1/28/2013

Recall

**Def** a maximization linear programming problem is in <u>canonical form</u> if it is written as.

$$\max \ z(\vec{x}) = \vec{c}^T \vec{x}$$
$$\text{subject to} \quad A\vec{x} \le \vec{b}$$
$$\vec{x} \geq 0$$

**Def** a minimization —//— — in <u>canonical form</u> if

$$\min \ z(\vec{x}) = \vec{c}^T \vec{x}$$
$$\text{subject to} \quad A\vec{x} \geq \vec{b}$$
$$\vec{x} \geq 0$$

**Def** a <u>standard form</u> of max. problem:

$$\max \ z(\vec{x}) = \vec{c}^T \vec{x}$$
$$\text{s.t.} \quad A\vec{x} = \vec{b}$$
$$\vec{x} \geq 0$$

**Claim** Every linear programming problem written in the canonical form can be written in the standard form.

E.g. $a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1$

Introduce $s_1$, $s_1 \geq 0$, called <u>slack</u> (or <u>surplus</u>) variable:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + s_1 = b_1$$

We do the same for other equations and we write these variables $s_1, s_2, \ldots, s_m$ (for $m$ equations) in a row $\vec{s}$. Then the linear problem can be written as

$$\begin{aligned} \max\ & z(\vec{x}) = \vec{c}^T \vec{x} \\ \text{s.t}\ & A\vec{x} + I_m \vec{s} = \vec{b} \\ & \vec{x} \geq 0 \\ & \vec{s} \geq 0 \end{aligned}$$

<u>Ex</u> Consider the Toy Maker problem. In the canonical form we have

$$\begin{aligned} \max\ & z(x_1, x_2) = 7x_1 + 6x_2 \\ \text{s.t.}\ & 3x_1 + x_2 \leq 120 \\ & x_1 + 2x_2 \leq 160 \\ & x_1 \leq 35 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

We introduce slack $s_1, s_2, s_3 \geq 0$ and then we can write the problem in the standard form:

$$\begin{aligned} \max\ & z(x_1, x_2) = 7x_1 + 6x_2 \\ \text{s.t.}\ & 3x_1 + x_2 + s_1 = 120 \\ & x_1 + 2x_2 + s_2 = 160 \\ & x_1 + s_3 = 35 \\ & x_1 \geq 0,\ x_2 \geq 0,\ s_1 \geq 0,\ s_2 \geq 0,\ s_3 \geq 0 \end{aligned}$$

For minimization problems, we can subtract slack variables:

row $i$ $\quad a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i$

Let $s_i \geq 0$ and

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - s_i = b_i$$

We can also go from standard to canonical.

max problem $\quad A\vec{x} = \vec{b} \iff A\vec{x} \geq \vec{b}$ and $A\vec{x} \leq \vec{b}$

$$A\vec{x} \geq \vec{b} \iff \boxed{-A\vec{x} \leq -\vec{b}}$$

Then

$$\begin{cases} \max\ z(x_1, x_2) \\ \text{s.t.}\ \ A\vec{x} \leq \vec{b} \\ \quad\ \boxed{-A\vec{x} \leq -\vec{b}} \\ \quad\ \ \vec{x} \geq 0 \end{cases}$$

<u>Note</u> The standard solution method for linear programming (LP) problem (the simplex method) requires all decision variables $x_1, x_2, \dots, x_n$ to be non-negative. Other constraints on $x_i$ can be re-written by introducing new variables s.t. new variables are non-negative.

<u>Ex</u> $x_i \le 0$ Let $y_i = -x_i \geq 0$ $x_i = -y_i$
then you would to replace $x_i$ with $-y_i$ in the objective function and all constraints

<u>Ex</u> $x_i \geq l_i$ Let $y_i = x_i - l_i \geq 0$ $x_i = y_i + l_i$
we replace $x_i$ with $y_i + l_i$ in the objective function and constraints

<u>Ex</u> $x_i \le u_i$ Let $y_i = u_i - x_i \geq 0$ $x_i = u_i - y_i$
then replace $x_i$ with $u_i - y_i$ ...

<u>Ex</u> if there is no constraint on the sign of $x_i$, then we can introduce two variables $y_i$ and $z_i$ : $x_i = y_i - z_i$, $y_i, z_i \geq 0$

## Gauss-Jordan elimination and Solution of the Linear System

<u>Def</u> Let $A$ be a $m \times n$ matrix. There are three elementary row operations on $A$.

