Notes on Vector and Matrix Operations

Robert A. van de Geijn
Department of Computer Science
The University of Texas at Austin
Austin, TX 78712
rvdg@cs.utexas.edu

September 2, 2014

1 (Hermitian) Transposition

In the following, assume that $\alpha \in \mathbb{C}$, $x \in \mathbb{C}^n$, and $A \in \mathbb{C}^{m \times n}$ with

$$x = \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{pmatrix}$$

1.1 Conjugating a complex scalar

Recall that if $\alpha = \alpha_r + i\alpha_c$, then its (complex) conjugate is given by

$$\overline{\alpha} = \alpha_r - i\alpha_c$$

and its length (absolute value) by

$$|\alpha| = |\alpha_r + i\alpha_c| = \sqrt{\alpha_r^2 + \alpha_c^2} = (\alpha_r + i\alpha_c)(\alpha_r - i\alpha_c) = \alpha\overline{\alpha} = \overline{\alpha}\alpha = |\overline{\alpha}|.$$

1.2 Conjugate of a vector

The *(complex) conjugate* of x is given by

$$\overline{x} \begin{pmatrix} \overline{\chi}_0 \\ \overline{\chi}_1 \\ \vdots \\ \overline{\chi}_{n-1} \end{pmatrix}.$$

1.3 Conjugate of a matrix

The *(complex) conjugate* of A is given by

$$\overline{A} = \begin{pmatrix} \overline{\alpha}_{0,0} & \overline{\alpha}_{0,1} & \cdots & \overline{\alpha}_{0,n-1} \\ \overline{\alpha}_{1,0} & \overline{\alpha}_{1,1} & \cdots & \overline{\alpha}_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ \overline{\alpha}_{m-1,0} & \overline{\alpha}_{m-1,1} & \cdots & \overline{\alpha}_{m-1,n-1} \end{pmatrix}.$$

1.4 Transpose of a vector

The *transpose* of x is given by

$$x^T = \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix}^T = \left(\begin{array}{c|c|c|c} \chi_0 & \chi_1 & \cdots & \chi_{n-1} \end{array} \right).$$

Notice that transposing a (column) vector rearranges its elements to make a row vector.

1.5 Hermitian transpose of a vector

The *Hermitian transpose* of x is given by

$$x^H (= x^c) = (\overline{x})^T = \overline{\begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix}}^T = \begin{pmatrix} \overline{\chi}_0 \\ \overline{\chi}_1 \\ \vdots \\ \overline{\chi}_{n-1} \end{pmatrix}^T = \left(\begin{array}{c|c|c|c} \overline{\chi}_0 & \overline{\chi}_1 & \cdots & \overline{\chi}_{n-1} \end{array} \right)^T.$$

1.6 Transpose of a matrix

The *transpose* of A is given by

$$A^T = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{pmatrix}^T = \begin{pmatrix} \alpha_{0,0} & \alpha_{1,0} & \cdots & \alpha_{m-1,0} \\ \alpha_{0,1} & \alpha_{1,1} & \cdots & \alpha_{m-1,1} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha_{0,n-1} & \alpha_{1,n-1} & \cdots & \alpha_{m-1,n-1} \end{pmatrix}.$$

1.7 Hermitian transpose (adjoint) of a matrix

The *Hermitian transpose* of A is given by

$$A^H (= A^c) = \overline{A}^T = \left(\overline{\begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{pmatrix}} \right)^T = \begin{pmatrix} \overline{\alpha}_{0,0} & \overline{\alpha}_{1,0} & \cdots & \overline{\alpha}_{m-1,0} \\ \overline{\alpha}_{0,1} & \overline{\alpha}_{1,1} & \cdots & \overline{\alpha}_{m-1,1} \\ \vdots & \vdots & \cdots & \vdots \\ \overline{\alpha}_{0,n-1} & \overline{\alpha}_{1,n-1} & \cdots & \overline{\alpha}_{m-1,n-1} \end{pmatrix}.$$

1.8 Exercises

Exercise 1. *Partition A*

$$A = \left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{n-1} \end{array} \right) = \left(\begin{array}{c} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{array} \right).$$

Convince yourself that the following hold:

