Preliminaries

Notations, conventions and assumptions

- $\mathbb{R}, \mathbb{C} \to$ sets of real and complex numbers respectively.
- $\mathbb{R}^n, \mathbb{C}^n \to \to$ set of real and complex vectors of length n respectively.
- $\mathbb{R}^{n\times m}, \mathbb{C}^{n\times m} \to$ set of real and complex $n \times m$ matrices respectively.
- $A, B, X, W, \ldots \to$ matrices.
- $x, y, a, b, \ldots \to$ vectors or scalars depending on the context.
- All vectors are column vectors.
- All matrices and vectors are assumed real unless otherwise stated.

Basic concepts of linear algebra and matrices will be assumed to be known.

Flop count

Every algorithm will be followed by a *flop count*. It is a count of the number of elementary arithmetic operations carried out by the algorithm.

A single flop will be an addition, subtraction, multiplication, division or comparison between two floating point numbers.

The flop count of an algorithm depends on the size of the input data.

It is indicative of the time complexity of the algorithm which measures how the time taken to run the algorithm increases with growth in the size of the input data.

Flop count

For $a, b, c \in \mathbb{R}$, $x, y \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^{m \times p}$,

Operation	Flop count
$ab + c$	2
ax	n
$ax + by$	$3n$
$x^T y (= \sum_{i=1}^n x_i y_i)$	$2n$
xz^T	nm
Az	$2mn$
AB	$2mnp$

If $p = m$, flop count of $AB + xz^T$ is $2nm^2 + 2nm$.

If $p = m = n$, flop count of $AB + (x^T y)(xz^T)$ is $2n^3 + 3n^2 + 2n$.

The Big-O notation in flop count

Operation	Flop count	The Big-O notation
$ab + c$	2	$O(1)$ ('order 1')
$ax + y$	$2n$	$O(n)$ ('order n')
$x^T y (= \sum_{i=1}^{n} x_i y_i)$	$2n$	$O(n)$
$c(xz^T)$	$2nm$	$O(nm)$ ('order nm')
Az	$2n^2$ (if $m = n$)	$O(n^2)$ ('order n-squared')
AB	$2n^3$ (if $n = m = p$)	$O(n^3)$ ('order n-cubed')

If $p = m = n$, the flop count $2n^3 + 3n^2 + 2n$ of $AB + (x^T y)(xz^T)$ is dominated by the term n^3 for large n. So this is an $O(n^3)$ process.

When comparing with another $O(n^3)$ process, the coefficient 2 of n^3 is important. Then the flop count is stated as $2n^3 + O(n^2)$.

In general the flop count is stated to be $O(n^p)$ if it is a polynomial in n of degree p.

However to be precise, *the flop count $g(n)$ of an algorithm is $O(f(n))$ if there is a positive integer n_0 and a constant c such that*

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

BLAS

The flop counts discussed are for some basic linear algebra operations that frequently occur in linear algebra computations.

BLAS $\longrightarrow$ Basic Linear Algebra Subprogrammes are programmes to execute these operations (often optimally designed for particular machines) for the use of software developers.

BLAS I $\longrightarrow$ execute scalar-vector, vector-vector operations costing $O(n)$ flops.
Ex: $ax, x + y, x^T y$.

BLAS II $\longrightarrow$ execute vector-vector and matrix-vector operations costing $O(n^2)$ or $O(nm)$ flops
Ex: Ax, Solving $Ax = b$ for x when A is upper or lower triangular.

BLAS III $\longrightarrow$ execute matrix-matrix operations costing $O(n^3)$, $O(n^2m)$, $O(nm^2)$ or $O(nmp)$ flops ($n \neq m \neq p$).
Ex: AB, $A^T B$.

To know more about BLAS see www.netlib.org/blas/

Optimal use of high level BLAS leads to efficient algorithms.

Partitioned matrices and vectors

Partitioning matrices and vectors involved in the computation is essential for efficient BLAS implementation and also parallelizing codes.

Consider $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times l}$, $x = \mathbb{R}^m$, partitioned conformally as

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1p} \\ \vdots & \ddots & \vdots \\ A_{q1} & \cdots & A_{qp} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & \cdots & B_{1r} \\ \vdots & \ddots & \vdots \\ B_{p1} & \cdots & B_{pr} \end{bmatrix}; \quad x = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_p \end{bmatrix},$$

where $A_{ij} \in \mathbb{R}^{n_i\times m_j}$, $B_{jk} \in \mathbb{R}^{m_j\times l_k}$, $\mathbf{x}_j \in \mathbb{R}^{m_j}$ where $\sum_{i=1}^q n_i = n$, $\sum_{k=1}^r l_k = l$ and $\sum_{j=1}^p m_j = m$. Then,

$$b := Ax = \begin{bmatrix} \underbrace{\sum_{j=1}^p A_{1j}\mathbf{x}_j}_{=:\mathbf{b}_1} \\ \vdots \\ \underbrace{\sum_{j=1}^p A_{qj}\mathbf{x}_j}_{=:\mathbf{b}_q} \end{bmatrix}, \ \& \ AB =: C = [C_{ik}] \text{ where } C_{ik} = \underbrace{\sum_{j=1}^p A_{ij}B_{jk}}_{n_j\times l_k}.$$