(i) Any row can be multiplied by any non-zero scalar

(ii) Any two rows can be interchanged

(iii) Any row can be multiplied by a non-zero scalar and added to any other row.

**Ex** $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}$ : upper triangular matrix

$2^{nd}$ row $\leftarrow$ $2^{nd}$ row $- 3 \cdot [1^{st}$ row$]$

$4 - 3 \cdot 2 = 4 - 6 = -2$

$2^{nd}$ row $\leftarrow$ $(-\frac{1}{2}) \cdot [2^{nd}$ row$]$

$\sim \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$ upper triangular matrix with "1"'s on the main diagonal

$1^{st}$ row $\leftarrow$ $1^{st}$ row $- 2 \cdot [2^{nd}$ row$]$

$\sim \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2$

<u>Claim</u> Every elementary row operation is equivalent to multiplication of a given matrix by a transformation matrix.

**Ex** We would like to add row 1 multiplied by $\alpha$ to row 2.

Consider $E = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \boxed{\alpha} & 1 & \cdots & 0 \\ & & \ddots & \\ 0 & & & 1 \end{pmatrix}$ : this is identity matrix in which row 2 is replaced with row 2 + $\alpha \cdot$[row 1]

Now if we want to add $1^{st}$ row $\times\ \alpha$ to $2^{nd}$ row of matrix $A$, we can write

$$EA$$

multiplication from the left

<u>Note</u> multiplication from the right is used for elementary column operations.

<u>Ex</u> Multiply row $i$ by constant $\alpha$

$$E = \begin{array}{cl} \begin{pmatrix} 1 & & & 0 \\ & \ddots & & \\ & & \alpha & \\ & & & \ddots \\ 0 & & & & 1 \end{pmatrix} & \begin{array}{l} \\ \\ - \text{ row } i \\ \\ \\ \end{array} \end{array}$$

identity with element in row $i$ replaced with $\alpha$

$$EA$$

<u>Ex</u> Swapping rows $i$ and $j$

$$E = \begin{array}{cl} \begin{pmatrix} 1 & & & & \\ & 0 & 1 & & \\ & 1 & 0 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} & \begin{array}{l} \\ - \text{ row } i \\ \text{row } j \\ \\ \\ \end{array} \end{array}$$

$$\begin{array}{cc} \text{column} & \text{col.} \\ i & j \end{array}$$

Suppose we apply 4 elementary row operations to matrix $A$ that are equivalent to transformation matrices $E_1, E_2, E_3, E_4$. The resulting matrix is

$$B = \underbrace{E_4 E_3 E_2 E_1}\, A$$

<u>Def</u> Let $A, B$ be two matrices. If there is sequence of elementary row operations with transformation matrices $E_1, \dots, E_k$ such that

$$B = E_k \dots E_1 A$$

then matrices $A$ and $B$ are called <u>row equivalent</u>.

## <u>Matrix Inverse</u>

<u>Def</u> Let $A \in \mathbb{R}^{n \times n}$ square matrix. Matrix $A^{-1}$ is called <u>inverse</u> of matrix $A$ if

$$A A^{-1} = A^{-1} A = I$$

If $A^{-1}$ exists, then $A$ is called <u>invertible</u>.

1/30/2013

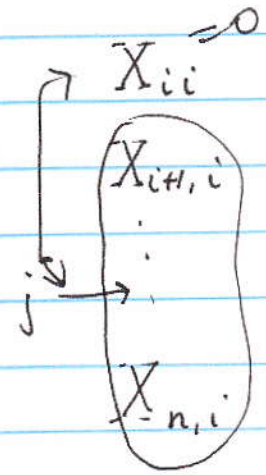
If a matrix $A$ is not invertible, it is called <u>singular</u>.

## Gauss-Jordan Elimination (Algorithm 1)
(Computing an inverse)

(1) Let $A \in \mathbb{R}^{n \times n}$. Let $X = [A \mid I_n]$
augmented matrix

(2) Let $i := 1$

(3) If $X_{ii} = 0$, then use row-swapping on $X$ to replace row $i$ with row $j$ $(j > i)$ so that $X_{ii} \neq 0$. If this is not possible, then $A$ is not invertible.




