Computational Methods
CMSC/AMSC/MAPL 460

Vectors, Matrices, Linear Systems, LU Decomposition,

Ramani Duraiswami,
Dept. of Computer Science

Class Outline

- Much of scientific computation involves solution of linear equations
  - Even non-linear problems are solved by linearization
- Matrices and vectors
- Matrix vector multiplication and complexity
- Identity, Inverse, Singular Matrices
- Permutation, Lower and Upper Triangular Matrices

Vectors

- Ordered set of numbers: (1,2,3,4)
- Example: (*x,y,z*) coordinates of a point in space.
- Line joining the origin of coordinates to the point
- Vectors usually indicated with bold lower case letters. Scalars with lower case
- Operations with vectors:
  - Addition operation $\mathbf{u} + \mathbf{v}$, with:
    - Identity $\mathbf{0}$ $\quad \mathbf{v} + \mathbf{0} = \mathbf{v}$
    - Inverse - $\quad \mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
  - Scalar multiplication:
    - Distributive rule: $\alpha(\mathbf{u} + \mathbf{v}) = \alpha(\mathbf{u}) + \alpha(\mathbf{v})$

      $(\alpha + \beta)\mathbf{u} = \alpha\mathbf{u} + \beta\mathbf{u}$

Vector Spaces

- A *linear combination* of vectors results in a new vector:

$$\mathbf{v} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_n\mathbf{v}_n$$

- If the only set of scalars such that

$$\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_n\mathbf{v}_n = \mathbf{0}$$

  is $\quad \alpha_1 = \alpha_2 = \ldots = \alpha_n = \mathbf{0}$

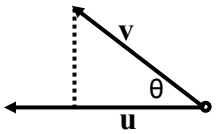
  then we say the vectors $\{\mathbf{v}_i\}$ are *linearly independent*
- The *dimension* of a space is the greatest number of linearly independent vectors possible in a vector set
- For a vector space of dimension *n*, any set of *n* linearly independent vectors form a *basis*

Dot Product

- The *dot product* or, more generally, *inner product* of two vectors is a scalar:

$$\mathbf{v}_1 \bullet \mathbf{v}_2 = x_1x_2 + y_1y_2 + z_1z_2 \qquad \text{(in 3D)}$$

- Useful for many purposes
  - Computing the length of a vector: $length(\mathbf{v}) = \text{sqrt}(\mathbf{v} \bullet \mathbf{v})$
  - *Normalizing* a vector, making it unit-length
  - Computing the angle between two vectors:
    $\mathbf{u} \bullet \mathbf{v} = |\mathbf{u}|\ |\mathbf{v}| \cos(\theta)$
  - Checking two vectors for orthogonality
  - *Projecting* one vector onto another



Linear Transformations: Matrices

- A *linear transformation*:
  - Maps one vector to another
  - Preserves linear combinations
- Thus behavior of linear transformation is completely determined by what it does to a basis
- Turns out any linear transform can be represented by a *matrix*
- A $M \times N$ matrix takes a vector with N elements to a vector with M elements

Matrix Transformations

- A *sequence* or *composition* of linear transformations corresponds to the product of the corresponding matrices
  - Note: the matrices to the *right* affect vector first
  - Note: order of matrices matters!
- The *identity matrix* **I** has no effect in multiplication
- Some (not all) matrices have an inverse:

$$\boxed{\mathbf{M}^{-1}(\mathbf{M}(\mathbf{v})) = \mathbf{v}}$$

Some special matrices

Example: A **tridiagonal** matrix

$$T = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 5 & 2 & 7 & 0 \\ 0 & 9 & 4 & 8 \\ 0 & 0 & 6 & 6 \end{bmatrix}$$

can be defined by

```
T = diag([1 2 4 6]) + diag([5 9 6],-1) + diag([3 7 8], 1);
```

- Matrices may be built up from “blocks” of smaller matrices

Some special matrices

Example: A **Vandermonde** matrix A is defined by a vector of elements $x_1, \ldots, x_n$. Its first column is all ones. Each later column is the preceding one times this vector.

- Matlab code
- How many operations and memory does this take?
- Vectorized operations

```
n = length(x);
V(:,1) = ones(n,1);
for j=2:n,
   V(:,j) = x.*V(:,j-1);
end
```

- Matrix may be sparse, i.e. most elements are zero.
- How many operations/memory?
- Answer still N^2 unless we avoid referring to the zero elements altogether

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}$$

```
D = diag([1 2 4 6]);
```

Matrix-Matrix product

- Cost of Matrix Matrix product?
  - N Matrix vector products … so O(N^3)
  - But there is also Strassen's algorithm --- mostly a curiosity
- Order of operations when doing matrix vector computations

ABx = (AB)x or A(Bx)