Homework 0 [TI8419 - Multilinear Algebra]

Lucas Abdalah

Professors: André Lima e Henrique Goulart

Table of Contents

Problem 1

For randomly generated $\mathbf{A} \in \mathbb{C}^{N \times N}$ and $\mathbf{B} \in \mathbb{C}^{N \times N}$, evaluate the computational performance (run time) of the following matrix inversion formulas:

(a)

Method 1:

$$\left(\mathbf{A}_{N \times N} \otimes \mathbf{B}_{N \times N}\right)^{-1}$$

Method 2:

$$\left(\mathbf{A}_{N \times N}\right)^{-1} \otimes \left(\mathbf{B}_{N \times N}\right)^{-1}$$

For $n \in \{2, 4, 6, 8, 16, 32, 64\}$.

Results

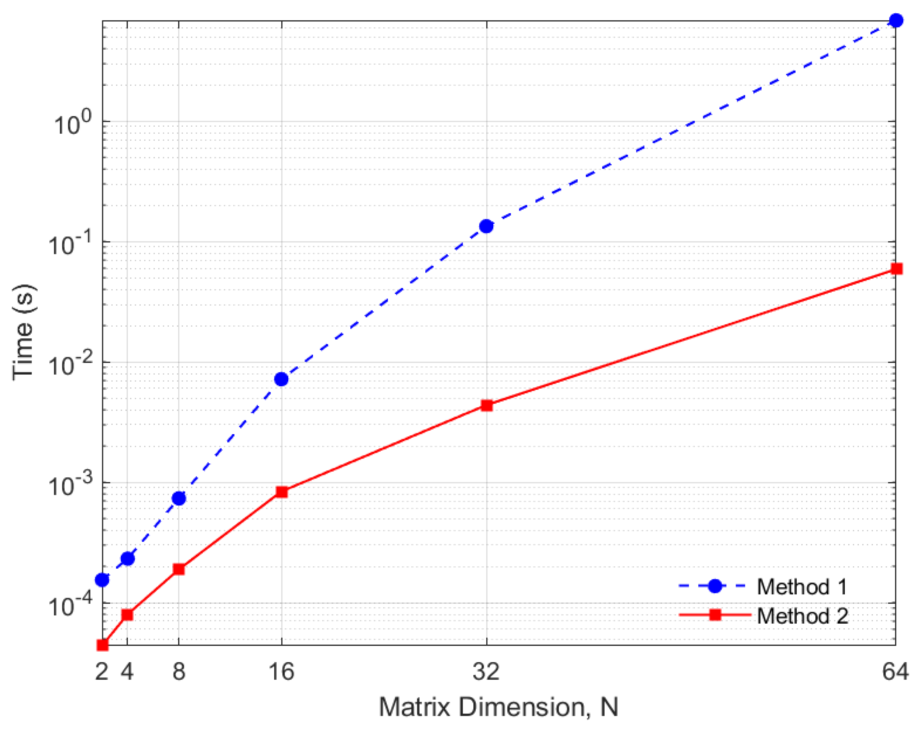
Simulation setup

- 100 Monte Carlo Runs;
- Each Monte Carlo iteration uses a new matrix initialization from a Normal distribution $\mathcal{N}(0, 1)$;
- Compute the mean for each value, for $N = 2, 4, 6, 8, 16, 32, 64$.

Discussion

We can see that for all values of N, the second method outperforms the first. For small values of N, the difference is more subtle, ten times faster. However as the N increases, the performance gap increases up to a hundred times faster.

Problem 1.a script



(b)

Method 1:

$$\left(\mathbf{A}^{(1)}_{N \times N} \otimes \mathbf{A}^{(2)}_{N \times N} \otimes \mathbf{A}^{(3)}_{N \times N} \otimes \cdots \otimes \mathbf{A}^{(K)}_{N \times N}\right)^{-1} = \left(\bigotimes_{k=1}^{K} \mathbf{A}^{(k)}_{N \times N}\right)^{-1}$$

Method 2:

$$\left(\mathbf{A}^{(1)}_{N \times N}\right)^{-1} \otimes \left(\mathbf{A}^{(2)}_{N \times N}\right)^{-1} \otimes \left(\mathbf{A}^{(3)}_{N \times N}\right)^{-1} \otimes \cdots \otimes \left(\mathbf{A}^{(K)}_{N \times N}\right)^{-1} = \bigotimes_{k=1}^{K} \left(\mathbf{A}^{(k)}_{N \times N}\right)^{-1}$$

For $k \in \{2, 4, 6, 8, 10\}$.

Results

Simulation setup

- 200 Monte Carlo Runs;
- Each Monte Carlo iteration uses a new matrix initialization from a Normal distribution $\mathcal{N}(0, 1)$;
- Compute the mean for each value for $N = 2$ and $K = 2, 4, 6, 8, 10$.

Discussion

On the scenario proposed, with $N = 4$, the amount of memory (ram) is up to greater than 64.0 Gb. Since a single complex element requires 16 bytes, the simulation using the homework setup fails from K = 8, since it's required more RAM memory than the available, 19.8 Gb. This value consider 100% of ram use, without taking into count the operational system (OS), backgroud scripts or matlab.

