\{x\} = 1$
- Use the Lagrange multiplier method

$$\nabla\left[x^T A^T A x - \lambda\left(x^T x - 1\right)\right] = 0 \Rightarrow A^T A x = \lambda x$$

- Which leads to

$$\|A\|_2 = \sqrt{x^T A^T A x} = \sqrt{\lambda}$$

- Also known as the spectral norm

System of Linear Equations: Condition Number

- We now come back to the question: how sensitive is the solution to small changes in [A] and/or {b}?
- Look at the worst-case scenario: upper bound of error
- Effect of change in {b}:

$$A(x+\delta x)=(b+\delta b)\Rightarrow \delta x = A^{-1}\delta b \Rightarrow \|\delta x\| \le \|A^{-1}\|.\|\delta b\|$$

$$\frac{\|\delta x\|}{\|x\|} \le \frac{\|A^{-1}\|.\|\delta b\|}{\|x\|}\frac{\|b\|}{\|b\|} \le \frac{\|A^{-1}\|.\|\delta b\|}{\|x\|}\frac{\|A\|.\|x\|}{\|b\|} \le \|A^{-1}\|.\|A\|\frac{\|\delta b\|}{\|b\|}$$

- Effect of change in [A]

$$(A+\delta A)(x+\delta x)=b \Rightarrow \delta x = -A^{-1}\delta A(x+\delta x)\Rightarrow \|\delta x\| \le \|A^{-1}\|.\|\delta A\|.\|x+\delta x\|$$

$$\frac{\|\delta x\|}{\|x+\delta x\|} \le \|A^{-1}\|.\|\delta A\|\frac{\|A\|}{\|A\|} \le \|A^{-1}\|.\|A\|\frac{\|\delta A\|}{\|A\|}$$

- Condition Number of matrix A, C(A), is, therefore, $\|A\|.\|A^{-1}\|$

Condition Number: Example

$$x_1 + \quad x_2 = 2.00$$
$$0.99x_1 + 1.01x_2 = 2.00$$

$$A = \begin{bmatrix} 1 & 1 \\ 0.99 & 1.01 \end{bmatrix} \qquad A^T A = \begin{bmatrix} 1.9801 & 1.9999 \\ 1.9999 & 2.0201 \end{bmatrix}$$

$$A^{-1} = 50\begin{bmatrix} 1.01 & -1 \\ -0.99 & 1 \end{bmatrix} \qquad \left(A^{-1}\right)^T A^{-1} = 2500\begin{bmatrix} 2.0002 & -2 \\ -2 & 2 \end{bmatrix}$$

- The Matrix norms are (How to find the *2*-norm is described later):

$$\|A\|_1 = 2.01 \qquad \|A\|_\infty = 2 \qquad \|A\|_2 = 2$$
$$\left\|A^{-1}\right\|_1 = 100 \qquad \left\|A^{-1}\right\|_\infty = 100.5 \qquad \left\|A^{-1}\right\|_2 = 100$$

Condition Number: Example

$$\begin{aligned} x_1 + \quad x_2 &= 2.00 \\ 0.99x_1 + 1.01x_2 &= 2.00 \end{aligned}$$

Solution: 1,1

$$\begin{aligned} x_1 + \quad x_2 &= 1.98 \\ 0.99x_1 + 1.01x_2 &= 2.02 \end{aligned}$$

Solution: −1.01,2.99

$$\begin{aligned} x_1 + \quad x_2 &= 2.00 \\ 1.00x_1 + 1.01x_2 &= 2.00 \end{aligned}$$

Solution: 2,0

Methods of Solution of Systems of Linear Equations

- Given $Ax = b$, how to find {x} for known A and b?
- Easiest: when A is a diagonal matrix
- A little more difficult, but still easy, if A is a triangular matrix (either upper triangular or lower triangular)
- Otherwise, we may perform some operations on the system to reduce A to one of these forms, and then solve the system directly (**Direct methods**). These will arrive at the solution in a finite number of steps.
- The other option is to start with a guess value of the solution and then use an iterative scheme to improve these (**Iterative methods**). Number of steps depends on the convergence properties of the system and the desired accuracy. May not even Converge!

