Máster Universitario en Computación Paralela y Distribuida
Algoritmos Matriciales Paralelos en Ingeniería

Tema 0. Análisis Matricial

Algoritmos, Notación, Evaluación, Ejemplos

Bibliografía:

"Matrix Computations". G.Golub & C. Van Loan. Baltimore ; London : Johns Hopkins University Press, 1996 (u otra edición del libro)

Lecturas recomendadas:

"Matrix Computations". G.Golub & C. Van Loan.

Capítulo 1. Puntos 1.1, 1.2 y 1.3

Capítulo 2. Puntos 2.1, 2.2 y 2.3

Escalares, Vectores y Matrices: Notación

- Escalares: suelen representarse por letras griegas o con subíndices.

$$\alpha \in \Re, \quad \beta \in C, \quad a_{ij} \in \Re$$

- Vectores: suelen representarse con letras latinas minúsculas

$$a \in \Re^{n} \quad \text{o} \quad a \in \Re^{n\times 1} \qquad a = \begin{bmatrix} a_1 \\ a_2 \\ .. \\ a_n \end{bmatrix} \qquad b^T = \begin{bmatrix} b_1 & b_2 & .. & b_n \end{bmatrix}$$

Escalares, Vectores y Matrices: Notación

Matrices: suelen representarse con letras latinas mayúsculas

$A \in \Re^{m\times n} \qquad A = (a_{ij}) \text{ con } a_{ij} \in \Re$

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

Partición de Matrices por filas y columnas:

$A \in \Re^{m\times n}$ puede expresarse como $A = \begin{bmatrix} a_1^T \\ a_2^T \\ .. \\ a_m^T \end{bmatrix}$ o como $B = \begin{bmatrix} b_1 & b_2 & .. & b_n \end{bmatrix}$

Operaciones básicas

- Producto escalar

$$a,b \in \Re^m: \quad \alpha = a^T b = \sum_{i=1}^{m} a_i b_i \in \Re$$

- Operación saxpy

$$x,y \in \Re^m, \alpha \in \Re: \quad z = \alpha x + y \in \Re^m$$

$$\Rightarrow z_i = \alpha x_i + y_i, i = 1,2,\ldots,m$$

Operaciones básicas

- Producto Matriz-Vector

$$A \in \Re^{m\times n}, b \in \Re^{n}: \quad x = Ab \in \Re^{m}$$

$$x = Ab = \begin{bmatrix} a_1{}^T b \\ a_2{}^T b \\ .. \\ a_m{}^T b \end{bmatrix} \qquad \text{con } A = \begin{bmatrix} a_1{}^T \\ a_2{}^T \\ .. \\ a_m{}^T \end{bmatrix}$$

$$x = Ab = b_1 A_1 + b_2 A_2 + \ldots b_n A_n \text{ con } A = \begin{bmatrix} A_1 & A_2 & \ldots & A_n \end{bmatrix} \text{ y } \mathrm{b} = \begin{bmatrix} b_1 \\ b_2 \\ .. \\ b_n \end{bmatrix}$$

Operaciones básicas

- Producto Matriz-Matriz

$$A \in \Re^{m\times r}, B \in \Re^{r\times n}: \quad C = AB = \left[C_{ij}\right] = \left[\sum_{k=1}^{r} A_{ik}B_{kj}\right] \in \Re^{m\times n}$$

$$C = AB = \begin{bmatrix} a_1{}^T \\ a_2{}^T \\ .. \\ a_m{}^T \end{bmatrix} \begin{bmatrix} b_1 & b_2 & .. & b_n \end{bmatrix} = \begin{bmatrix} a_1{}^T B \\ a_2{}^T B \\ .. \\ a_m{}^T B \end{bmatrix} = \begin{bmatrix} a_1{}^T b_1 & a_1{}^T b_2 & .. & a_1{}^T b_n \\ a_2{}^T b_1 & a_2{}^T b_2 & .. & a_2{}^T b_n \\ .. & .. & .. & .. \\ a_m{}^T b_1 & a_m{}^T b_2 & .. & a_m{}^T b_n \end{bmatrix}$$