(4) Replace $X_{i\cdot}$ by

$\frac{1}{X_{ii}} X_{i\cdot}$ Element $(i,i)$ of $X$ should be 1.

(5) For each $j \neq i$, replace $X_{j\cdot}$ by

$$-\frac{X_{ji}}{\underbrace{X_{ii}}_{=1}} X_{i\cdot} + X_{j\cdot}$$

(6) Set $i := i+1$

(7) If $i > n$, then $A$ should be replaced by $I_n$ and $I_n$ has been replaced by $A^{-1}$ in $X$.
If $i \le n$, then go to step 3.

$$\begin{pmatrix} A_{11}^{(1)} & & & & \\ & A_{22}^{(2)} & & & \\ & & \ddots & & \\ & & & A_{ii}^{(i)} & \cdots \\ & & & \vdots & \end{pmatrix}$$

<u>Note</u> Step (3) is called <u>pivoting</u>.

<u>Ex</u> $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

$$\begin{aligned} 1 \cdot x_1 + 2x_2 &= 1 \\ 3 \cdot x_1 + 4x_2 &= 0 \end{aligned} \qquad \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

<u>Step 1</u>

Form $X = [A \mid I_2] = \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{array}\right]$

<u>Step 2</u>: $i := 1$

<u>Step 3</u>: eliminate variable $x_1$ from $2^{nd}$ eq$^{n}$

Replace row 2 with row 2 − 3 · row 1

$\sim \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & -2 & -3 & 1 \end{array}\right]$ <u>Step 4</u>: $i := i+1 = 2$

<u>Step 5</u>: multiply $2^{nd}$ row by $\left(-\frac{1}{2}\right)$

$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 1 & \frac{3}{2} & -\frac{1}{2} \end{array}\right]$$

<u>Step 6</u>

Eliminate variable $x_2$ from eq$^{\underline{n}}$ (1):

row 1 $\leftarrow$ row 1 $-$ 2 $\cdot$ row 2

$$\sim \left[\begin{array}{cc|cc} 1 & 0 & -2 & 1 \\ 0 & 1 & \frac{3}{2} & -\frac{1}{2} \end{array}\right] \qquad \begin{array}{l} 1 - \frac{3}{2} \cdot 2 = -2 \\ 0 - 2\left(-\frac{1}{2}\right) = 1 \end{array}$$

<u>Step 7</u>: $i := i+1 = 3 > n = 2$. This terminates the algorithm

<u>Forward Elimination</u>

$$\underbrace{\begin{pmatrix} l_{11} & & & 0 \\ l_{21} & l_{22} & & \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}}_{\text{lower triangular matrix}} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \qquad \downarrow$$

$x_1 = b_1 / l_{11}$

$l_{21} \cdot x_1 + l_{22} \cdot x_2 = b_2 \Rightarrow x_2 = (b_2 - l_{21} \cdot x_1) / l_{22}$

$$l_{n1} x_1 + l_{n2} \cdot x_2 + \dots + l_{nn} \cdot x_n = b_n$$

$$x_n = \left( b_n - [l_{n1} x_1 + l_{n2} \cdot x_2 + \dots + l_{n,n-1} x_{n-1}] \right) / l_{nn}$$

<u>Back substitution</u>

$$\underbrace{\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ 0 & & u_{n-1,n-1} & u_{n-1,n} \\ & & & u_{nn} \end{pmatrix}}_{\text{upper triangular matrix}} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}$$

$$x_n = b_n / u_{nn}$$

$$x_{n-1} = \left[ b_{n-1} - u_{n-1,n-1} x_{n-1} \right] / u_{n-1,n}$$

$$x_1 = \left( b_1 - [u_{12} x_2 + \dots + u_{1n} x_n] \right) / u_{11}$$

<u>Note</u> We could Algorithm 1 to solve a linear system $Ax = b$ (this is not usually used in practice). Then instead of $I_n$ we will write $b$. At the end when $A$ is replaced by $I_n$, vector $b$ will be replaced with $A^{-1} b$, i.e. $X = [I_n \mid A^{-1} b]$

