Math 304 (Spring 2010) - Lecture 2

Emre Mengi
Department of Mathematics
Koç University

emengi@ku.edu.tr

Floating Point Operation Count

- Efficiency of an algorithm is determined by the total $\#$ of $\oplus, \otimes, \ominus, \oslash$ required.

Floating Point Operation Count

- Efficiency of an algorithm is determined by the total $\#$ of $\oplus, \otimes, \ominus, \oslash$ required.
- Crudeness in floating point operation (flop) count

Floating Point Operation Count

- Efficiency of an algorithm is determined by the total $\#$ of $\oplus, \otimes, \ominus, \oslash$ required.
- Crudeness in floating point operation (flop) count
  - Time required for data transfers is ignored.

Floating Point Operation Count

- Efficiency of an algorithm is determined by the total $\#$ of $\oplus, \otimes, \ominus, \oslash$ required.
- Crudeness in floating point operation (flop) count
  - Time required for data transfers is ignored.
  - All of the operations $\oplus, \otimes, \ominus, \oslash$ are considered of same computational difficulty. In reality $\otimes, \oslash$ are more expensive.

Floating Point Operation Count

- Inner (or dot) product : Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ be defined as

$$f(x) = a_1 x_1 + a_2 x_2 + \ldots a_n x_n = a^T x$$

where $a = \begin{bmatrix} a_1 & \ldots & a_n \end{bmatrix}^T \in \mathbf{R}^n$ and $x = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}^T \in \mathbf{R}^n$.

Floating Point Operation Count

- Inner (or dot) product : Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ be defined as

$$f(x) = a_1 x_1 + a_2 x_2 + \ldots a_n x_n = a^T x$$

where $a = \begin{bmatrix} a_1 & \ldots & a_n \end{bmatrix}^T \in \mathbf{R}^n$ and $x = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}^T \in \mathbf{R}^n$.

  - Pseudocode to compute $f(x)$

$f \leftarrow 0$

for $j = 1, n$ **do**

$$\underbrace{f \leftarrow f + a_j x_j}_{2\ flops}$$

end for

Return f

Floating Point Operation Count

- Inner (or dot) product : Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ be defined as

$$f(x) = a_1x_1 + a_2x_2 + \ldots a_nx_n = a^Tx$$

where $a = \begin{bmatrix} a_1 & \ldots & a_n \end{bmatrix}^T \in \mathbf{R}^n$ and $x = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}^T \in \mathbf{R}^n$.

  - Pseudocode to compute $f(x)$

    $f \leftarrow 0$

    for $j = 1, n$ **do**

    $\underbrace{f \leftarrow f + a_jx_j}_{2\ flops}$

    end for

    Return f

  - Total flop count : 2 flops per iteration for $j = 1, \ldots, n$

$$Total\ \#\ of\ flops\ = \sum_{j=1}^{n} 2 = 2n$$

Floating Point Operation Count

- Matrix-vector product : Let g : $\mathbf{R}^n \rightarrow \mathbf{R}^m$ be defined as

$$g(x) = Ax = x_1 A_1 + x_2 A_2 + \cdots + x_n A_n$$

where $A = \begin{bmatrix} A_1 & \ldots & A_n \end{bmatrix}^T$ is an $m \times n$ real matrix with $A_1, \ldots, A_n \in \mathbf{R}^m$ and $x = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}^T \in \mathbf{R}^n$.

Floating Point Operation Count

- Matrix-vector product : Let g : $\mathbf{R}^n \rightarrow \mathbf{R}^m$ be defined as

$$g(x) = Ax = x_1 A_1 + x_2 A_2 + \cdots + x_n A_n$$

where $A = \begin{bmatrix} A_1 & \ldots & A_n \end{bmatrix}^T$ is an $m \times n$ real matrix with $A_1, \ldots, A_n \in \mathbf{R}^m$ and $x = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}^T \in \mathbf{R}^n$.

e.g.

$$\begin{bmatrix} 2 & 1 & -2 \\ 1 & 0 & -1 \\ 3 & -1 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix} = 2 \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} - 2 \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + 1 \begin{bmatrix} -2 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 10 \end{bmatrix}$$

Floating Point Operation Count

- Pseudocode to compute $g(x) = Ax$

  Given an $m \times n$ real matrix A and $x \in \mathbf{R}^n$.

  $g \leftarrow 0$ (where $g \in \mathbf{R}^n$)

  for $j = 1, n$ **do**

  $$\underbrace{g \leftarrow g + x_j A_j}_{2m\ flops}$$

  end for

  Return g

Floating Point Operation Count

- Pseudocode to compute $g(x) = Ax$

  Given an $m \times n$ real matrix A and $x \in \mathbf{R}^n$.

