



A System maintenance is scheduled for Wednesday, August 29, 2018 from 14:30-15:30 UTC. Courses might not be available during this time.

Course > Week... > 4.3 M... > 4.3.1 ...

4.3.1 Transpose Matrix-Vector Multiplication

4.3.1 Transpose Matrix-Vector Multiplication

Start of transcript. Skip to the end.

Dr. Robert van de Geijn: Computational scientists

who use linear algebra are typically concerned about two things.

One is performance.

If you spend \$30 million on a supercomputer,

you don't want to only utilize 5% of that computer because of the 0:00 / 5:56 1.0x CC way you

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file



Discussion

Topic: Week 4 / 4.3.1

Hide Discussion

Add a Post

Show all posts by recent activity ▼ There are no posts in this topic yet.

Homework 4.3.1.1

1/1 point (graded)

Algorithm: $y := \text{MVMULT_T_UNB_VAR1}(A, x, y)$

Partition
$$A \to \left(A_L \middle| A_R \right), y \to \left(\frac{y_T}{y_B} \right)$$

where A_L is $m \times 0$ and y_T is 0×1

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

Repartition

$$\left(A_L \middle| A_R \right) \to \left(A_0 \middle| a_1 \middle| A_2 \right), \left(\frac{y_T}{y_B} \right) \to \left(\frac{y_0}{\psi_1} \middle| \frac{y_0}{y_2} \right)$$

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

Continue with

$$\left(A_L \middle| A_R\right) \leftarrow \left(A_0 \middle| a_1 \middle| A_2\right), \left(\frac{y_T}{y_B}\right) \leftarrow \left(\frac{y_0}{\psi_1}\right)$$

endwhile

Algorithm: $y := \text{MVMULT_T_UNB_VAR2}(A, x, y)$

Partition
$$A \to \left(\frac{A_T}{A_B}\right), x \to \left(\frac{x_T}{x_B}\right)$$

where A_T is $0 \times n$ and x_T is 0×1

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

Repartition

$$\left(\frac{A_T}{A_B}\right) \to \left(\frac{A_0}{\frac{a_1^T}{A_2}}\right), \left(\frac{x_T}{x_B}\right) \to \left(\frac{x_0}{\frac{\chi_1}{\chi_2}}\right)$$

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

Continue with

$$\left(\frac{A_T}{A_B}\right) \leftarrow \left(\frac{\frac{A_0}{a_1^T}}{A_2}\right), \left(\frac{x_T}{x_B}\right) \leftarrow \left(\frac{\frac{x_0}{\chi_1}}{\frac{\chi_2}{\chi_2}}\right)$$

endwhile

Write routines

- [y_out] = Mvmult_t_unb_var1(A, x, y)
- [y_out] = Mvmult_t_unb_var2(A, x, y)

that compute $y := A^T x + y$ using the algorithms in this unit.

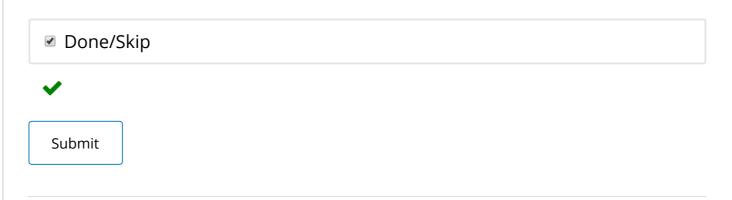
Some links that will come in handy:

- Spark (alternatively, open the file LAFF-2.0xM/Spark/index.html)
- <u>PictureFLAME</u> (alternatively, open the file LAFF-
 - 2.0xM/PictureFLAME/PictureFLAME.html)

You may want to use the following scripts to test your implementations (these should be in your directory LAFF-2.0xM/Programming/Week04/):

test Mvmult t unb var1.m

test Mvmult t unb var2.m



Homework 4.3.1.2

2/2 points (graded)

Implementations achieve better performance (finish faster) if one accesses data consecutively in memory. Now, most scientific computing codes store matrices in "column-major order" which means that the first column of a matrix is stored consecutively in memory, then the second column and so forth. Now, this means that an algorithm that accesses a matrix by columns tends to be faster than an algorithm that accesses a matrix by rows. That, in turn, means that when one is presented with more than one algorithm, one should pick the algorithm that accesses the matrix by columns.

Our FLAME notation makes it easy to recognize algorithms that access the matrix by columns.

ullet For the matrix-vector multiplication of Ax+y, would you recommend the algorithm that uses dot products or the algorithm that uses axpy operations?



ullet For the matrix-vector multiplication of A^Tx+y , would you recommend the algorithm that uses dot products or the algorithm that uses axpy operations?