Example

To illustrate, the function kron_dim may be applied for the example with $N = 4$ $k = 7$:

```
Matrix Dimensions: 16384X16384
N of elements: 268435456
Memory use:  4 Gb
```

Since each matrix is 4×4, each Kronnecker product multiplies by 16 the amount of RAM required, hence the matrix product with $K = 8$ leads it to an error.

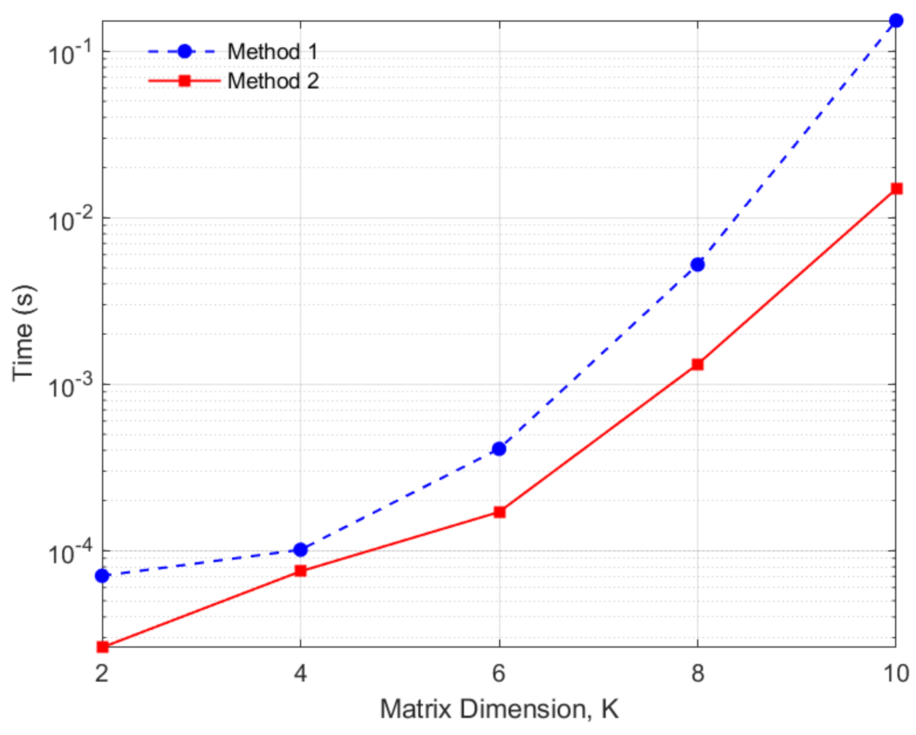
```
Requested 4x16384x4x16384 (64.0GB) array exceeds maximum array size preference (19.8GB). This might cause MATLAB to become unre
```

Finally, we set $N = 2$ for maximum usage when $K = 10$, since it leads to a $2^{10} \times 2^{10}$ matrix, with 1048576 elements and only 16 Mb of ram use.

```
Matrix Dimensions: 1024X1024
N of elements: 1048576
Memory use: 16 Mb
```

We can see that for all values of K, the second method outperforms the first. Both results support the hypothesis that the inversion of smaller matrices in Matlab is much more effective.

Problem 1.b script



Problem 2

Let $\text{eig}(\mathbf{X})$ be the function that returns the matrix $\sum_{K \times K}$ of eigenvalues of $\mathbf{X}$. Show algebraically that $\text{eig}(\mathbf{A} \otimes \mathbf{B}) = \text{eig}(\mathbf{A}) \otimes \text{eig}(\mathbf{B})$.

Hint: Use the property $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$

We write the SVD for each matrix, $\mathbf{A}$ and $\mathbf{B}$, as:

$$\begin{aligned} \mathbf{A} &= \mathbf{U}_A \mathbf{\Sigma}_A \mathbf{V}_A^H \\ \mathbf{B} &= \mathbf{U}_B \mathbf{\Sigma}_B \mathbf{V}_B^H, \end{aligned}$$

We take advantage of the definitions to the equation $\text{eig}(\mathbf{A} \otimes \mathbf{B})$ and using two times the property suggested by the exercise, we have:

$$\begin{aligned} \text{eig}\left(\mathbf{U}_A \mathbf{\Sigma}_A \mathbf{V}_A^H \otimes \mathbf{U}_B \mathbf{\Sigma}_B \mathbf{V}_B^H\right) &= \text{eig}\left[(\mathbf{U}_A \otimes \mathbf{U}_B)(\mathbf{\Sigma}_A \mathbf{V}_A^H \otimes \mathbf{\Sigma}_B \mathbf{V}_B^H)\right] \\ &= \text{eig}\left[(\mathbf{U}_A \otimes \mathbf{U}_B)(\mathbf{\Sigma}_A \otimes \mathbf{\Sigma}_B)(\mathbf{V}_A \otimes \mathbf{V}_B)^H\right] \\ &= \mathbf{\Sigma}_A \otimes \mathbf{\Sigma}_B = \text{eig}(\mathbf{A}) \otimes \text{eig}(\mathbf{B}), \end{aligned}$$

by applying the operator $\text{eig}(\cdot)$ that returns the eigenvalue matrix $\mathbf{\Sigma}_A \otimes \mathbf{\Sigma}_B$.