$b = I_n \cdot b$

It may happen that algorithm 1 does not terminate with $X = [I_n \mid A^{-1} b]$. Instead we can get $X = [A' \mid b']$ with at <u>least one zero row</u>. Such $A'$ has the form

$$A' = \left(\begin{array}{ccccccc} 1 & 0 & \cdots & & & & 0 \\ 0 & 1 & \cdots & & & & 0 \\ \vdots & & \ddots & & & & \\ & & & 1 & & & 0 \\ \hline 0 & 0 & & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & \\ 0 & 0 & & 0 & 0 & \cdots & 0 \end{array}\right) \qquad b' = \left(\begin{array}{c} * \\ * \\ \vdots \\ * \\ \hline \square \\ \vdots \\ \square \end{array}\right)$$

In this case, there are two possibilities:

(1) For every zero row in $A'$, the corresponding element $\square$ in $b'$ is $0$. In this case, there are infinitely many alternative solutions.

(2) If at least one element $\square$ in $b'$ is $\neq 0$, then the system has no solutions.

<u>Ex</u> Consider the system of equations:

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 7 \\ 4x_1 + 5x_2 + 6x_3 = 8 \\ 7x_1 + 8x_2 + 9x_3 = 9 \end{cases}$$

After applying Gauss-Jordan elimination with RHS $\vec{b}^T = [7\ 8\ 9]$ we get

$$\bar{X} = \left[\begin{array}{ccc|c} 1 & 0 & -1 & -19/3 \\ 0 & 1 & 2 & 20/3 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

$0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = 0$ and we have only 2 equations $1 \cdot x_1 - x_3 = -19/3$

$$x_2 + 2x_3 = 20/3$$

to find 3 unknowns $x_1, x_2, x_3$. We have an underdetermined system.

Let $x_3 = t$: free parameter

# LU factorization (optional)

$A = LU$

$$U = A^{(n)} = \begin{pmatrix} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} \\ & \ddots & \vdots \\ 0 & & a_{nn}^{(1)} \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & m_{22} & 1 & & \\ \vdots & & \ddots & & \\ m_{n1} & m_{n2} & \cdots & & 1 \end{pmatrix}$$

$Ax = b$

1) $L\underbrace{Ux}_{y} = b$ $\qquad A = LU$

2) Solve $Ly = b$ for $y$

3) Solve $Ux = y$ for $x$

$= 0.31427 \cdot 10^1$ in floating pt representation 2/1/2013

$x = \underbrace{3.1427}$ 4 digits are accurate

$y = \underbrace{3.1425} = \underbrace{0.31425} \cdot 10^1$

$x - y = 0.0002 = \underbrace{0.20000} \cdot 10^{-3}$

no digits are correct

<u>This is loss of precision.</u>

<u>Ex</u> (Cont'd)

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 7 \\ 4x_1 + 5x_2 + 6x_3 &= 8 \\ 7x_1 + 8x_2 + 9x_3 &= 9 \end{aligned}$$

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 7 \\ 4 & 5 & 6 & 8 \\ 7 & 8 & 9 & 9 \end{array}\right] \underset{\text{Jordan elimination}}{\overset{\text{Gauss-}}{\sim}} \left[\begin{array}{ccc|c} 1 & 0 & -1 & -19/3 \\ 0 & 1 & 2 & 20/3 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

$0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = 0$ true for any $x_1, x_2, x_3$

Let $x_3 = t$ be a free parameter.

2nd eq$^{\underline{n}}$: $x_2 + 2x_3 = 20/3 \Rightarrow x_2 = \frac{20}{3} - 2x_3 = \frac{20}{3} - 2t$

1st eq$^{\underline{n}}$: $x_1 - x_3 = -19/3 \Rightarrow x_1 = -\frac{19}{3} + x_3 = -\frac{19}{3} + t$

Hence $\left(-\frac{19}{3}+t,\ \frac{20}{3}-2t,\ t\right)$, $t \in \mathbb{R}$, is a solution of $Ax=b$, so we have infinitely many solutions

<u>Ex</u> Suppose we have the system

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 7 \\ 4x_1 + 5x_2 + 6x_3 &= 8 \\ 7x_1 + 8x_2 + 9x_3 &= 10 \end{aligned} \qquad \vec{b} = \begin{bmatrix} 7 \\ 8 \\ 10 \end{bmatrix}$$

After Gauss-Jordan elimination we get

$$\tilde{X} = \left[\begin{array}{ccc|c} 1 & 0 & -1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

The $3^{rd}$ eq$^{n}$: $0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = \underline{1}$ : no solution

$\Rightarrow$ the problem doesn't have a solution.