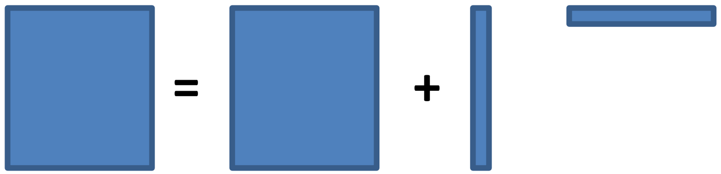
$$C = AB = \begin{bmatrix} Ab_1 & Ab_2 & .. & Ab_n \end{bmatrix}$$

Operaciones básicas

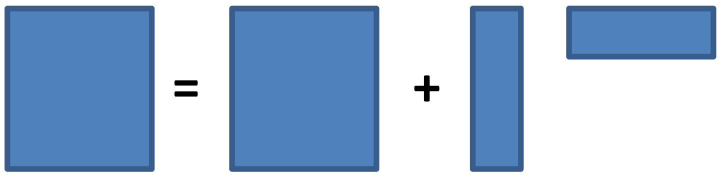
- Actualización de rango *1* o *k* de una matriz

$$A \in \Re^{m\times n}, b \in \Re^{m\times 1}, c^T \in \Re^{1\times n}: \quad E = A + bc^T = A + \begin{bmatrix} b_1 \\ b_2 \\ .. \\ b_m \end{bmatrix} \begin{bmatrix} c_1 & c_2 & .. & c_n \end{bmatrix} \Rightarrow$$

$$\left[E_{ij}\right] = \left[A_{ij} + b_i c_j\right] \in \Re^{m\times n}$$



$$A \in \Re^{m\times n}, B \in \Re^{m\times k}, C \in \Re^{k\times n}: \quad E = A + BC = \left[E_{ij}\right] = \left[A_{ij} + \sum_{s=1}^{k} B_{is} C_{sj}\right] \in \Re^{m\times n}$$



Algoritmos: Notación

Producto escalar

$$a,b \in \Re^m: \quad \alpha = a^T b \in \Re$$

$\alpha = 0$

Para $i = 1,2,\ldots,m$

$\quad \alpha = \alpha + a_i b_i$

FinPara

Producto matriz-vector

$$A \in \Re^{m\times n}, b \in \Re^n, 0 \in \Re^m: \quad x = Ab \in \Re^m$$

$x = 0$

Para $i = 1,2,\ldots,m$

$\quad$ Para $j = 1,2,\ldots,n$

$\quad\quad x_i = x_i + A_{ij} b_i$

$\quad$ FinPara

FinPara

Producto de una matriz triangular superior por una inferior

$L, U \in \Re^{m\times m}$: Calcular UL

L es triangular inferior y U triangular superior

Para $i = 1,2,\ldots,m$

$\quad$ Para $j = 1,2,\ldots,m$

$\quad\quad C_{ij} = 0$

$\quad\quad$ Para $k = \max(i,j), \max(i,j)+1,\ldots,m$

$\quad\quad\quad C_{ij} = C_{ij} + U_{ik} * L_{kj}$

$\quad\quad$ FinPara

$\quad$ FinPara

FinPara

Evaluación de los algoritmos. Concepto de Flop

- **Flop: Operación elemental en coma flotante {+,-,*,/}**

Permite evaluar algoritmos

Producto matriz-vector

$a \in \Re^{m\times n}, b \in \Re^{n}, 0 \in \Re^{m}: \quad x = Ab \in \Re^{m}$

$x = 0$

Para $i = 1,2,\ldots,m$

Para $j = 1,2,\ldots,n$

$x_i = x_i + A_{ij} b_i$

FinPara

FinPara

$\boxed{T_1 = 2mn \; Flops}$

Resolución de un sistema triangular

$Ux=b,\ U \in \Re^{nxn},\ U_{ii} \neq 0,\ x,b \in \Re^{nx1}.$

$x_n=b_n/U_{nn};$

Para j=n-1 hasta 1

$x_j = b_j;$

Para i=j+1 hasta n

$x_j = x_j - U_{ji} * x_i;$

Finpara

$x_j = x_j/U_{jj};$

Finpara.

