+3 for presentation

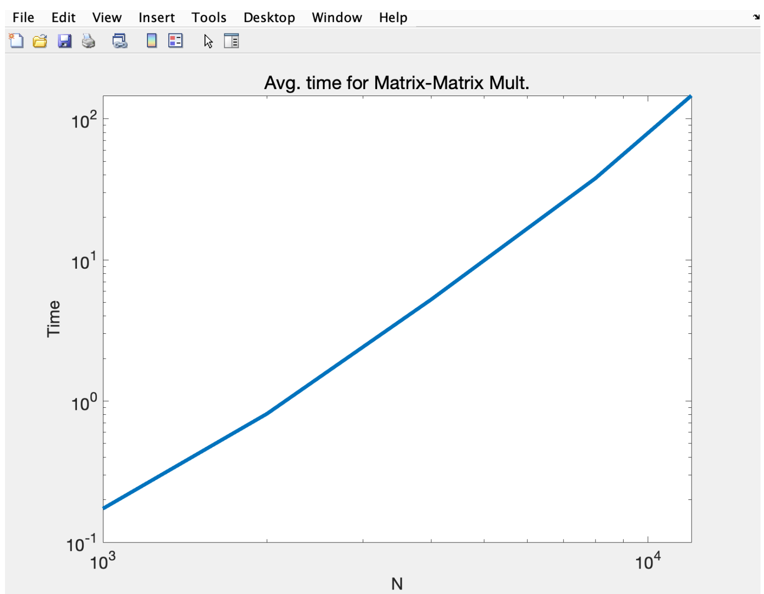
CS375 HW4

Ryan Scherbarth, University of New Mexico

September 2024

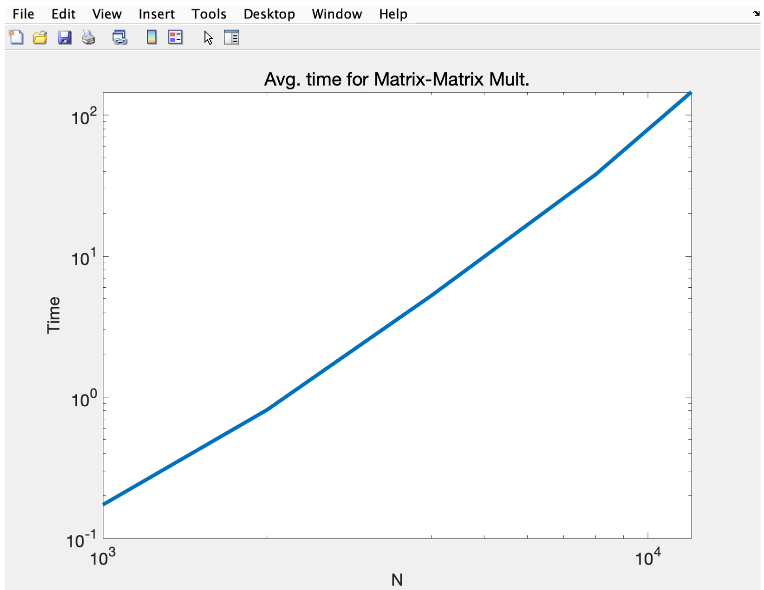
1. FLOPS
   On the course canvas page (attached to this homework assignment), you will find a script test_flops.m that generates a plot for the run-time of a mat-vec operation over different problem sizes.

   - (a) Modify this fie in order to plot a similar timing graph for matrix-matrix multiplication. Make sure to use appropriate axis scaling and readable labels. Use the Matlab command $A * B$ to perform the matrix-matrix multiply where A, B are n x n matrices. Find the exhibited big-Oh notation for the runtime, e.g., $O(n)$, $O(n^2)$, $O(n^3)$, ... Justify your answer from your plot.



This graph shows that as we increase our problem size by one exponential factor, the runtime increases by 3 exponential factors. In other words, our runtime in terms of n with mat-mat multiplication, is $O(n^3)$.

- (b) Repeat the previous problem, but now instead of a single matrix operation ($A * B$), consider multiple mat-mat operations. Time the cost of calculation $A * B + C * D$ for n x n matrices A, B, C, and D, and generate the relevant plot. Does this operation still have the same big-Oh runtime as in part (a)? Explain



Again, we see the same behavior of increasing 3 exponential factors upward within a single factor on the horizontal axis. This is because due to the properties of Big O notation, summing the matrices costs less time than multiplying them, so this time is absorbed into the quicker-growing term. We simplify this operation to still having a running time of $O(n^3)$.

