# IE170: Algorithms in Systems Engineering: Lecture 27

Jeff Linderoth

Department of Industrial and Systems Engineering
Lehigh University

April 9, 2007



## Taking Stock

**Last Time**

- Easy Quiz

**This Time**

- Numerical Linear Algebra
- Matrix representations



## Vectors and Matrices

- Vectors and matrices are constructs that arise naturally in many applications.
- Operating on vectors and matrices requires numerical algorithms.
- An $m \times n$ matrix is an array of $mn$ real numbers:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- $A$ is said to have $n$ columns and $m$ rows.



## More Stuff You Know

- An $n$-dimensional column vector is a matrix with one column.
- An $n$-dimensional row vector is a matrix with one row .
- By default, a vector will be considered a column vector
- The set of all $n$-dimensional vectors will be denoted $\mathbb{R}^n$.
- The set of all $m \times n$ matrices will be denoted $\mathbb{R}^{m \times n}$.

## Matrices

- The transpose of a matrix $A$ is

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

- If $x, y \in \mathbb{R}^n$, then $x^T y = \sum_{i=1}^n x_i y_i$.
- This is called the inner product of $x$ and $y$.
- If $A \in \mathbb{R}^{m \times n}$, then $A_j$ is the $j^{th}$ column, and $a_j$ is the $j^{th}$ row.
- If $A \in \mathbb{R}^{m \times k}, B \in \mathbb{R}^{k \times n}$, then $[AB]_{ij} = a_i^T B_j$.
- That is, you find the $i, j$ element of the matrix $AB$, by taking the inner product of the $i^{\text{th}}$ row of $A$ with the $j^{\text{th}}$ column of $B$.



## Sparse and Dense

- The density of a matrix is the percentage of entries that are nonzero.
- Dense vectors can simply be stored in an array.
- Dense matrices can be stored in a 2-dimensional array.
- (Here I will show you a bit of code...)
- Matrices that arise in practice, however, are typically sparse.
- For example, in linear programming, it is rare to find a practial instance that has more than 10 nonzeros/column, even though they may have tens of thousands of rows



## Sparse Matrix Storage

- Sparse matrices can be stored using a variety of different strategies
- We will learn three

**Three Sparse Matrix Formats**

- Yale Sparse Matrix Format (Triples)
- (Compressed) Sparse Column Format
- (Compressed) Sparse Row Format



## Yale Sparse Matrix Format

- Stores the matrix in three arrays:
  - A: `double` array holding the element values
  - IA: `int` array holding the row indices of the non-zero values
  - JA: `int` array holding the column indices of the non-zero values
- The arrays A, IA, and JA all have length equal to the number of non-zero elements in the matrix
- Is this the "sparsest" way to hold a matrix?
- Does it support efficient operations that we would like to do (like inner product of two columns?)

## Compressed Sparse Column Format

- Again stores the matrix in three arrays, but the arrays here have different meanings:
  - matval: `double` array holding the element values
  - matind: `int` array holding the row indices of the nonzero entries in each column.
  - matbeg: `int` array holding the location (index into) the `matval` and `matind` arrays for the first element of each column
- matval and matind: each have length equal to number of non-zeros
- matbeg: has length one more than the number of columns in the matrix



## Compressed Sparse Row Format

- Like Compressed Column Format, except "row-wise"
  - matval: `double` array holding the element values
  - matind: `int` array holding the columns indices of the nonzero entries in each column.
  - matbeg: `int` array holding the location (index into) the `matval` and `matind` arrays for the first element of each row
- matval and matind: each have length equal to number of non-zeros
- matbeg: has length one more than the number of rows in the matrix



## Taking Stock

**IE170**

- Yet To Grade: Problem Sets 8,9,10
- Roughly 60% of grade accounted for. Final: 30%

**IE171**

- Yet To Grade: Lab 8 (Spanning Tree), Lab 9 (TSP)
- Roughly 50% of the grade accounted for. Quiz: 20%

- Questions on Quiz #2?



## Score Distributions

**Quiz 2**

| | |
|---|---|
| $\geq 110$ | 1 |
| $[100, 110)$ | 1 |
| $[95, 100)$ | 1 |
| $[90, 95)$ | 1 |
| $[85, 90)$ | 1 |
| $[80, 85)$ | 2 |
| $[75, 80)$ | 1 |
| $[70, 75)$ | 1 |
| $[65, 70)$ | 2 |
| $[60, 65)$ | 0 |
| $< 60$ | 1 |

**IE170**

| | |
|---|---|
| $\geq 80$ | 4 |
| $[75, 80)$ | 1 |
| $[70, 75)$ | 2 |
| $[65, 70)$ | 2 |
| $[60, 65)$ | 1 |
| $[50, 60)$ | 1 |
| $< 50$ | 1 |

**IE171**

| | |
|---|---|
| $\geq 95$ | 1 |
| $[90, 95)$ | 4 |
| $[85, 90)$ | 1 |
| $[80, 85)$ | 1 |
| $[75, 80)$ | 0 |
| $[70, 75)$ | 0 |
| $[65, 70)$ | 2 |
| $[60, 65)$ | 1 |
| $[55, 60)$ | 1 |
| $< 55$ | 1 |