  $g \leftarrow 0$ (where $g \in \mathbf{R}^n$)

  for $j = 1, n$ **do**

  $$\underbrace{g \leftarrow g + x_j A_j}_{2m\ flops}$$

  end for

  Return g

- Above $g + x_j A_j$ requires m addition and m multiplication for each j.

Floating Point Operation Count

- Pseudocode to compute $g(x) = Ax$

  Given an $m \times n$ real matrix A and $x \in \mathbf{R}^n$.

  $g \leftarrow 0$ (where $g \in \mathbf{R}^n$)

  for $j = 1, n$ **do**

  $$\underbrace{g \leftarrow g + x_j A_j}_{2m\ flops}$$

  end for

  Return g

- Above $g + x_j A_j$ requires m addition and m multiplication for each j.

- Total flop count : $2m$ flops per iteration for $j = 1, \ldots, n$

$$Total\ \#\ of\ flops\ = \sum_{j=1}^{n} 2m = 2mn$$

Floating Point Operation Count

- Inner product view of the matrix-vector product $g(x) = Ax$.

$$g(x) = \begin{bmatrix} \bar{A}_1 x \\ \bar{A}_2 x \\ \vdots \\ \bar{A}_m x \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{nn}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix} \text{ where } A = \begin{bmatrix} \bar{A}_1 \\ \bar{A}_2 \\ \vdots \\ \bar{A}_m \end{bmatrix}$$

and $\bar{A}_1, \ldots, \bar{A}_m$ are the rows of A and a_{ij} is the entry of A at the ith row and jth column.

Floating Point Operation Count

- Inner product view of the matrix-vector product $g(x) = Ax$.

$$g(x) = \begin{bmatrix} \bar{A}_1 x \\ \bar{A}_2 x \\ \vdots \\ \bar{A}_m x \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{nn}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix} \text{ where } A = \begin{bmatrix} \bar{A}_1 \\ \bar{A}_2 \\ \vdots \\ \bar{A}_m \end{bmatrix}$$

and $\bar{A}_1, \ldots, \bar{A}_m$ are the rows of A and a_{ij} is the entry of A at the ith row and jth column.

e.g.

$$\begin{bmatrix} 2 & 1 & -2 \\ 1 & 0 & -1 \\ 3 & -1 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} (2)(2) + (1)(-2) + (-2)(1) \\ (1)(2) + (0)(-2) + (-1)(1) \\ (3)(2) + (-1)(-2) + (2)(1) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 10 \end{bmatrix}$$

Floating Point Operation Count

- Pseudocode to compute $g(x) = Ax$ exploiting the inner-product view

  Given an $m \times n$ real matrix A and $x \in \mathbf{R}^n$.

  $g \leftarrow 0$ (where $g \in \mathbf{R}^n$)

  for $i = 1, m$ **do**

      for $j = 1, n$ **do**

  $$\underbrace{g_i \leftarrow g_i + a_{ij} x_j}_{2\ flops}$$

      end for

  end for

  Return g

Floating Point Operation Count

- Pseudocode to compute $g(x) = Ax$ exploiting the inner-product view

  Given an $m \times n$ real matrix A and $x \in$ **R**^{n}.

  $g \leftarrow 0$ (where $g \in$ **R**^{n})

  for $i = 1, m$ **do**

      for $j = 1, n$ **do**

  $$\underbrace{g_i \leftarrow g_i + a_{ij}x_j}_{2\ flops}$$

      end for

  end for

  Return g

- Total flop count : 2 flops per iteration for each $j = 1, \ldots, n$ and $i = 1, \ldots, m$

$$Total\ \#\ of\ flops\ = \sum_{i=1}^{m} \sum_{j=1}^{n} 2 = \sum_{i=1}^{m} 2n = 2mn$$

Floating Point Operation Count

- Matrix-matrix product : Given an $n \times p$ matrix A and a $p \times m$ matrix X. The product $B = AX$ is an $n \times m$ matrix and defined as

$$b_{ij} = \bar{A}_i X_j = \sum_{k=1}^{p} a_{ik} x_{kj}$$

where $\bar{A}_i$ is the ith row of A, X_j is the jth column of X and b_{ij}, a_{ij}, x_{ij} denote the (i, j)-entry of B, A and X,respectively.

Floating Point Operation Count

- Matrix-matrix product : Given an $n \times p$ matrix A and a $p \times m$ matrix X. The product $B = AX$ is an $n \times m$ matrix and defined as

$$b_{ij} = \bar{A}_i X_j = \sum_{k=1}^{p} a_{ik} x_{kj}$$

where $\bar{A}_i$ is the ith row of A, X_j is the jth column of X and b_{ij}, a_{ij}, x_{ij} denote the (i, j)-entry of B, A and X,respectively.

e.g.

$$\begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} 2(-1) + 1(1) & 2(1) + 1(-2) \\ 1(-1) + 0(1) & 1(1) + 0(-2) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix}$$

Floating Point Operation Count

- Pseudocode to compute the product $B = AX$

  Given $n \times p$ and $p \times m$ matrices A and X.