- $\left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{n-1} \end{array} \right)^T = \left(\begin{array}{c} a_0^T \\ \hline a_1^T \\ \hline \vdots \\ \hline a_{m-1}^T \end{array} \right).$
- $\left(\begin{array}{c} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{array} \right)^T = \left(\begin{array}{c|c|c|c} \widehat{a}_0 & \widehat{a}_1 & \cdots & \widehat{a}_{n-1} \end{array} \right).$
- $\left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{n-1} \end{array} \right)^H = \left(\begin{array}{c} a_0^H \\ \hline a_1^H \\ \hline \vdots \\ \hline a_{m-1}^H \end{array} \right).$
- $\left(\begin{array}{c} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{array} \right)^H = \left(\begin{array}{c|c|c|c} \overline{\widehat{a}_0} & \overline{\widehat{a}_1} & \cdots & \overline{\widehat{a}_{n-1}} \end{array} \right).$

Exercise 2. *Partition x into subvectors:*

$$x = \left(\begin{array}{c} x_0 \\ \hline x_1 \\ \hline \vdots \\ \hline x_{N-1} \end{array} \right).$$

Convince yourself that the following hold:

- $\overline{x} = \left(\begin{array}{c} \overline{x_0} \\ \hline \overline{x_1} \\ \hline \vdots \\ \hline \overline{x_{N-1}} \end{array} \right).$
- $x^T = \left(\begin{array}{c|c|c|c} x_0^T & x_1^T & \cdots & x_{N-1}^T \end{array} \right).$
- $x^H = \left(\begin{array}{c|c|c|c} x_0^H & x_1^H & \cdots & x_{N-1}^H \end{array} \right).$

Exercise 3. *Partition A*

$$A = \left(\begin{array}{cccc} A_{0,0} & A_{0,1} & \cdots & A_{0,N-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N-1} \\ \vdots & \vdots & \cdots & \vdots \\ A_{M-1,0} & A_{M-1,1} & \cdots & A_{M-1,N-1} \end{array} \right),$$

where $A_{i,j} \in \mathbb{C}^{m_i \times n_i}$. Here $\sum_{i=0}^{M-1} m_i = m$ and $\sum_{j=0}^{N-1} n_i = n$.
Convince yourself that the following hold:

- $\left(\begin{array}{cccc} A_{0,0} & A_{0,1} & \cdots & A_{0,N-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N-1} \\ \vdots & \vdots & \cdots & \vdots \\ A_{M-1,0} & A_{M-1,1} & \cdots & A_{M-1,N-1} \end{array} \right)^T = \left(\begin{array}{cccc} A_{0,0}^T & A_{1,0}^T & \cdots & A_{M-1}^T \\ A_{0,1}^T & A_{1,1}^T & \cdots & A_{M-1,1}^T \\ \vdots & \vdots & \cdots & \vdots \\ A_{0,N-1}^T & A_{1,N-1}^T & \cdots & A_{M-1,N-1}^T \end{array} \right).$
- $\left(\begin{array}{cccc} A_{0,0} & A_{0,1} & \cdots & A_{0,N-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N-1} \\ \vdots & \vdots & \cdots & \vdots \\ A_{M-1,0} & A_{M-1,1} & \cdots & A_{M-1,N-1} \end{array} \right)^H = \left(\begin{array}{cccc} A_{0,0}^H & A_{1,0}^H & \cdots & A_{M-1}^H \\ A_{0,1}^H & A_{1,1}^H & \cdots & A_{M-1,1}^H \\ \vdots & \vdots & \cdots & \vdots \\ A_{0,N-1}^H & A_{1,N-1}^H & \cdots & A_{M-1,N-1}^H \end{array} \right).$

2 Vector-vector Operations

2.1 Scaling a vector (`scal`)

Let $x \in \mathbb{C}^n$ and $\alpha \in \mathbb{C}$, with

$$x = \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix}.$$

Then αx equals the vector x "stretched" by a factor α:

$$\alpha x = \alpha \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix} = \begin{pmatrix} \alpha\chi_0 \\ \alpha\chi_1 \\ \vdots \\ \alpha\chi_{n-1} \end{pmatrix}.$$

If $y := \alpha x$ with

$$y = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{pmatrix},$$

then the following loop computes y:

for $i := 0, \ldots, n-1$
  $\psi_i := \alpha \chi_i$
endfor

Exercise 4. *Convince yourself of the following:*

- $\alpha x^T = \left(\begin{array}{c|c|c|c} \alpha\chi_0 & \alpha\chi_1 & \cdots & \alpha\chi_{n-1} \end{array} \right).$
- $(\alpha x)^T = \alpha x^T$.
- $(\alpha x)^H = \overline{\alpha} x^H$.
- $\alpha \left(\begin{array}{c} x_0 \\ \hline x_1 \\ \hline \vdots \\ \hline x_{N-1} \end{array} \right) = \left(\begin{array}{c} \alpha x_0 \\ \hline \alpha x_1 \\ \hline \vdots \\ \hline \alpha x_{N-1} \end{array} \right)$

