Matrix-Vector Operations

Jirasak Sittigorn
Department of Computer Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang

Opening Remarks

- Timmy Two Space

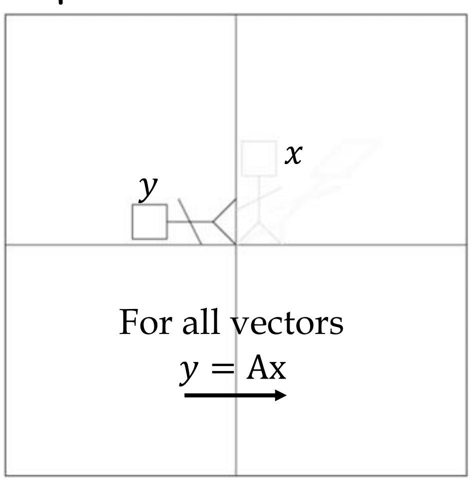
Homework 3.1.1.1 Click on the below link to open a browser window with the "Timmy Two Space" exercise. This exercise was suggested to us by our colleague Prof. Alan Cline. It was first implemented using an IPython Notebook by Ben Holder. During the Spring 2014 offering of LAFF on the edX platform, one of the participants, Ed McCardell, rewrote the activity as ☛ Timmy! on the web. (If this link does not work, open `LAFF-2.0xM/Timmy/index.html`).
If you get really frustrated, here is a hint:

☛ View at edX

http://edx-org-utaustinx.s3.amazonaws.com/UT501x/Spring2015/Timmy/index.html

Opening Remarks

- Timmy Two Space



Special Matrices

- The Zero Matrix
- The Identity Matrix
- Diagonal Matrices
- Triangular Matrices
- Transpose Matrix
- Symmetric Matrices

Special Matrices

$$0 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- The Zero Matrix
  - Definition 3.1
    - A matrix $A \in \mathbb{R}^{m \times n}$ equals the $m \times n$ zero matrix if all of its elements equal zero.
  - Let $L_0 : \mathbb{R}^n \to \mathbb{R}^m$ be the function defined for every $x \in \mathbb{R}^n$ as $L_0(x) = 0$, where 0 denotes the zero vector "of appropriate size". L_0 is a linear transformation.
    - True/False

$$L_0\left(\begin{pmatrix} \chi_0 \\ \chi_1 \\ \chi_2 \end{pmatrix}\right) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Througout this course, we will use the number 0 to indicate a scalar, vector, or matrix of "appropriate size".

Special Matrices

- The Zero Matrix

Spark PictureFLAME

Algorithm: $[A] := \text{SET\_TO\_ZERO}(A)$

Partition $A \to \left(A_L \,\middle|\, A_R \right)$
  where A_L has 0 columns

while $n(A_L) < n(A)$ **do**

  Repartition

  $\left(A_L \,\middle|\, A_R \right) \to \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right)$
  where a_1 has 1 column

  $a_1 := 0$ (Set the current column to zero)

  Continue with

  $\left(A_L \,\middle|\, A_R \right) \leftarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right)$

endwhile

Special Matrices

- The Zero Matrix

Homework 3.2.1.3 In the MATLAB Command Window, type

```
A = zeros( 5,4 )
```

What is the result?

Homework 3.2.1.4 Apply the zero matrix to Timmy Two Space. What happens?

1. Timmy shifts off the grid.
2. Timmy disappears into the origin.
3. Timmy becomes a line on the x-axis.
4. Timmy becomes a line on the y-axis.
5. Timmy doesn't change at all.

Special Matrices

$$0 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- The Identity Matrix
  - Definition 3.2
    - A matrix $\mathrm{I} \in \mathbb{R}^{n \times n}$ equals the $n \times n$ identity matrix if all its elements equal zero, except for the elements on the diagonal, which all equal one.
      - The diagonal of a matrix A consists of the entries $\alpha_{0,0}$, $\alpha_{1,1}$, etc. In other words, all elements $\alpha_{i,i}$.
      - Througout this course, we will use the capital letter I to indicate an identity matrix "of appropriate size".