Partitioning and the use of high level BLAS

Pseudocode for computing b without partitioning:

Set $b \longrightarrow$ length m zero vector.

for $i = 1 : n$
    for $j = 1 : m$
        $b_i = b_i + a_{ij}x_j$ (uses BLAS I)
    end
end

Pseudocode for computing b with partitioning (blocked version):

for $i = 1 : q$
    for $j = 1 : p$
        $\mathbf{b}_i = \mathbf{b}_i + A_{ij}\mathbf{x}_j$ (uses BLAS II and reduces memory traffic)
    end
end

Partitioning and the use of high level BLAS

Pseudocode for computing C without partitioning:

Set $C \longrightarrow$ size $n \times l$ zero matrix.

for $i = 1 : n$
    for $t = 1 : l$
        for $j = 1 : m$
            $c_{it} = c_{it} + a_{ij} b_{jt}$ (uses BLAS I)
        end
    end
end

Pseudocode for computing C with partitioning (blocked version):

for $i = 1 : q$
    for $k = 1 : r$
        for $j = 1 : p$
            $C_{ik} = C_{ik} + A_{ij} B_{jk}$ (uses BLAS III & reduces memory traffic)
        end
    end
end

Matrix and Vector norms

Let $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$.

Vector norms: A vector norm is a norm on $\mathbb{F}^n$. It is a non-negative function $\|\cdot\|$ defined on $\mathbb{F}^n$ such that for all $x, y \in \mathbb{F}^n$ and $a \in \mathbb{F}$,

1. $\|x\| = 0 \Leftrightarrow x = 0$.
2. $\|ax\| = |a| \, \|x\|$.
3. $\|x + y\| \leq \|x\| + \|y\|$.

Examples:
$\|x\|_p = \left(\sum_{i=0}^{n} |x_i|^p\right)^{1/p}$, $p = 1, 2, \ldots,$ (p-norm)
$\|x\|_\infty = \max_{1\leq i\leq n} |x_i|$, (max-norm/sup-norm/$\infty$-norm)

Most commonly used vector norms:
$\|x\|_1$ (1-norm or taxi-cab norm), $\|x\|_\infty$ (∞-norm),
$\|x\|_2$ (2-norm).

Matrix and Vector norms

Matrix norms: A matrix norm is a norm on $\mathbb{F}^{n\times m}$. It is a non-negative function on $\mathbb{F}^{n\times m}$ such that for all $A, B \in \mathbb{F}^{n\times m}$ and $a \in \mathbb{F}$,

1. $\|A\| = 0 \Leftrightarrow A = 0$.
2. $\|aA\| = |a| \, \|A\|$.
3. $\|A + B\| \leq \|A\| + \|B\|$.
4. If $n = m$, then $\|AB\| \leq \|A\| \, \|B\|$.

Examples:

$\|A\|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} |a_{ij}|^2\right)^{1/2}$ (Frobenius norm)

$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$

(Operator norm $\|\cdot\|$ on matrices induced by vector norm $\|\cdot\|$)

Most commonly used matrix norms:

$\|A\|_F$ (This is not induced by any vector norm! Why?)

$\|A\|_2$ (2-norm/spectral norm induced by vector norm $\|\cdot\|_2$);

$\|A\|_1$ (1-norm induced by vector norm $\|\cdot\|_1$);

$\|A\|_\infty$ (∞-norm induced by vector norm $\|\cdot\|_\infty$).

Important properties of matrix and vector norms

Exercise: For $x, y \in \mathbb{F}^n$,

1. $|x^T y| \leq \|x\|_2 \|y\|_2$.
2. $\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\,\|x\|_2 \leq n\,\|x\|_\infty$.

Exercise: For any $A \in \mathbb{F}^{n\times n}$,

1. $\|A\|_1 = \max_{1\leq j\leq n}\left(\sum_{1\leq i\leq n} |a_{ij}|\right)$.
2. $\|A\|_\infty = \max_{1\leq i\leq n}\left(\sum_{1\leq j\leq n} |a_{ij}|\right)$.
3. $\|A\|_1 \leq \sqrt{n}\,\|A\|_2 \leq n\,\|A\|_1$.
4. $\|A\|_\infty \leq \sqrt{n}\,\|A\|_2 \leq n\,\|A\|_\infty$.
5. $\|A\|_2 \leq \|A\|_F \leq n\,\|A\|_2$.
6. $\|Ax\|_2 \leq \|A\|_F\,\|x\|_2$ for all $x \in \mathbb{F}^n$.

Important properties of matrix and vector norms

Exercise: For any induced operator norm on $A \in \mathbb{F}^{n \times n}$,

1. $\|Ax\| \leq \|A\| \, \|x\|$, for all $x \in \mathbb{F}^n$.
2. $\|A\| = \sup_{x \neq 0} \|Ax\| / \|x\| = \sup_{\|x\|=1} \|Ax\| = \sup_{\|x\| \leq 1} \|Ax\|$.
3. Each of the optimizations in part 2 is attained by some $x_0 \in \mathbb{F}^n$. Therefore the sup may be replaced by max in each case.

Solve all problems in pp 113-121, in *Fundamentals of Matrix Computations, 2nd Edition.*