Direct Methods: Gauss Jordan Method

- Reduce A to a diagonal matrix (generally Identity matrix)
- Take the first equation and divide it by a_{11} to make the diagonal element unity. Then, express x_1 as a function of other (n-1) variables:

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3 \ldots - a_{1n}x_n}{a_{11}}$$

- Using this, eliminate x_1 from all other equations. This will change the coefficients of these equations. E.g., the second equation:

$$a_{21}\frac{b_1 - a_{12}x_2 - a_{13}x_3 \ldots - a_{1n}x_n}{a_{11}} + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$

$$\left(a_{22} - a_{21}\frac{a_{12}}{a_{11}}\right)x_2 + \ldots + \left(a_{2n} - a_{21}\frac{a_{1n}}{a_{11}}\right)x_n = b_2 - a_{21}\frac{b_1}{a_{11}}$$

Direct Methods: Gauss Jordan Method

- Written compactly: $R'_1=R_1/a_{11}$; $R'_i=R_i-a_{i1}xR'_1$ i=2 to n
- After this step, the first column of [A] is the same as that of an n x n Identity matrix [I]
- Now, take the second equation and divide it by a'_{22} to make the diagonal element unity. Then, express x_2 as a function of other (n-2) variables :

$$x_2 = \frac{b_2^{'} - a_{23}^{'}x_3 - a_{24}^{'}x_4 \ldots - a_{2n}^{'}x_n}{a_{22}^{'}}$$

- Using this, eliminate x_2 from all other equations, including the first one. E.g.

$$b_3^{''} = b_3^{'} - a_{32}^{'}\frac{b_2^{'}}{a_{22}^{'}}; a_{33}^{''} = a_{33}^{'} - a_{32}^{'}\frac{a_{23}^{'}}{a_{22}^{'}}; \ldots; a_{3n}^{''} = a_{3n}^{'} - a_{32}^{'}\frac{a_{2n}^{'}}{a_{22}^{'}}$$

- Compactly: $R''_2=R'_2/a'_{22}$; $R''_i=R'_i-a'_{i2}xR'_2$ i=1, 3 to n

Direct Methods: Gauss Jordan Method

- After this step, first two columns are same as [I]. Also note that the row-modifications in this step need to be done only for columns 3 to n.
- Similarly, for the third equation, computations needed for only columns 4 to n.
- Repeat till the last equation and the modified {b} vector is the solution!
- Note that any *Pivot element* a_{ii} should not become 0. Otherwise, the order of the equations needs to be changed.
- Changing the order of equations to bring a non-zero element at the pivot position is called **pivoting**.

Gauss Jordan Method: Example

$$\begin{bmatrix} 2 & 4 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 13 \\ 14 \\ 20 \end{Bmatrix}; \quad \begin{bmatrix} 1 & 2 & 0.5 \\ 1 & 2 & 3 \\ 3 & 1 & 5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 6.5 \\ 14 \\ 20 \end{Bmatrix}; \quad \begin{bmatrix} 1 & 2 & 0.5 \\ 0 & 0 & 2.5 \\ 0 & -5 & 3.5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 6.5 \\ 7.5 \\ 0.5 \end{Bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 0.5 \\ 0 & -5 & 3.5 \\ 0 & 0 & 2.5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 6.5 \\ 0.5 \\ 7.5 \end{Bmatrix}; \begin{bmatrix} 1 & 2 & 0.5 \\ 0 & 1 & -0.7 \\ 0 & 0 & 2.5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 6.5 \\ -0.1 \\ 7.5 \end{Bmatrix}; \begin{bmatrix} 1 & 0 & 1.9 \\ 0 & 1 & -0.7 \\ 0 & 0 & 2.5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 6.7 \\ -0.1 \\ 7.5 \end{Bmatrix}$$

Pivoting

$$\begin{bmatrix} 1 & 0 & 1.9 \\ 0 & 1 & -0.7 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 6.7 \\ -0.1 \\ 3 \end{Bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \end{Bmatrix}$$