  $B \leftarrow 0$

  for $i = 1, n$ **do**

      for $j = 1, m$ **do**

          for $k = 1, p$ **do**

  $$\underbrace{b_{ij} \leftarrow b_{ij} + a_{ik}x_{kj}}_{2\ flops}$$

          end for

      end for

  end for

  Return g

Floating Point Operation Count

- Pseudocode to compute the product $B = AX$

  Given $n \times p$ and $p \times m$ matrices A and X.

  $B \leftarrow 0$

  for $i = 1, n$ **do**

  for $j = 1, m$ **do**

  for $k = 1, p$ **do**

  $$\underbrace{b_{ij} \leftarrow b_{ij} + a_{ik}x_{kj}}_{2\ flops}$$

  end for

  end for

  end for

  Return g

- Total flop count : 2 flops per iteration for each $k = 1, \ldots, p$, $j = 1, \ldots, m$ and $i = 1, \ldots, n$

$$Total\ \#\ of\ flops\ = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} 2 = 2nmp$$

Floating Point Operation Count

- Big-O notation

Floating Point Operation Count

- Big-O notation
  - The inner product $a^T x$ requires $2n = O(n)$ flops (linear $\#$ of flops).

Floating Point Operation Count

- Big-O notation
  - The inner product $a^T x$ requires $2n = O(n)$ flops (linear $\#$ of flops).
  - The matrix-vector product Ax for a square matrix A (with $m = n$) requires $2n^2 = O(n^2)$ flops (quadratic $\#$ of flops).

Floating Point Operation Count

- Big-O notation
  - The inner product $a^T x$ requires $2n = O(n)$ flops (linear # of flops).
  - The matrix-vector product Ax for a square matrix A (with $m = n$) requires $2n^2 = O(n^2)$ flops (quadratic # of flops).
  - The matrix-matrix product AX for square $n \times n$ matrices A and X (with $m = n = p$) requires $2n^3 = O(n^3)$ flops (cubic # of flops).

Floating Point Operation Count

- Big-O notation
  - The inner product $a^T x$ requires $2n = O(n)$ flops (linear $\#$ of flops).
  - The matrix-vector product Ax for a square matrix A (with $m = n$) requires $2n^2 = O(n^2)$ flops (quadratic $\#$ of flops).
  - The matrix-matrix product AX for square $n \times n$ matrices A and X (with $m = n = p$) requires $2n^3 = O(n^3)$ flops (cubic $\#$ of flops).

- The notation $g(n) = O(f(n))$ means asymptotically $f(n)$ scaled up to a constant grows at least as fast as $g(n)$, *i.e.*

$g(n) = O(f(n))$ if there exists an n_0 and c such that

$$g(n) \leq cf(n) \text{ for all } n \geq n_0$$

Floating Point Operation Count

- Big-O notation
  - The inner product $a^T x$ requires $2n = O(n)$ flops (linear $\#$ of flops).
  - The matrix-vector product Ax for a square matrix A (with $m = n$) requires $2n^2 = O(n^2)$ flops (quadratic $\#$ of flops).
  - The matrix-matrix product AX for square $n \times n$ matrices A and X (with $m = n = p$) requires $2n^3 = O(n^3)$ flops (cubic $\#$ of flops).

- The notation $g(n) = O(f(n))$ means asymptotically $f(n)$ scaled up to a constant grows at least as fast as $g(n)$, *i.e.*

$g(n) = O(f(n))$ if there exists an n_0 and c such that

$$g(n) \leq cf(n) \text{ for all } n \geq n_0$$

Examples:
$2n = O(n)$ as well as $2n = O(n^2)$ and $2n = O(n^3)$

Floating Point Operation Count

- Big-O notation
  - The inner product $a^T x$ requires $2n = O(n)$ flops (linear # of flops).
  - The matrix-vector product Ax for a square matrix A (with $m = n$) requires $2n^2 = O(n^2)$ flops (quadratic # of flops).
  - The matrix-matrix product AX for square $n \times n$ matrices A and X (with $m = n = p$) requires $2n^3 = O(n^3)$ flops (cubic # of flops).

- The notation $g(n) = O(f(n))$ means asymptotically $f(n)$ scaled up to a constant grows at least as fast as $g(n)$, *i.e.*

> $g(n) = O(f(n))$ if there exists an n_0 and c such that
> $$g(n) \leq cf(n) \text{ for all } n \geq n_0$$

Examples:

$2n = O(n)$ as well as $2n = O(n^2)$ and $2n = O(n^3)$

$2n^2 = O(n^2)$ as well as $2n^2 = O(n^3)$, but $2n^2$ is not $O(n)$.