$\boxed{T_1 \approx n^2 Flops}$

Aproximaciones

$$\sum_{i=1}^{n} i \approx \frac{n^2}{2} \qquad \sum_{i=1}^{n} i^2 \approx \frac{n^3}{3}$$

Organización de los algoritmos matriciales por niveles

- Concepto de nivel: clase de algoritmos que contienen operaciones con el mismo número de Flops (en términos de orden superior) y hacen uso de una cantidad de memoria proporcional al número de Flops:
- Nivel 1: Algoritmos de complejidad $O(n)$
- Nivel 2: Algoritmos de complejidad $O(n^2)$ o $O(mn)$
- Nivel 3: Algoritmos de complejidad $O(n^3)$ o $O(m^2n)$

Organización de los algoritmos matriciales por bloques

- La organización por bloques de las matrices suele dar lugar a algoritmos más eficientes
- Se usa especialmente en algoritmos de nivel 3

$$A = \begin{bmatrix} a_{11} & a_{12} & .. & a_{1n} \\ a_{21} & a_{22} & .. & a_{2n} \\ .. & .. & .. & .. \\ a_{m1} & a_{m2} & .. & a_{mn} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & .. & A_{1r} \\ A_{21} & A_{22} & .. & A_{2r} \\ .. & .. & .. & .. \\ A_{s1} & A_{s2} & .. & A_{sr} \end{bmatrix}$$

con $r = n/tb_1$ y $s = m/tb_2$ y $A_{ij} \in \Re^{tb_1 \times tb_2}$

Organización de los algoritmos matriciales por bloques

- La multiplicación de matrices por bloques sigue las mismas pautas que con matrices de escalares, respetando la coherencia de los bloques multiplicando:

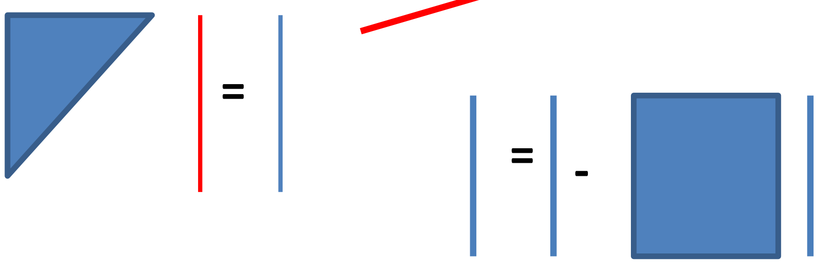
$$A = \begin{bmatrix} A_{11} & A_{12} & .. & A_{1q} \\ A_{21} & A_{22} & .. & A_{2q} \\ .. & .. & .. & .. \\ A_{p1} & A_{p2} & .. & A_{pq} \end{bmatrix}, \mathrm{B} = \begin{bmatrix} B_{11} & B_{12} & .. & B_{1t} \\ B_{21} & B_{22} & .. & B_{2t} \\ .. & .. & .. & .. \\ B_{q1} & B_{q2} & .. & B_{qt} \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} & .. & C_{1t} \\ C_{21} & C_{22} & .. & C_{2t} \\ .. & .. & .. & .. \\ C_{p1} & C_{p2} & .. & C_{pt} \end{bmatrix}, \text{con } C_{ij} = \sum_{k=1}^{q} A_{ik} B_{kj},\ i = 1,2,\ldots,p\ \ ;\ \ j = 1,2,\ldots,t$$