## <u>Linear Combinations, Linear Dependence / Independence</u>

<u>Def</u> Let $\vec{x}_1, \dots, \vec{x}_m$ be vectors from $\mathbb{R}^n$. Let $\alpha_1, \dots, \alpha_m$ be some scalars. Then

$$\alpha_1 \vec{x}_1 + \alpha_2 \vec{x}_2 + \dots + \alpha_m \vec{x}_m \quad \in \mathbb{R}^n$$

is a <u>linear combination</u> of vectors $\vec{x}_1, \dots, \vec{x}_m$.

**Def** Vectors $\vec{x}_1, \dots, \vec{x}_m \in \mathbb{R}^n$ are <u>linearly dependent</u> if there exist constants $\alpha_1, \dots, \alpha_m$ not all being zero such that

$$\alpha_1 \vec{x}_1 + \dots + \alpha_m \vec{x}_m = \vec{0}$$

Otherwise, vectors $\vec{x}_1, \dots, \vec{x}_m$ are linearly independent, i.e.

$$\alpha_1 \vec{x}_1 + \dots + \alpha_m \vec{x}_m = \vec{0}$$

implies $\alpha_1 = \dots = \alpha_m = 0$.

**Ex** In $\mathbb{R}^3$, consider vectors

$$\vec{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \qquad \vec{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \qquad \vec{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

We will show that vectors are linear independent. (LI)
Form a linear combination:

$$\alpha_1 \vec{x}_1 + \alpha_2 \vec{x}_2 + \alpha_3 \vec{x}_3 = \vec{0}$$

$$\alpha_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{cases} \alpha_1 + \alpha_2 = 0 \\ \alpha_1 + \alpha_3 = 0 \\ \alpha_2 + \alpha_3 = 0 \end{cases}$$

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}}_{A} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Note that vectors $\vec{x}_1, \vec{x}_2, \vec{x}_3$ became columns of the matrix.

We use Gauss-Jordan elimination and we get

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \alpha_1 = \alpha_2 = \alpha_3 = 0$$

$$\Downarrow$$

vectors are LI

Another way to see this is to note that matrix $A$ is non-singular, so that system

$$A \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

has a unique trivial solution $\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$.

<u>Note</u> The number of vectors is not always the same as dimension of the space vectors are from. i.e. their length. We can consider 3 vectors in $\mathbb{R}^4$ space etc.

<u>Ex</u> Consider two vectors in $\mathbb{R}^3$:

$$\vec{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \vec{x}_2 = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Gauss-Jordan elimination gives

$$\left[\begin{array}{cc|c} 1 & 4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

$3^{rd}$ eq$^n$: $0 \cdot \alpha_1 + 0 \cdot \alpha_2 = 0$ ✓ true for any $\alpha_1, \alpha_2$

$2^{nd}$ eq$^n$: $\alpha_2 = 0$

$1^{st}$ eq$^n$: $\alpha_1 + 4\cancel{\alpha_2}^{\,0} = 0 \Rightarrow \alpha_1 = 0$

$\alpha_1 = \alpha_2 = 0 \Rightarrow \vec{x}_1, \vec{x}_2$ are LI

<u>Note</u> Adding a zero vector to any set of vectors makes the set linearly dependent.

<u>Claim</u> Let $\vec{x}_1, \dots, \vec{x}_m \in \mathbb{R}^n$. If $m > n$, then the vectors are linearly dependent.

## Basis

<u>Def</u> Let $\vec{x}_1, \dots, \vec{x}_m$ be vectors from $\mathbb{R}^n$. The set $\{\vec{x}_1, \dots, \vec{x}_m\}$ is a basis of $\mathbb{R}^n$ if $\{\vec{x}_1, \dots, \vec{x}_m\}$

is the set of linearly independent vectors and any vector $\vec{x} \in \mathbb{R}^n$ can be written as a linear combination of $\vec{x}_1, \dots, \vec{x}_m$, i.e. there exist constants $\alpha_1, \dots, \alpha_m$:

$$\vec{x} = \alpha_1 \vec{x}_1 + \dots + \alpha_m \vec{x}_m = \sum_{i=1}^{m} \alpha_i \vec{x}_i$$