2.2 Scaled vector addition (`axpy`)

Let $x, y \in \mathbb{C}^n$ and $\alpha \in \mathbb{C}$, with

$$x = \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix} \quad \text{and} \quad y = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{pmatrix}.$$

Then $\alpha x + y$ equals the vector

$$\alpha x + y = \alpha \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix} + \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{pmatrix} = \begin{pmatrix} \alpha\chi_0 \\ \alpha\chi_1 \\ \vdots \\ \alpha\chi_{n-1} \end{pmatrix} + \begin{pmatrix} \alpha\psi_0 \\ \alpha\psi_1 \\ \vdots \\ \alpha\psi_{n-1} \end{pmatrix}.$$

This operation is known as the `axpy` operation: scalar α times x plus y. Typically, the vector y is overwritten with the result:

$$y := \alpha x + y = \alpha \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix} + \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{pmatrix} = \begin{pmatrix} \alpha\chi_0 \\ \alpha\chi_1 \\ \vdots \\ \alpha\chi_{n-1} \end{pmatrix} + \begin{pmatrix} \alpha\psi_0 \\ \alpha\psi_1 \\ \vdots \\ \alpha\psi_{n-1} \end{pmatrix}$$

so that the following loop updates y:

for $i := 0, \ldots, n-1$
  $\psi_i := \alpha\chi_i + \psi$
endfor

Exercise 5. *Convince yourself of the following:*

- $\alpha \begin{pmatrix} x_0 \\ \hline x_1 \\ \hline \vdots \\ \hline x_{N-1} \end{pmatrix} + \begin{pmatrix} y_0 \\ \hline y_1 \\ \hline \vdots \\ \hline y_{N-1} \end{pmatrix} = \begin{pmatrix} \alpha x_0 + y_0 \\ \hline \alpha x_1 + y_1 \\ \hline \vdots \\ \hline \alpha x_{N-1} + y_{N-1} \end{pmatrix}$. *(Provided $x_i, y_i \in \mathbb{C}^{n_i}$ and $\sum_{i=0}^{N-1} n_i = n$.)*

2.3 Dot (inner) product (`dot`)

Let $x, y \in \mathbb{C}^n$ with

$$x = \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix} \quad \text{and} \quad y = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{pmatrix}.$$

Then the dot product of x and y is defined by

$$\begin{aligned} x^H y &= \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix}^H \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{pmatrix} = \begin{pmatrix} \overline{\chi_0} & \overline{\chi_1} & \cdots & \overline{\chi_{n-1}} \end{pmatrix} \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{pmatrix} \\ &= \overline{\chi_0}\psi_0 + \overline{\chi_1}\psi_1 + \cdots + \overline{\chi_{n-1}}\psi_{n-1} = \sum_{i=0}^{n-1} \overline{\chi_i}\psi_i. \end{aligned}$$

The following loop computes $\alpha := x^H y$:

$$\begin{array}{l} \alpha := 0 \\ \textbf{for } i := 0, \ldots, n-1 \\ \quad \alpha := \overline{\chi_i}\psi + \alpha \\ \textbf{endfor} \end{array}$$

Exercise 6. *Convince yourself of the following:*

- $\left(\begin{array}{c} x_0 \\ \hline x_1 \\ \hline \vdots \\ \hline x_{N-1} \end{array} \right)^H \left(\begin{array}{c} y_0 \\ \hline y_1 \\ \hline \vdots \\ \hline y_{N-1} \end{array} \right) = \sum_{i=0}^{N-1} x_i^H y_i$. *(Provided $x_i, y_i \in \mathbb{C}^{n_i}$ and $\sum_{i=0}^{N-1} n_i = n$.)*

Exercise 7. *Prove that $x^H y = \overline{y^H x}$.*

As we discuss matrix-vector multiplication and matrix-matrix multiplication, the closely related operation $x^T y$ is also useful:

$$\begin{aligned} x^T y &= \left(\begin{array}{c} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{array} \right)^T \left(\begin{array}{c} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{array} \right) = \left(\begin{array}{cccc} \chi_0 & \chi_1 & \cdots & \chi_{n-1} \end{array} \right) \left(\begin{array}{c} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{array} \right) \\ &= \chi_0\psi_0 + \chi_1\psi_1 + \cdots + \chi_{n-1}\psi_{n-1} = \sum_{i=0}^{n-1} \chi_i \psi_i. \end{aligned}$$

3 Matrix-vector Operations

3.1 Matrix-vector multiplication

Be sure to understand the relation between linear transformations and matrix-vector multiplication (LAFF Notes Week 2).