2. Numerical invertibility

- (a) Show that the matrix $A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$ is singular. Find a vector in the null-space of A (i.e., a nonzero vector y such that $Ay = 0$).

  Does the equation $Ax = b$ have no solutions, one solution, or infinitely many solutions if $b = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}$

We can start by finding the determinant of the matrix:

$$\begin{aligned} det(A) &= 1(5*9 - 8*6) - 4(2*9 - 8*3) + 7(2*6 - 5*3) \\ &= 1(-3) - 4(-6) + 7(-3) \\ &= 0 \end{aligned}$$

A matrix with the determinant equal to zero is singular. We can now simplify the matrix to solve;

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4 & 7 & 0 \\ 2 & 5 & 8 & 0 \\ 3 & 6 & 9 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 7 & 1 \\ 0 & -3 & -6 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

Here, we see the bottom row of our matrix simplifies to $0 = 0$. This isn't *necessarily* a bad thing, but in this case when solving for a 3x3 matrix, this results in our matrix becoming under-defined. if we let $b = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}$, the entire matrix still simplifies to $\begin{bmatrix} 1 & 4 & 7 & 1 \\ 0 & -3 & -6 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.
Since the last row is under-determined, we can conclude there are infinitely many solutions due to the introduction of a free variable as we cancel out the final row with $0 = 0$.

- (b) Solve the linear system $Ax = b$ in matlab with a backslash. Does matlab think the matrix is singular? If not, describe why.

Yes, when plugged into matlab to solve with the backslash operator we get a warning that the matrix is singular to working precision, and we get a fault result of $\begin{bmatrix} NaN \\ -Inf \\ Inf \end{bmatrix}$.

- (c) Now define a new problem $Ax = b$ where $\tilde{A} = \frac{1}{10}A$, $\tilde{b} = \frac{1}{10}b$. This problem should have the same solution x, you would encounter the same issue in part (b). Try to solve this problem in matlab using the backslash command. Does matlab think the matrix is singular? If not, describe why or why not.

When solving in matlab we get a differnet error this time, wihle matlab doesn't tell us directly that it is singular, we get a warning that the matrix is "close to singular or badly scaled". Despite this, we get an answer of $\begin{bmatrix} 2.9167 \\ -1.500 \\ 0.05833 \end{bmatrix}$. While we get a different answer this time, it's important to keep in mind that due to the warning, we can pretty much disregard what the number values are in the matrix. The matlab singularity warning tells us this.

why is it not returned as singular?-5

3. Linear Algebra Exercises

   - (a) Prove that if L is lower-triangular and invertible, then the matrix $A = L^{-1}$ is also lower triangular.

A matrix L is lower triangular if all the netries above the main diagonal are zero, i.e. $L_{i,j} = 0$ for $i < j$. Since L is invertible, there exists a matrix $A = L^{-1}$ s.t. $LA = I$, where I is the identity matrix.

The product $LA = I$ for a given nxn matrix;

$$\sum_{k=1}^{n} L_{i,k}A_{k,j} = D_{i,j} \text{ for } 1 \leq i, j \leq n$$

We can now give a proof by induction that this identity will hold true.

Base case, $i = 1$:

$$L_{1,1}A_{1,j} = D_{1,j}$$

Which means $A_{1,1} = \frac{1}{L_{1,1}}$ and $A_{1,j} = 0$ for $j > 1$. So, the first row of A is lower triangular and our base case holds true.

Inductive Hypothesis:
Assume that the base case holds true, we will prove it also holds for $i = m + 1$.

Inductive Step, $i = m + 1$;

$$\sum_{k=1}^{m+1} L_{m+1,k}A_{k,j} = D_{m+1,j}$$

If $j > m+1$ the left size is zero because $L_{m+1,k} = 0$ for $k > (m+1)$ and $A_{k,j} = 0$ for $k \leq m$. Therefore, we can conclude $A_{m+1,j} = 0$ for $j > m + 1$.

We've proven that the case holds true for matrices of size nxn and $n + 1$x$x + 1$, therefore via proof by induction it holds true for all $n > 1$.

   - (b) A square matrix is symmetric if transposing (or reflecting the entries across the diagonal) doesn't change the matrix. Put another way, A is symmetric if the entries obey $a_{i,j} = a_{j,i}$. Prove the following statement or find a counter example: if A and B are both symmetric, then $C = AB$ is also symmetric.

This statement does not hold true. Given $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, b = \begin{bmatrix} 4 & 5 \\ 5 & 6 \end{bmatrix}$

$$C = AB = \begin{bmatrix} 14 & 17 \\ 23 & 28 \end{bmatrix}$$

$$C^T = BA = \begin{bmatrix} 14 & 23 \\ 17 & 28 \end{bmatrix}$$

Since $C \neq C^T$, the statement does not hold true.