<u>Ex</u> We can show that vectors

$$\vec{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \vec{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

form a basis in $\mathbb{R}^3$. We already showed that $\vec{x}_1, \vec{x}_2, \vec{x}_3$ are LI. We need to show that for any vector $\vec{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ we can find

$\alpha_1, \alpha_2, \alpha_3$ :

$$\vec{x} = \alpha_1 \vec{x}_1 + \alpha_2 \vec{x}_2 + \alpha_3 \vec{x}_3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$\underbrace{\left[\begin{array}{ccc|c} 1 & 1 & 0 & a \\ 1 & 0 & 1 & b \\ 0 & 1 & 1 & c \end{array}\right]}_{A} \underset{\text{elim.}}{\overset{\text{G-J.}}{\sim}} \left[\begin{array}{ccc|c} 1 & 0 & 0 & \frac{1}{2}a + \frac{1}{2}b - \frac{1}{2}c = \alpha_1 \\ 0 & 1 & 0 & -\frac{1}{2}b + \frac{1}{2}a + \frac{1}{2}c = \alpha_2 \\ 0 & 0 & 1 & \frac{1}{2}c + \frac{1}{2}b - \frac{1}{2}a = \alpha_3 \end{array}\right]$$

$$\Rightarrow \quad \alpha_1 = \frac{1}{2}a + \frac{1}{2}b - \frac{1}{2}c$$

$$\alpha_2 = \dots$$

$$\alpha_3 = \dots$$

Another way to show that $\alpha_1, \alpha_2, \alpha_3$ exist is to note that $A$ is invertible.

2/4/2013

# Gauss-Jordan elimination
## (for finding inverse)

$Ax = b$

Suppose we want to find $A^{-1}$

<u>Note</u> Typically, to solve a linear system $Ax=b$, one would use LU factorization or iterative methods.

Here we concentrate on finding $A^{-1}$, which is about 3 times more expensive (operation count) than to do LU factorization.

$$[A \mid I] \rightsquigarrow [I \mid A^{-1}]$$

$X = [A \mid I]$ : augmented matrix

$A \in \mathbb{R}^{n \times n} \Rightarrow X \in \mathbb{R}^{n \times 2n}$

```
for k = 1 : n-1
    for i = 1 : n
        X_{k,1:2n} = X_{k,1:2n} / X_{kk}        % we
                                                % assume
                                                % X_kk ≠ 0
        if i ≠ k
            for j = k+1 : 2n
                X_ij = X_ij - X_kj · X_ik

X_{n,1:2n} = X_{n,1:2n} / X_nn
```

## Partial pivoting:

$$\begin{pmatrix} a_{11}^{(1)} & \cdots & \cdots & \cdots \\ & a_{22}^{(2)} & \cdots & \cdots \\ & & a_{kk}^{(k)} & \cdots \\ 0 & & \vdots & \vdots \\ & & a_{nk}^{(k)} & \cdots \; a_{nn}^{(k)} \end{pmatrix}$$

if $a_{kk}^{(k)} \neq 0$, then just pivoting is to find the first row $i$ $k+1 \leq i \leq n$ in which $a_{ik}^{(k)} \neq 0$, and then switch rows $i$ and $k$

Partial pivoting: find row $i$, $k+1 \leq i \leq n$, such that $|a_{ik}|$ is the largest among $\{|a_{k+1,k}|, |a_{k+2,k}|, \ldots \ldots |a_{nk}|\}$, then switch rows $i$ and $k$ and proceed.

To correct: for partial pivoting, you choose row $i$: $k \leq i \leq n$, even if $a_{kk}^{(k)} \neq 0$.