Special Matrices

Algorithm: $[A] := \text{SET\_TO\_IDENTITY}(A)$

Partition $A \rightarrow \left(A_L \,\middle|\, A_R \right)$
where A_L has 0 columns

while $n(A_L) < n(A)$ **do**

Repartition

$$\left(A_L \,\middle|\, A_R \right) \rightarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right)$$
where a_1 has 1 column

$a_1 := e_j$ (Set the current column to the correct unit basis vector)

Continue with

$$\left(A_L \,\middle|\, A_R \right) \leftarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right)$$

endwhile

Special Matrices

Algorithm: $[A] := \text{SET\_TO\_IDENTITY}(A)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$
where A_{TL} is 0×0

while $m(A_{TL}) < m(A)$ **do**

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$
where α_{11} is 1×1

set current column to appropriate unit basis vector

$a_{01} := 0$ set a_{01}'s components to zero

$\alpha_{11} := 1$

$a_{21} := 0$ set a_{21}'s components to zero

Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

endwhile

Special Matrices

- Diagonal Matrices
  - Definition 3.3
    - A matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is said to be diagonal if $\alpha_{i,j} = 0$ for all $i \neq j$ so that

$$A = \begin{pmatrix} \alpha_{0,0} & 0 & \ldots & 0 \\ 0 & \alpha_{1,1} & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \alpha_{n-1,n-1} \end{pmatrix}$$

Special Matrices

Algorithm: $[A] := \text{SET\_TO\_DIAGONAL\_MATRIX}(A, x)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$, $x \rightarrow \left(\begin{array}{c} x_T \\ \hline x_B \end{array}\right)$
where A_{TL} is 0×0, x_T has 0 elements

while $m(A_{TL}) < m(A)$ **do**

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right), \left(\begin{array}{c} x_T \\ \hline x_B \end{array}\right) \rightarrow \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ \hline x_2 \end{array}\right)$$
where α_{11} is 1×1, χ_1 is a scalar

$a_{01} := 0$

$\alpha_{11} := \chi_1$

$a_{21} := 0$

Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right), \left(\begin{array}{c} x_T \\ \hline x_B \end{array}\right) \leftarrow \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ \hline x_2 \end{array}\right)$$

endwhile

Special Matrices

- Triangular Matrices
  - Definition 3.4 (Triangular matrix)
    - A matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is said to be

lower triangular	$\alpha_{i,j} = 0$ if $i < j$	$\begin{pmatrix} \alpha_{0,0} & 0 & \cdots & 0 & 0 \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n-2,0} & \alpha_{n-2,1} & \cdots & \alpha_{n-2,n-2} & 0 \\ \alpha_{n-1,0} & \alpha_{n-1,1} & \cdots & \alpha_{n-1,n-2} & \alpha_{n-1,n-1} \end{pmatrix}$
strictly lower triangular	$\alpha_{i,j} = 0$ if $i \leq j$	$\begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \alpha_{1,0} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n-2,0} & \alpha_{n-2,1} & \cdots & 0 & 0 \\ \alpha_{n-1,0} & \alpha_{n-1,1} & \cdots & \alpha_{n-1,n-2} & 0 \end{pmatrix}$
unit lower triangular	$\alpha_{i,j} = \begin{cases} 0 & \text{if } i < j \\ 1 & \text{if } i = j \end{cases}$	$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ \alpha_{1,0} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n-2,0} & \alpha_{n-2,1} & \cdots & 1 & 0 \\ \alpha_{n-1,0} & \alpha_{n-1,1} & \cdots & \alpha_{n-1,n-2} & 1 \end{pmatrix}$