Let $y \in \mathbb{C}^m$, $A \in \mathbb{C}^{m \times n}$, and $x \in \mathbb{C}^n$ with

$$y = \left(\begin{array}{c} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{m-1} \end{array} \right), \quad A = \left(\begin{array}{cccc} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,n-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{array} \right), \quad \text{and} \quad x = \left(\begin{array}{c} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{array} \right).$$

Then $y = Ax$ means that

$$\left(\begin{array}{c} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{m-1} \end{array} \right) = \left(\begin{array}{cccc} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,n-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{array} \right) \left(\begin{array}{c} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{array} \right)$$

$$= \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,n-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{pmatrix} \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix}.$$

Now, partition

$$A = \left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{n-1} \end{array} \right) = \left(\begin{array}{c} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{array} \right).$$

Focusing on how A can be partitioned by columns, we find that

$$\begin{aligned} y &= Ax = \left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{n-1} \end{array} \right) \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix} \\ &= a_0\chi_0 + a_1\chi_1 + \cdots + a_{n-1}\chi_{n-1} \\ &= \chi_0 a_0 + \chi_1 a_1 + \cdots + \chi_{n-1} a_{n-1} \\ &= \chi_{n-1} a_{n-1} + (\cdots + (\chi_1 a_1 + (\chi_0 a_0 + 0)) \cdots), \end{aligned}$$

where 0 denotes the zero vector of size m. This suggests the following loop for computing $y := Ax$:

$y := 0$
for $j := 0, \ldots, n-1$
  $y := \chi_j a_j + y$ (`axpy`)
endfor

In Figure 1 (left), we present this algorithm using the FLAME notation (LAFF Notes Week 3).

Focusing on how A can be partitioned by rows, we find that

$$y = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{pmatrix} = Ax = \left(\begin{array}{c} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{array} \right) x = \left(\begin{array}{c} \widehat{a}_0^T x \\ \hline \widehat{a}_1^T x \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T x \end{array} \right).$$

This suggests the following loop for computing $y := Ax$:

for $i := 0, \ldots, m-1$
  $\psi_i := \widehat{a}_i^T x + \psi_i$ (`''dot''`)
endfor

Here we use the term "dot" because for complex valued matrices it is not really a dot product. In Figure 1 (right), we present this algorithm using the FLAME notation (LAFF Notes Week 3).

It is important to notice that this first "matrix-vector" operation (matrix-vector multiplication) can be "layered" upon vector-vector operations (`axpy` or `''dot''`).

Algorithm: $[y] := \textsc{Mvmult\_unb\_var1}(A, x, y)$

Partition $A \rightarrow \left(A_L \,\middle|\, A_R \right)$, $x \rightarrow \left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right)$
 where A_L is 0 columns, x_T has 0 elements
while $n(A_L) < n(A)$ **do**
 Repartition
$$\left(A_L \,\middle|\, A_R \right) \rightarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right), \quad \left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right) \rightarrow \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ \hline x_2 \end{array} \right)$$

$$y := \chi_1 a_1 + y$$

 Continue with
$$\left(A_L \,\middle|\, A_R \right) \leftarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right), \quad \left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right) \leftarrow \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ \hline x_2 \end{array} \right)$$
endwhile

Algorithm: $[y] := \textsc{Mvmult\_unb\_var2}(A, x, y)$

Partition $A \rightarrow \left(\begin{array}{c} A_T \\ \hline A_B \end{array} \right)$, $y \rightarrow \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right)$
 where A_T has 0 rows, y_T has 0 rows
while $m(A_T) < m(A)$ **do**
 Repartition
$$\left(\begin{array}{c} A_T \\ \hline A_B \end{array} \right) \rightarrow \left(\begin{array}{c} A_0 \\ \hline a_1^T \\ \hline A_2 \end{array} \right), \quad \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) \rightarrow \left(\begin{array}{c} y_0 \\ \hline \psi_1 \\ \hline y_2 \end{array} \right)$$

$$\psi_1 := a_1^T x + \psi_1$$

 Continue with
$$\left(\begin{array}{c} A_T \\ \hline A_B \end{array} \right) \leftarrow \left(\begin{array}{c} A_0 \\ \hline a_1^T \\ \hline A_2 \end{array} \right), \quad \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) \leftarrow \left(\begin{array}{c} y_0 \\ \hline \psi_1 \\ \hline y_2 \end{array} \right)$$
endwhile