## Partial pivoting:

$[y, p] = \max(X_{k:n,k})$

$X_{temp} = X_{k,1:2n}$ % temp. vector

$X_{k,1:2n} = X_{k-1+p,1:2n}$

shifting index

$X_{k-1+p,1:2n} = X_{temp}$

Ex $X = [1 \; 2 \; \underbrace{3 \; 4 \; 5}_{3^{rd}}]$

max $X[3:5] = [5, 3]$

index

true index is $(\underset{k}{3-1}) + \underset{p}{3}$

## Back to basis, linear independence.

**Result** Let $\{x_1, x_2, \dots, x_m\}$ be a basis of $\mathbb{R}^m$ and $x_{m+1}$ is another, non-zero vector from $\mathbb{R}^m$. We can write $x_{m+1}$ as a linear combination of $x_1, x_2, \dots, x_m$:

$$x_{m+1} = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_m x_m$$

We can swap $x_{m+1}$ with any vector $x_j$, $j=1,\dots,m$ $x_1, x_2, \dots, x_m$ and still have a basis if the corresponding coefficient $\alpha_j \neq 0$, i.e. $x_1, x_2, \dots, x_{j-1}, x_{m+1}, x_{j+1}, \dots, x_m$ is another basis.

**Claim:** In $\mathbb{R}^n$ every set of $n$ linearly independent vectors is a basis.

## Rank

**Def** Let $A \in \mathbb{R}^{m \times n}$. The row rank of $A$ is the size of the largest set of vectors (rows) from $A$ that are linearly independent.

**Def** Column rank is defined in a similar way.

**Claim** If $A \in \mathbb{R}^{m \times n}$, then its row rank equals its column rank. In addition, $\text{rank}(A) \leq \min(m, n)$

<u>Claim</u>. The elementary row operations do not change the row rank.

<u>Claim</u> If $A \in \mathbb{R}^{m \times m}$, i.e. $A$ is a square matrix, and $\text{rank}(A) = m$, then matrix $A$ is invertible.

<u>Def</u> For $A \in \mathbb{R}^{m \times n}$, if $\text{rank}(A) = \min(m, n)$, then we say that matrix $A$ has <u>full rank</u>. Otherwise, $A$ is <u>rank-deficient</u>.

<u>Ex</u>

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \underset{\text{elim}}{\overset{\text{G.-J.}}{\sim}} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

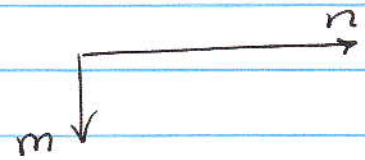
$\text{rank}(A) = 2$, rows 1 and 2 are lin. independent

2/6/2013

Matlab Optimization Toolbox is installed in JEB computer lab: JEB 331

## Solving Systems with More Variables than Equations

Let $A \in \mathbb{R}^{m \times n}$, $m \leq n$




Assume that $A$ has full rank $m$. Then the system

$$Ax = b$$

is undetermined and there are infinitely many solutions. We will discuss how to express these solutions.

Since $A$ has rank $m \Rightarrow A$ has $n$ linearly independent columns $A_{\cdot i_1}, \ldots, A_{\cdot i_m}$ that correspond to components $x_{i_1}, \ldots, x_{i_m}$ of the solution

Form matrix $B = [A_{\cdot i_1}, \ldots, A_{\cdot i_m}]$. This matrix is invertible since it consists of linearly independent columns $A_{\cdot i_1}, \ldots, A_{\cdot i_m}$.

We can use elementary <u>column</u> operations to write

$$A = [B \mid N]$$

where $N$ has the rest $n-m$ columns of $A$.
Similarly we can sub-divide solution vector

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$$

where $x_B = [x_{i_1}, \dots, x_{i_m}]^T$ and $x_N$ has the rest of components that correspond to matrix $N$.

$$Ax = b \quad \Rightarrow \quad [B \mid N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b$$

(1) or $\quad B x_B + N x_N = b$

<u>Def</u> Components in vector $x_B$ are called <u>basic variables</u> and components of vector $x_N$: <u>non-basic variables</u>.

$B$ is invertible $\Rightarrow B^{-1}$ exists
We can multiply both sides of (1) by $B^{-1}$ to get

$$\boxed{x_B = B^{-1} b - B^{-1} N x_N}$$

We expressed basic variables $x_B$ in terms of non-basic variables $x_N$ and by varying $x_N$ we can get all possible solutions ($\infty$ many).