upper triangular	$\alpha_{i,j} = 0$ if $i > j$	$\begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-2} & \alpha_{0,n-1} \\ 0 & \alpha_{1,1} & \cdots & \alpha_{1,n-2} & \alpha_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \alpha_{n-2,n-2} & \alpha_{n-2,n-1} \\ 0 & 0 & \cdots & 0 & \alpha_{n-1,n-1} \end{pmatrix}$
strictly upper triangular	$\alpha_{i,j} = 0$ if $i \geq j$	$\begin{pmatrix} 0 & \alpha_{0,1} & \cdots & \alpha_{0,n-2} & \alpha_{0,n-1} \\ 0 & 0 & \cdots & \alpha_{1,n-2} & \alpha_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \alpha_{n-2,n-1} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$
unit upper triangular	$\alpha_{i,j} = \begin{cases} 0 & \text{if } i > j \\ 1 & \text{if } i = j \end{cases}$	$\begin{pmatrix} 1 & \alpha_{0,1} & \cdots & \alpha_{0,n-2} & \alpha_{0,n-1} \\ 0 & 1 & \cdots & \alpha_{1,n-2} & \alpha_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \alpha_{n-2,n-1} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$

Special Matrices

Algorithm: $[A] := \text{SET\_TO\_LOWER\_TRIANGULAR\_MATRIX}(A)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$
  where A_{TL} is 0×0
while $m(A_{TL}) < m(A)$ **do**

  Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

  where α_{11} is 1×1

  set the elements of the current column above the diagonal to zero
  $a_{01} := 0$ set a_{01}'s components to zero

  Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

endwhile

Special Matrices

- Transpose Matrix
  - Definition 3.5
    - Let $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times m}$. Then B is said to be the transpose of A if, for $0 \leq i < m$ and $0 \leq i < n$, $\beta_{j,i} = \alpha_{i,j}$. The transpose of a matrix A is denoted by A^T so that $B = A^T$.

Special Matrices

Algorithm: $[B] := \text{TRANSPOSE}(A, B)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_L & A_R \end{array}\right)$, $B \rightarrow \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right)$
  where A_L has 0 columns, B_T has 0 rows
while $n(A_L) < n(A)$ **do**

  Repartition

$$\left(\begin{array}{c|c} A_L & A_R \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_0 & a_1 & A_2 \end{array}\right), \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right) \rightarrow \left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)$$

  where a_1 has 1 column, b_1 has 1 row

  $b_1^T := a_1^T$ (Set the current row of B to the current column of A)

  Continue with

$$\left(\begin{array}{c|c} A_L & A_R \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_0 & a_1 & A_2 \end{array}\right), \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right) \leftarrow \left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)$$

endwhile

Special Matrices

- Symmetric Matrices
  - A matrix $A \in \mathbb{R}^{n \times n}$ is said to be symmetric if $A = A^T$.
  - In other words, if $A \in \mathbb{R}^{n \times n}$ is symmetric, then $\alpha_{i,j} = \alpha_{j,i}$ for all $0 \le i, j < n$. Another way of expressing this is that

$$A = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-2} & \alpha_{0,n-1} \\ \alpha_{0,1} & \alpha_{1,1} & \cdots & \alpha_{1,n-2} & \alpha_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{0,n-2} & \alpha_{1,n-2} & \cdots & \alpha_{n-2,n-2} & \alpha_{n-2,n-1} \\ \alpha_{0,n-1} & \alpha_{1,n-1} & \cdots & \alpha_{n-2,n-1} & \alpha_{n-1,n-1} \end{pmatrix} \qquad A = \begin{pmatrix} \alpha_{0,0} & \alpha_{1,0} & \cdots & \alpha_{n-2,0} & \alpha_{n-1,0} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{n-2,1} & \alpha_{n-1,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n-2,0} & \alpha_{n-2,1} & \cdots & \alpha_{n-2,n-2} & \alpha_{n-1,n-2} \\ \alpha_{n-1,0} & \alpha_{n-1,1} & \cdots & \alpha_{n-1,n-2} & \alpha_{n-1,n-1} \end{pmatrix}.$$