- **Ejemplo: Resolución por bloques de un sistema de ecuaciones triangular superior**

$$\begin{bmatrix} U_{11} & U_{12} & .. & U_{1q} \\ & U_{22} & .. & U_{2q} \\ .. & .. & .. & .. \\ & & .. & U_{qq} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ .. \\ x_q \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ .. \\ b_q \end{bmatrix} \Rightarrow \begin{cases} \begin{bmatrix} U_{11} & U_{12} & .. & U_{1,q-1} \\ & U_{22} & .. & U_{2,q-1} \\ .. & .. & .. & .. \\ & & .. & U_{q-1,q-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ .. \\ x_{q-1} \end{bmatrix} + \begin{bmatrix} U_{1q} \\ U_{2q} \\ .. \\ U_{q-1,q} \end{bmatrix} x_q = \begin{bmatrix} b_1 \\ b_2 \\ .. \\ b_{q-1} \end{bmatrix} \\ U_{qq} x_q = b_q \end{cases}$$

$$\begin{bmatrix} U_{11} & U_{12} & .. & U_{1,q-1} \\ & U_{22} & .. & U_{2,q-1} \\ .. & .. & .. & .. \\ & & .. & U_{q-1,q-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ .. \\ x_{q-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ .. \\ b_{q-1} \end{bmatrix} - \begin{bmatrix} U_{1q} \\ U_{2q} \\ .. \\ U_{q-1,q} \end{bmatrix} x_q$$



$Para \;\; j = q, q-1, .., 1$

  Resolver $U_{jj} x_j = b_j$

  $Para \;\; i = 1, 2, .., j-1$

    $b_i = b_i - U_{ij} x_j$

  $FinPara$

$FinPara$

Algebra lineal

- Dependencia e independencia lineal

  Un conjunto de vectores $\{a_1, a_2, .., a_n\}$, con $a_i \in \Re^m$, es linealmente independiente si

  $$\sum_{\mathrm{j=1}}^{n} \alpha_j a_j = 0 \Rightarrow \alpha_j = 0, \forall j$$

  En otro caso, existe una combinación lineal $\sum_{\mathrm{j=1}}^{n} \alpha_j a_j = 0$ con algún $\alpha_j \neq 0$,

  el conjunto es linealmente dependiente

- Subespacios en

  Dado un conjunto de vectores $\{a_1, a_2, .., a_n\}$, con $a_i \in \Re^m$,

  se denomina $span\{a_1, a_2, .., a_n\} = \left\{ \sum_{j=1}^{n} \beta_j a_j,\ con\ \beta_j \in \Re \right\}$

  Este conjunto es un subespacio de $\Re^m$

- Base de un subespacio:

$\{a_{i_1}, a_{i_2}, .., a_{i_k}\}$, con $a_{i_j} \in \Re^m$, es una base de $span\{a_1, a_2, .., a_n\}$ si :

* es linealmente independiente

* no está propiamente incluido en ningún otro subconjunto linealmente independiente de $\{a_1, a_2, .., a_n\}$

* Dado un subespacio cualquiera de $\Re^m$, siempre existe una base, conjunto de vectores $\{a_1, a_2, .., a_n\}$ linealmente independiente, que genera el subespacio $S = span\{a_1, a_2, .., a_n\}$

* Todas las bases de un subepacio de $\Re^m$ tienen el mismo cardinal. A este cardinal se le denomina dimensión.

- Espacio rango de A: range(A) $\quad A \in \Re^{m\times n}$

$$range(A) = \{ y \in \Re^m : y = Ax \text{ para algún } x \in \Re^n \}$$

Si $A \in \Re^{m\times n}$ se expresa por columnas $A = \begin{bmatrix} a_1 & a_2 & .. & a_n \end{bmatrix}$:

$$range(A) = span\{a_1, a_2, \ldots, a_n\}$$

- Rango de una matriz: rank(A)

$$rank(A) = \dim(range(A))$$

- Espacio nulo

$$null(A) = \{ x \in \Re^n : Ax = 0 \}$$