<u>Def</u> Solution $x_B$ when $x_N = 0$ is called the <u>basic solution</u>. It is $\boxed{x_B = B^{-1} b}$

Ex $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$

Let $x_3$ be a non-basic variable. Then

$$B = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$$

Then

$$x_B = B^{-1} b = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} -19/3 \\ 20/3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}^{-1} = \frac{1}{5-8} \begin{bmatrix} 5 & -2 \\ -4 & 1 \end{bmatrix} = -\frac{1}{3} \begin{bmatrix} 5 & -2 \\ -4 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -5/3 & 2/3 \\ 4/3 & -1/3 \end{bmatrix}$$

$$\begin{bmatrix} -5/3 & 2/3 \\ 4/3 & -1/3 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} -19/3 \\ 20/3 \end{bmatrix}$$

Other two possibilities are to use columns 1 and 3, or 2 and 3 in matrix B.

## Linear programming programs in Matlab

Matlab assumes LP problems in the form:

$$\begin{cases} \min \ z(x) = c^T x \\ \text{s.t.} \quad Ax \le b \\ \qquad\quad Hx = r \\ \qquad\quad x \geq l \\ \qquad\quad x \le u \end{cases}$$

Here $x \in \mathbb{R}^{n\times 1}$, $c \in \mathbb{R}^{n\times 1}$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^{m\times 1}$
$H \in \mathbb{R}^{p\times n}$, $r \in \mathbb{R}^{p\times 1}$, $l, u \in \mathbb{R}^{n\times 1}$

$l, u$: lower and upper bounds on $x$

Matlab program to solve LP problem is linprog
It has parameters:

(1) $c$
(2) $A$
(3) $b$
(4) $H$
(5) $r$
(6) $l$
(7) $u$

If some constraints are missing, then you need to specify empty matrices, eg. $A=[]$, $b=[]$
If there is no lower bound, then $l$ can be set $-\infty$, ie. -inf. Similarly, if upper bound $u$ is not specified, you set $u$ to inf.

Ex Suppose we wish to design a diet consisting of Raman noodles and ice cream. We're interested in spending as little money as possible but we want to ensure that we eat at least 1200 calories per day and that we get at least 20 grams of protein per day. Assume that each serving of noodles costs \$1 and contains 100 calories and 2 grams of protein. Assume that each serving of ice cream costs \$1.5 and contains 200 calories and 3 grams of protein.

Let $x_1$, $x_2$ be the amount / # of servings of noodles and ice cream, respectively.

We want to minimize cost:

$$z(x_1, x_2) = x_1 + 1.5x_2$$

Protein requirement: $2x_1 + 3x_2 \geq 20$

Calorie requirement: $x_1 + 2x_2 \geq 12$
(calories in 100's)

This leads to the LP problem:

$$\begin{cases} \min & x_1 + 1.5x_2 \\ \text{s.t.} & 2x_1 + 3x_2 \geq 20 \\ & x_1 + 2x_2 \geq 12 \\ & x_1, x_2 \geq 0 \end{cases}$$

Before we solve with Matlab, let's write problem in the standard form.

<u>Recall</u> <u>canonical max problem:</u>

$$\begin{aligned} \max \quad & z(x) = c^T x \\ \text{s.t.} \quad & Ax \le b \\ & x \geq 0 \end{aligned}$$

<u>Canonical min problem:</u>

$$\begin{aligned} \min \quad & z(x) = c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

<u>Standard form</u>

$$\begin{aligned} \max \quad & z(x) = c^T x \\ \text{or } \min & \\ \text{s.t} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

---

We introduce <u>surplus variables</u> $s_1, s_2 \geq 0$, so our problem becomes

$$\left\{ \begin{aligned} \min \quad & z(x_1, x_2) = x_1 + 1.5 x_2 \\ \text{s.t.} \quad & 2x_1 + 3x_2 - s_1 = 20 \\ & x_1 + 2x_2 - s_2 = 12 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{aligned} \right.$$

This gives two equations for 4 unknowns $x_1, x_2, s_1, s_2$:

$$2x_1 + 3x_2 - s_1 = 20$$
$$x_1 + 2x_2 - s_2 = 12$$

This is an undetermined system.

$$A = \begin{bmatrix} 2 & 3 & -1 & 0 \\ 1 & 2 & 0 & -1 \end{bmatrix} \qquad b = \begin{bmatrix} 20 \\ 12 \end{bmatrix}$$

Our decision vector:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \end{bmatrix}$$

We will use Gauss-Jordan elimination and will need to decidl which variables will be basic and which non-basic.

Suppose $x_B = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$.