Figure 1: Matrix-vector multiplication algorithms for $y := Ax + y$. Left: via `axpy` operations (by columns). Right: via "dot products" (by rows).

3.2 Rank-1 update

Let $y \in \mathbb{C}^m$, $A \in \mathbb{C}^{m \times n}$, and $x \in \mathbb{C}^n$ with

$$y = \left(\begin{array}{c} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{m-1} \end{array} \right), \quad A = \left(\begin{array}{cccc} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,n-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{array} \right), \quad \text{and} \quad x = \left(\begin{array}{c} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{array} \right).$$

The outerproduct of y and x is given by

$$\begin{aligned} yx^T &= \left(\begin{array}{c} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{m-1} \end{array} \right) \left(\begin{array}{c} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{array} \right)^T = \left(\begin{array}{c} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{m-1} \end{array} \right) \left(\begin{array}{cccc} \chi_0 & \chi_1 & \cdots & \chi_{n-1} \end{array} \right) \\ &= \left(\begin{array}{cccc} \psi_0\chi_0 & \psi_0\chi_1 & \cdots & \psi_0\chi_{n-1} \\ \psi_1\chi_0 & \psi_1\chi_1 & \cdots & \psi_1\chi_{n-1} \\ \vdots & \vdots & & \vdots \\ \psi_{m-1}\chi_0 & \psi_{m-1}\chi_1 & \cdots & \psi_{m-1}\chi_{n-1} \end{array} \right). \end{aligned}$$

Also,

$$yx^T = y \begin{pmatrix} \chi_0 & \chi_1 & \cdots & \chi_{n-1} \end{pmatrix} = \begin{pmatrix} \chi_0 y & \chi_1 y & \cdots & \chi_{n-1} y \end{pmatrix}.$$

This shows that all columns are a multiple of vector y. Finally,

$$yx^T = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{m-1} \end{pmatrix} x^T = \begin{pmatrix} \psi_0 x^T \\ \psi_1 x^T \\ \vdots \\ \psi_{m-1} x^T \end{pmatrix},$$

which shows that all columns are a multiple of row vector x^T. This motivates the observation that the matrix yx^T has rank at most equal to one (LAFF Notes Week 10).

The operation $A := yx^T + A$ is called a rank-1 update to matrix A:

$$\begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,n-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{pmatrix} :=$$

$$\begin{pmatrix} \psi_0\chi_0 + \alpha_{0,0} & \psi_0\chi_1 + \alpha_{0,1} & \cdots & \psi_0\chi_{n-1} + \alpha_{0,n-1} \\ \psi_1\chi_0 + \alpha_{1,0} & \psi_1\chi_1 + \alpha_{1,1} & \cdots & \psi_1\chi_{n-1} + \alpha_{1,n-1} \\ \vdots & \vdots & & \vdots \\ \psi_{m-1}\chi_0 + \alpha_{m-1,0} & \psi_{m-1}\chi_1 + \alpha_{m-1,1} & \cdots & \psi_{m-1}\chi_{n-1} + \alpha_{m-1,n-1} \end{pmatrix}.$$

Now, partition

$$A = \left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{n-1} \end{array} \right) = \left(\begin{array}{c} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{array} \right).$$

Focusing on how A can be partitioned by columns, we find that

$$\begin{array}{rcl} yx^T + A & = & \left(\begin{array}{c|c|c|c} \chi_0 y & \chi_1 y & \cdots & \chi_{n-1} y \end{array} \right) + \left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{n-1} \end{array} \right) \\ & = & \left(\begin{array}{c|c|c|c} \chi_0 y + a_0 & \chi_1 y + a_1 & \cdots & \chi_{n-1} y + a_{n-1} \end{array} \right). \end{array}$$

Notice that each column is updated with an `axpy` operation. This suggests the following loop for computing $A := yx^T + A$:

for $j := 0, \ldots, n-1$
  $a_j := \chi_j y + a_j$ (`axpy`)
endfor

In Figure 2 (left), we present this algorithm using the FLAME notation (LAFF Notes Week 3).