4. Matrix Inversion
Consider the system of equations

$$\begin{aligned} x + 2y - z &= 10 \\ 2x - y + z &= 3 \\ 3x - 2y + 2z &= 3 \end{aligned}$$

- (a) Solve this system by hand using techniques from high school algebra and keep track of the "flops" required (anytime you multiply, add, subtract, or divide two numbers).

$$\begin{aligned} (x + 2y - z) + (2x - y + z) &= 10 + 3 \\ 3x + y &= 13 \end{aligned}$$

3 flops

$$\begin{aligned} (2x - y + z) + (3x - 2y + 2z) &= 3 + 3 \\ 5x - 3y + 3z &= 6 \end{aligned}$$

3 flops

$$\begin{aligned} (5x - 3y + 3z) - (x + 2y - z) &= 6 - 10 \\ 4x - 5y + 4z &= -4 \end{aligned}$$

3 flops

$$y = 13 - 3x$$

1 flop

$$\begin{aligned} 4x - 5(13 - 3x) + 4z &= -4 \\ 4x - 65 + 15x + 4z &= -4 \\ 19x + 4z &= 61 \end{aligned}$$

4 flops

$$\begin{aligned} 4z &= 61 - 19x \\ z &= \frac{61 - 19x}{4} \end{aligned}$$

3 flops
So our final solution is $x = 3$, $y = 4$, $z = 1$.
The total number of flops spent on solving the system is 17 flops.

- (b) Now reform the problem as a matrix equation $A\overrightarrow{x} = \overrightarrow{b}$.

$$\begin{bmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 3 & -2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 10 \\ 3 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & -1 & 10 \\ 2 & -1 & 1 & 3 \\ 3 & -2 & 2 & 3 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 & -1 & 10 \\ 0 & -5 & 3 & -17 \\ 3 & -2 & 2 & 3 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 & -1 & 10 \\ 0 & -5 & 3 & -17 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 & -1 & 10 \\ 0 & -5 & 0 & -14 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Now we back substitute to solve;
z:

$$z = 1$$

y:

$$\begin{aligned} -5y &= -14 \\ y &= 4 \end{aligned}$$

x:

$$\begin{aligned} x + 2(4) - 1 &= 10 \\ x + 8 - 1 &= 10 \\ x + 7 &= 10 \\ x &= 3 \end{aligned}$$

So our final solution is $x = 3$, $y = 4$, $z = 1$.

- (c) Calculate the inverse matrix A^{-1} and perform the multiplication $A^{-1}\overrightarrow{b}$. Count the number of flops required and compare with (a).

The inverse of matrix A is:

$$A^{-1} = \begin{bmatrix} \frac{4}{3} & \frac{2}{3} & -1 \\ \frac{5}{3} & \frac{1}{3} & -1 \\ 1 & 1 & -1 \end{bmatrix}$$

Now, calculate $A^{-1}\overrightarrow{b}$:

$$\begin{aligned} A^{-1}\overrightarrow{b} &= \begin{bmatrix} \frac{4}{3} & \frac{2}{3} & -1 \\ \frac{5}{3} & \frac{1}{3} & -1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 10 \\ 3 \\ 3 \end{bmatrix} \\ &= \begin{bmatrix} \frac{4}{3} \times 10 + \frac{2}{3} \times 3 - 1 \times 3 \\ \frac{5}{3} \times 10 + \frac{1}{3} \times 3 - 1 \times 3 \\ 1 \times 10 + 1 \times 3 - 1 \times 3 \end{bmatrix} \\ &= \begin{bmatrix} \frac{40}{3} + 2 - 3 \\ \frac{50}{3} + 1 - 3 \\ 10 + 3 - 3 \end{bmatrix} \\ &= \begin{bmatrix} \frac{40}{3} - 1 \\ \frac{50}{3} - 2 \\ 10 \end{bmatrix} \\ &= \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} \end{aligned}$$

Counting the number of flops:

- Each element requires 2 multiplications and 2 additions/subtractions.
- Total number of operations for each element = 4 flops.
- Total number of flops for the matrix multiplication $= 4 \times 3 = 12$ flops.

Comparing with part (a), we have:

- Part (a) required a total of 17 flops.
- Calculating $A^{-1}\overrightarrow{b}$ requires 12 flops.
- Therefore, solving the system using matrix inversion is more efficient in terms of flops than the method used in part (a).

Need to add in the number of flops required to calculate A_inverse.

-3