Special Matrices

Algorithm: $[A] :=$ SYMMETRIZE_FROM_LOWER_TRIANGLE(A)

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$

where A_{TL} is 0×0

while $m(A_{TL}) < m(A)$ **do**

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

where α_{11} is 1×1

(set a_{01}'s components to their symmetric parts below the diagonal)

$a_{01} := (a_{10}^T)^T$

Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

endwhile

Operations with Matrices

- Scaling a Matrix
- Adding Matrices

Operations with Matrices

- Scaling a Matrix
  - Theorem 3.6
    - Let $L_A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a linear transformation and, for all $x \in \mathbb{R}^n$, define the function $L_B: \mathbb{R}^n \rightarrow \mathbb{R}^m$ by $L_B(x) = \beta L_A(x)$, where β is a scalar. Then $L_B(x)$ is a linear transformation.
- Adding Matrices

Operations with Matrices

Algorithm: $[A] := \text{SCALE\_MATRIX}(\beta, A)$

Partition $A \rightarrow \left(A_L \,\middle|\, A_R \right)$
where A_L has 0 columns

while $n(A_L) < n(A)$ **do**

Repartition

$$\left(A_L \,\middle|\, A_R \right) \rightarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right)$$
where a_1 has 1 column

$a_1 := \beta a_1$ (Scale the current column of A)

Continue with

$$\left(A_L \,\middle|\, A_R \right) \leftarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right)$$

endwhile

Operations with Matrices

- Adding Matrices

Algorithm: $[A] := \text{ADD\_MATRICES}(A, B)$

Partition $A \rightarrow \left(A_L \,\middle|\, A_R \right)$, $B \rightarrow \left(B_L \,\middle|\, B_R \right)$
where A_L has 0 columns, B_L has 0 columns

while $n(A_L) < n(A)$ **do**

Repartition

$$\left(A_L \,\middle|\, A_R \right) \rightarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right), \left(B_L \,\middle|\, B_R \right) \rightarrow \left(B_0 \,\middle|\, b_1 \,\middle|\, B_2 \right)$$
where a_1 has 1 column, b_1 has 1 column

$a_1 := a_1 + b_1$ (Add the current column of B to the current column of A)

Continue with

$$\left(A_L \,\middle|\, A_R \right) \leftarrow \left(A_0 \,\middle|\, a_1 \,\middle|\, A_2 \right), \left(B_L \,\middle|\, B_R \right) \leftarrow \left(B_0 \,\middle|\, b_1 \,\middle|\, B_2 \right)$$

endwhile

Matrix-Vector Multiplication Algorithms

- Via Dot Products
- Via AXPY Operations
- Compare and Contrast
- Cost of Matrix-Vector Multiplication

Matrix-Vector Multiplication Algorithms

- Via Dot Products

Motivation

Recall that if $y = Ax$, where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, and $y \in \mathbb{R}^m$, then

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

If one looks at a typical row,

$$\alpha_{i,0}\chi_0 + \quad \alpha_{i,1}\chi_1 + \quad \cdots + \quad \alpha_{i,n-1}\chi_{n-1}$$

one notices that this is just the dot product of vectors

$$\widetilde{a}_i = \begin{pmatrix} \alpha_{i,0} \\ \alpha_{i,1} \\ \vdots \\ \alpha_{i,n-1} \end{pmatrix} \quad \text{and} \quad x = \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_n - 1 \end{pmatrix}.$$

In other words, the dot product of the ith row of A, viewed as a column vector, with the vector x, which one can visualize as

$$\begin{pmatrix} \psi_0 \\ \vdots \\ \boxed{\psi_i} \\ \vdots \\ \psi_{m-1} \end{pmatrix} = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,n-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{i,0} & \alpha_{i,1} & \cdots & \alpha_{i,n-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,n-1} \end{pmatrix} \boxed{\begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix}}$$