Focusing on how A can be partitioned by rows, we find that

$$yx^T + A = \left(\begin{array}{c} \psi_0 x^T \\ \hline \psi_1 x^T \\ \hline \vdots \\ \hline \psi_{m-1} x^T \end{array} \right) + \left(\begin{array}{c} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{array} \right)$$

Algorithm: $[A] := \text{RANK1\_UNB\_VAR1}(y, x, A)$

Partition $x \rightarrow \left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right)$, $A \rightarrow \left(\begin{array}{c|c} A_L & A_R \end{array} \right)$
  where x_T has 0 rows, A_L has 0 columns

while $m(x_T) < m(x)$ **do**

  Repartition

$$\left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right) \rightarrow \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ \hline x_2 \end{array} \right), \quad \left(\begin{array}{c|c} A_L & A_R \end{array} \right) \rightarrow \left(\begin{array}{c|c|c} A_0 & a_1 & A_2 \end{array} \right)$$

$$a_1 := y\chi_1 + a_1$$

  Continue with

$$\left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right) \leftarrow \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ \hline x_2 \end{array} \right), \quad \left(\begin{array}{c|c} A_L & A_R \end{array} \right) \leftarrow \left(\begin{array}{c|c|c} A_0 & a_1 & A_2 \end{array} \right)$$

endwhile

Algorithm: $[A] := \text{RANK1\_UNB\_VAR2}(y, x, A)$

Partition $y \rightarrow \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right)$, $A \rightarrow \left(\begin{array}{c} A_T \\ \hline A_B \end{array} \right)$
  where y_T has 0 rows, A_T has 0 rows

while $m(y_T) < m(y)$ **do**

  Repartition

$$\left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) \rightarrow \left(\begin{array}{c} y_0 \\ \hline \psi_1 \\ \hline y_2 \end{array} \right), \quad \left(\begin{array}{c} A_T \\ \hline A_B \end{array} \right) \rightarrow \left(\begin{array}{c} A_0 \\ \hline a_1^T \\ \hline A_2 \end{array} \right)$$

$$a_1^T := \psi_1 x^T + a_1^T$$

  Continue with

$$\left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) \leftarrow \left(\begin{array}{c} y_0 \\ \hline \psi_1 \\ \hline y_2 \end{array} \right), \quad \left(\begin{array}{c} A_T \\ \hline A_B \end{array} \right) \leftarrow \left(\begin{array}{c} A_0 \\ \hline a_1^T \\ \hline A_2 \end{array} \right)$$

endwhile

Figure 2: Rank-1 update algorithms for computing $A := yx^T + A$. Left: by columns. Right: by rows.

$$= \left(\begin{array}{c} \psi_0 x^T + \widehat{a}_0^T \\ \hline \psi_1 x^T + \widehat{a}_1^T \\ \hline \vdots \\ \hline \psi_{m-1} x^T + \widehat{a}_{m-1}^T \end{array} \right).$$

Notice that each row is updated with an `axpy` operation. This suggests the following loop for computing $A := yx^T + A$:

for $i := 0, \ldots, m-1$
  $\widehat{a}_j^T := \psi_i x^T + \widehat{a}_j^T$ (`axpy`)
endfor

In Figure 2 (right), we present this algorithm using the FLAME notation (LAFF Notes Week 3).

Again, it is important to notice that this "matrix-vector" operation (rank-1 update) can be "layered" upon the `axpy` vector-vector operation.

4 Matrix-matrix multiplication

Be sure to understand the relation between linear transformations and matrix-matrix multiplication (LAFF Notes Weeks 3 and 4).

We will now discuss the computation of $C := AB + C$, where $C \in \mathbb{C}^{m\times n}$, $A \in \mathbb{C}^{m\times k}$, and $B \in \mathbb{C}^{k\times n}$. (If one wishes to compute $C := AB$, one can always start by setting $C := 0$, the zero matrix.)