Gauss Jordan Method: Computational Effort

- The method looks simple, and is easily programmable, then why should we not use it?
- Consider the number of floating point operations (flops)
- First step requires n division, and, for each of the other $n-1$ rows, $n-1$ multiplications, and $n-1$ subtractions
- Second step requires $n-1$ division, and, for each of the other $n-1$ rows, $n-2$ multiplications, and $n-2$ subtractions
- The total flops are, then, $\sum_{i=1}^{n} i + (n-1)(2i) = \sum_{i=1}^{n} (2n-1)i \approx n^3$
- There is a more efficient method, *Gauss Elimination*, which requires 2/3 times the computational effort.

Direct Methods: Gauss Elimination Method

- Reduce A to upper-triangular matrix (forward elimination)
- Solve the triangular system (back substitution)
- Using the first equation express x_1 as a function of other (n-1) variables:

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3 \ldots - a_{1n}x_n}{a_{11}}$$

- Eliminate x_1 from all other equations. This will change the coefficients of these equations. E.g., the second equation:

$$a_{21}\frac{b_1 - a_{12}x_2 - a_{13}x_3 \ldots - a_{1n}x_n}{a_{11}} + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$

$$\left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right)x_2 + \ldots + \left(a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}\right)x_n = b_2 - \frac{a_{21}}{a_{11}}b_1$$

Gauss Elimination Method

- Written compactly: $R'_i = R_i - (a_{i1}/a_{11}) \times R_1$ $\quad$ i=2 to n
- After this step, the first column of [A] has a_{11} on the diagonal and zeroes everywhere else
- Now, take the second equation, express x_2 as a function of other (n-2) variables :

$$x_2 = \frac{b_2' - a_{23}' x_3 - a_{24}' x_4 \ldots - a_{2n}' x_n}{a_{22}'}$$

- Using this, eliminate x_2 from all other equations, <u>excluding</u> the first one. E.g.

$$b_3'' = b_3' - \frac{a_{32}'}{a_{22}'} b_2' ; a_{33}'' = a_{33}' - \frac{a_{32}'}{a_{22}'} a_{23}' ; \ldots ; a_{3n}'' = a_{3n}' - \frac{a_{32}'}{a_{22}'} a_{2n}'$$

- Compactly: $R''_i = R'_i - (a'_{i2}/a'_{22}) \times R'_2$ $\quad$ i= 3 to n

Gauss Elimination Method

- After this step, first two columns have all below-diagonal elements as zero.
- Repeat till the last equation to obtain an upper triangular matrix and a modified {b} vector.
- The last equation can now be used to obtain x_n directly.
- The second-last equation has only two "unknowns," x_{n-1} and x_n, and x_n is already computed. Solve for x_{n-1}.
- Repeat, going backwards to the first equation, to obtain the complete solution.
- The total flops are, $\sum_{i=1}^{n}\left[i+2i(i-1)\right]+\left[\approx\sum_{i=1}^{n} i\right]\approx\sum_{i=1}^{n} 2i^2\approx\frac{2}{3}n^3$
- More efficient by about 50%

Gauss Elimination Method: Example

Forward Elimination

$$\begin{bmatrix} 2 & 4 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 13 \\ 14 \\ 20 \end{Bmatrix}; \qquad \begin{bmatrix} 2 & 4 & 1 \\ 0 & 0 & 2.5 \\ 0 & -5 & 3.5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 13 \\ 7.5 \\ 0.5 \end{Bmatrix};$$

$$\begin{bmatrix} 2 & 4 & 1 \\ 0 & -5 & 3.5 \\ 0 & 0 & 2.5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 13 \\ 0.5 \\ 7.5 \end{Bmatrix};$$

Pivoting

Back substitution

$$x_3 = \frac{7.5}{2.5} = 3; x_2 = \frac{0.5 - 3.5 \times 3}{-5} = 2; x_1 = \frac{13 - 4 \times 2 - 1 \times 3}{2} = 1$$