Matrix-Vector Multiplication Algorithms

- Via Dot Products

Algorithm (traditional notation)

An algorithm for computing $y := Ax + y$ (notice that we add the result of Ax to y) via dot products is given by

$$
\begin{array}{l}
\textbf{for } i = 0, \ldots, m-1 \\
\quad \textbf{for } j = 0, \ldots, n-1 \\
\qquad \psi_i := \psi_i + \alpha_{i,j}\chi_j \\
\quad \textbf{endfor} \\
\textbf{endfor}
\end{array}
$$

Matrix-Vector Multiplication Algorithms

Algorithm: $y := \text{MVMULT\_N\_UNB\_VAR1}(A, x, y)$

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 is $0 \times n$ and y_T is 0×1

while $m(A_T) < m(A)$ **do**

Repartition

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

where a_1 is a row

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

Continue with

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

endwhile

Matrix-Vector Multiplication Algorithms

- Via AXPY Operations

Motivation

Note that, by definition,

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

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

Matrix-Vector Multiplication Algorithms

- Via AXPY Operations

Algorithm (traditional notation)

The above suggests the alternative algorithm for computing $y := Ax + y$ given by

$$
\begin{array}{l}
\textbf{for } j = 0, \ldots, n-1 \\
\quad \textbf{for } i = 0, \ldots, m-1 \\
\qquad \psi_i := \psi_i + \alpha_{i,j}\chi_j \\
\quad \textbf{endfor} \\
\textbf{endfor}
\end{array}
$$

Matrix-Vector Multiplication Algorithms

Algorithm: $y := \text{MVMULT\_N\_UNB\_VAR2}(A, x, y)$

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 $m \times 0$ and x_T is 0×1

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

Repartition

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

where a_1 is a column

$y := \chi_1 a_1 + y$

Continue with

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

endwhile

Matrix-Vector Multiplication Algorithms

- Compare and Contrast

Motivation

It is always useful to compare and contrast different algorithms for the same operation.

Algorithms (traditional notation)

Let us put the two algorithms that compute $y := Ax + y$ via "double nested loops" next to each other:

for $j = 0, \ldots, n-1$	**for** $i = 0, \ldots, m-1$
for $i = 0, \ldots, m-1$	**for** $j = 0, \ldots, n-1$
$\psi_i := \psi_i + \alpha_{i,j}\chi_j$	$\psi_i := \psi_i + \alpha_{i,j}\chi_j$
endfor	**endfor**
endfor	**endfor**

Matrix-Vector Multiplication Algorithms

Algorithm: $y := \text{MVMULT\_N\_UNB\_VAR1}(A, x, y)$

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 is $0 \times n$ and y_T is 0×1

while $m(A_T) < m(A)$ **do**

Repartition

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

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

Continue with

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

endwhile

Algorithm: $y := \text{MVMULT\_N\_UNB\_VAR2}(A, x, y)$

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 $m \times 0$ and x_T is 0×1

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

Repartition

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

$y := \chi_1 a_1 + y$

Continue with

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

endwhile

Matrix-Vector Multiplication Algorithms

- Cost of Matrix-Vector Multiplication

Consider $y := Ax + y$, where $A \in \mathbb{R}^{m \times n}$:

$$y = \left(\begin{array}{c} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{m-1} \end{array} \right) = \left(\begin{array}{ccccc} \alpha_{0,0}\chi_0 + & \alpha_{0,1}\chi_1 + & \cdots + & \alpha_{0,n-1}\chi_{n-1} + & \psi_0 \\ \alpha_{1,0}\chi_0 + & \alpha_{1,1}\chi_1 + & \cdots + & \alpha_{1,n-1}\chi_{n-1} + & \psi_2 \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_{m-1,0}\chi_0 + & \alpha_{m-1,1}\chi_1 + & \cdots + & \alpha_{m-1,n-1}\chi_{n-1} + & \psi_{m-1} \end{array} \right).$$

Questions and Answers