4.1 Element-by-element computation

Let

$$C = \begin{pmatrix} \gamma_{0,0} & \gamma_{0,1} & \cdots & \gamma_{0,n-1} \\ \gamma_{1,0} & \gamma_{1,1} & \cdots & \gamma_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ \gamma_{m-1,0} & \gamma_{m-1,1} & \cdots & \gamma_{m-1,n-1} \end{pmatrix}, A = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,k-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,k-1} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,k-1} \end{pmatrix}$$

$$B = \begin{pmatrix} \beta_{0,0} & \beta_{0,1} & \cdots & \beta_{0,n-1} \\ \beta_{1,0} & \beta_{1,1} & \cdots & \beta_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{k-1,0} & \beta_{k-1,1} & \cdots & \beta_{k-1,n-1} \end{pmatrix}.$$

Then

$$\begin{aligned} C &:= AB + C \\ &= \begin{pmatrix} \sum_{p=0}^{k-1} \alpha_{0,p}\beta_{p,0} + \gamma_{0,0} & \sum_{p=0}^{k-1} \alpha_{0,p}\beta_{p,1} + \gamma_{0,1} & \cdots & \sum_{p=0}^{k-1} \alpha_{0,p}\beta_{p,n-1} + \gamma_{0,n-1} \\ \sum_{p=0}^{k-1} \alpha_{1,p}\beta_{p,0} + \gamma_{1,0} & \sum_{p=0}^{k-1} \alpha_{1,p}\beta_{p,1} + \gamma_{1,1} & \cdots & \sum_{p=0}^{k-1} \alpha_{1,p}\beta_{p,n-1} + \gamma_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ \sum_{p=0}^{k-1} \alpha_{m-1,p}\beta_{p,0} + \gamma_{m-1,0} & \sum_{p=0}^{k-1} \alpha_{m-1,p}\beta_{p,1} + \gamma_{m-1,1} & \cdots & \sum_{p=0}^{k-1} \alpha_{m-1,p}\beta_{p,n-1} + \gamma_{m-1,n-1} \end{pmatrix}. \end{aligned}$$

This can be more elegantly stated by partitioning

$$A = \begin{pmatrix} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{pmatrix} \quad \text{and} \quad B = \left(\begin{array}{c|c|c|c} b_0 & b_1 & \cdots & b_{n-1} \end{array} \right).$$

Then

$$\begin{aligned} C \quad := \quad AB + C &= \begin{pmatrix} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{pmatrix} \left(\begin{array}{c|c|c|c} b_0 & b_1 & \cdots & b_{n-1} \end{array} \right) \\ &= \begin{pmatrix} \widehat{a}_0^T b_0 + \gamma_{0,0} & \widehat{a}_0^T b_1 + \gamma_{0,1} & \cdots & \widehat{a}_0^T b_{n-1} + \gamma_{0,n-1} \\ \widehat{a}_1^T b_0 + \gamma_{1,0} & \widehat{a}_1^T b_1 + \gamma_{1,1} & \cdots & \widehat{a}_1^T b_{n-1} + \gamma_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ \widehat{a}_{m-1}^T b_0 + \gamma_{m-1,0} & \widehat{a}_{m-1}^T b_1 + \gamma_{m-1,1} & \cdots & \widehat{a}_{m-1}^T b_{n-1} + \gamma_{m-1,n-1} \end{pmatrix}. \end{aligned}$$

4.2 Via matrix-vector multiplications

Partition

$$C = \left(\begin{array}{c|c|c|c} c_0 & c_1 & \cdots & c_{n-1} \end{array} \right) \quad \text{and} \quad B = \left(\begin{array}{c|c|c|c} b_0 & b_1 & \cdots & b_{n-1} \end{array} \right).$$

Then

$$C := AB + C = A \left(\begin{array}{c|c|c|c} b_0 & b_1 & \cdots & b_{n-1} \end{array} \right) + \left(\begin{array}{c|c|c|c} c_0 & c_1 & \cdots & c_{n-1} \end{array} \right) = \left(\begin{array}{c|c|c|c} Ab_0 + c_0 & Ab_1 + c_1 & \cdots & Ab_{n-1} + c_{n-1} \end{array} \right)$$

which shows that each column of C is updated with a matrix-vector multiplication: $c_j := Ab_j + c_j$:

for $j := 0, \ldots, n-1$
  $c_j := Ab_j + c_j$ (`matrix-vector multiplication`)
endfor

4.3 Via row-vector times matrix multiplications

Partition

$$C = \begin{pmatrix} \widehat{c}_0^T \\ \hline \widehat{c}_1^T \\ \hline \vdots \\ \hline \widehat{c}_{m-1}^T \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{pmatrix}.$$

Then

$$C := AB + C = \begin{pmatrix} \widehat{a}_0^T \\ \hline \widehat{a}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T \end{pmatrix} B + \begin{pmatrix} \widehat{c}_0^T \\ \hline \widehat{c}_1^T \\ \hline \vdots \\ \hline \widehat{c}_{m-1}^T \end{pmatrix} = \begin{pmatrix} \widehat{a}_0^T B + \widehat{c}_0^T \\ \hline \widehat{a}_1^T B + \widehat{c}_1^T \\ \hline \vdots \\ \hline \widehat{a}_{m-1}^T B + \widehat{c}_{m-1}^T \end{pmatrix}$$

which shows that each row of C is updated with a row-vector time matrix multiplication: $\widehat{c}_i^T := \widehat{a}_i^T B + \widehat{c}_i^T$:

for $i := 0, \ldots, m-1$
  $\widehat{c}_i^T := \widehat{a}_i^T B + \widehat{c}_i^T$ (`row-vector times matrix-vector multiplication`)
endfor

4.4 Via rank-1 updates

Partition

$$A = \left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{k-1} \end{array} \right) \quad \text{and} \quad B = \begin{pmatrix} \widehat{b}_0^T \\ \hline \widehat{b}_1^T \\ \hline \vdots \\ \hline \widehat{b}_{k-1}^T \end{pmatrix}.$$

The

$$\begin{aligned} C := AB + C &= \left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{k-1} \end{array} \right) \begin{pmatrix} \widehat{b}_0^T \\ \hline \widehat{b}_1^T \\ \hline \vdots \\ \hline \widehat{b}_{k-1}^T \end{pmatrix} + C \\ &= a_0 \widehat{b}_0^T + a_1 \widehat{b}_1^T + \cdots + a_{k-1} \widehat{b}_{k-1}^T + C \\ &= a_{k-1} \widehat{b}_{k-1}^T + (\cdots (a_1 \widehat{b}_1^T + (a_0 \widehat{b}_0^T + C)) \cdots) \end{aligned}$$

which shows that C can be updated with a sequence of rank-1 update, suggesting the loop

for $p := 0, \ldots, k-1$
  $C := a_p \widehat{b}_p^T + C$ (`rank-1 update`)
endfor

5 Summarizing All

The following actually summarizes ALL observations in this document:

Let

$$C = \begin{pmatrix} C_{0,0} & C_{0,1} & \cdots & C_{0,N-1} \\ C_{1,0} & C_{1,1} & \cdots & C_{1,N-1} \\ \vdots & \vdots & \cdots & \vdots \\ C_{M-1,0} & C_{M-1,1} & \cdots & C_{M-1,N-1} \end{pmatrix}, A = \begin{pmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,K-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,K-1} \\ \vdots & \vdots & \cdots & \vdots \\ A_{M-1,0} & A_{M-1,1} & \cdots & A_{M-1,K-1} \end{pmatrix}$$

$$B = \begin{pmatrix} B_{0,1} & B_{0,1} & \cdots & B_{0,N-1} \\ B_{1,0} & B_{1,1} & \cdots & B_{1,N-1} \\ \vdots & \vdots & \cdots & \vdots \\ B_{K-1,0} & B_{K-1,1} & \cdots & B_{K-1,N-1} \end{pmatrix}.$$

Then

$$\begin{aligned} C &:= AB + C \\ &= \begin{pmatrix} \sum_{p=0}^{K-1} A_{0,p}B_{p,0} + C_{0,0} & \sum_{p=0}^{K-1} A_{0,p}B_{p,1} + C_{0,1} & \cdots & \sum_{p=0}^{K-1} A_{0,p}B_{p,N-1} + C_{0,N-1} \\ \sum_{p=0}^{K-1} A_{1,p}B_{p,0} + C_{1,0} & \sum_{p=0}^{K-1} A_{1,p}B_{p,1} + C_{1,1} & \cdots & \sum_{p=0}^{K-1} A_{1,p}B_{p,N-1} + C_{1,N-1} \\ \vdots & \vdots & \cdots & \vdots \\ \sum_{p=0}^{K-1} A_{M-1,p}B_{p,0} + C_{M-1,0} & \sum_{p=0}^{k-1} A_{M-1,p}B_{p,1} + C_{M-1,1} & \cdots & \sum_{p=0}^{k-1} A_{M-1,p}B_{p,N-1} + C_{M-1,N-1} \end{pmatrix}. \end{aligned}$$

(Provided the partitionings of C, A, and B are "conformal.)