# Introduction to Scientific Computing

**Indian Institute of Technology, Madras**

## Assignment 4

Maximum Marks: 100

Assigned: April 3, 2024
Deadline: **April 15**, 2024

## General Instructions

- You are expected to use the VM for this assignment. Create a directory in your home directory called `assignment_4`. Use this directory to work on the assignment.
- For each question (for question $i$), create a bash file called `question_i.sh` in the `assignment_4` directory. This bash file should contain the necessary code or commands to solve the respective question.
- We will be using an evaluation script to assess and evaluate your submission. Therefore, kindly ensure that the naming convention (as mentioned in usage section of each question) is strictly adhered to, and that the output which you get from running a script, matches the structure of the sample output.
- For submission, upload the MD5 checksum of the `assignment_4` directory on Moodle along with the report. You can use the following command. Make sure that you are in `HOME` directory for this command to work as intended.

```
find ./assignment_4/* -exec md5sum {} \; | cut -f 1 -d " " | md5sum
```

- After submitting the MD5 checksum on Moodle, **do not** update any file(s). Doing so will change your checksum, and your submission will not be evaluated.
- Every script of yours should output the values in the exact format as given in sample output of the given task. Any deviations would lead to non-evaluation of that part of your answer.
- You are free to read through various resources. However, please ensure that you cite your sources to avoid plagiarism. Any detected instances of plagiarism will result in penalties.
- Please contact your assigned TA for any doubts or queries regarding this assignment.
- The **hard deadline** for this assignment is **11:59 PM** on **April 15**, 2024. Submissions after this deadline will not be evaluated.

---

[30 marks] 1. **Breaking The Spectre:**
**Context:**

- Spectral theorem is a very powerful tool in the field of linear algebra, which applications are diverse.
- Rank-1 decomposition of matrices finds a lot of applications in creation of quantum logic gates and analysis and prediction for a variety of inputs.[1]
- The spectrum of the Hermitian operators used in quantum mechanics like Hamiltonian ($\mathcal{H}$) and momentum operators, gives us the physical observables like energy, momentum.[2]
- Eigenvalue decomposition of matrices using spectral theorem finds a variety of uses in machine learning, for eg. Principal Component Analysis (PCA), Spectral Clustering, etc.[3]
- It also finds use in Harris Corner Detection Algorithm, in the field of Computer Vision, where the magnitude of eigenvalues of the intensity matrix, gives us information of whether a given point in a window of pixels is a corner, edge or a flat patch.[4]

[1]Credits to MA6005
[2]Credits to CY1001
[3]Credits to CS5691
[4]Credits to EE5178

- It's used for converting an ODE to a simple eigenvalue problem, which is much easier to solve.

**Task:**
Your task is to create a script, which takes as an argument, a YAML serialized file, containing Matrix object inside it (consider this matrix as $\boldsymbol{A}$, until this discussion) and gives as output the following values:

[5 marks] (a) The eigenvalues of the matrix $\boldsymbol{A}$,

[5 marks] (b) The unitary matrix $\boldsymbol{U}$ which diagonalizes $\boldsymbol{A}$ (i.e: $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^*$, where $\boldsymbol{D}$ is a diagonal matrix),

[5 marks] (c) The diagonal matrix $\boldsymbol{D}$ resulting from the previous step,

[7 marks] (d) The spectral decomposition of the matrix $\boldsymbol{A}$,

[8 marks] (e) The classification[5] of the matrix $\boldsymbol{A}$ into any one of these special matrices (Hermitian, Unitary, Positively Semidefinite, Positively Definite, or just Normal matrix). (Hint: use step a) [6]

**Usage:**

```
./question_1.sh obj.yml
```

**Sample Input:**

```
$ head -n 5 obj.yml
!!python/object:sympy.matrices.dense.MutableDenseMatrix
_rep: !!python/object/new:sympy.polys.matrices.domainmatrix.DomainMatrix
args:
- 0: &id002
    0: !!python/object/apply:gmpy2.gmpy2.from_binary
```

**Sample Output:**

```
Eigenvalues:
```

$$\begin{bmatrix} 0 & 0 & 3 \end{bmatrix}$$

```
U:
```

$$\begin{bmatrix} 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix}$$

```
D:
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

```
Decomposition:
```

$$\underbrace{3}_{\lambda_1} * \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{P_1} + \underbrace{0}_{\lambda_2} * \underbrace{\begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}}_{P_2}$$

```
Classification:
A is Hermitian, Positively Semidefinite
```

[30 marks] 2. **Optimaze:**
**Context:**

[5] Should be best describing

[6] Note that all the testing of your scripts would be done with normal $3 \times 3$ matrices

- Optimization is at the core of machine learning. Every abstract machine learning problem is converted and formulated as a mathematical problem with the help of statistics and probability, where a particular objective function is to be maximized or minimized, optionally following a given set of constraints.
- This also finds use in Mechanical Sciences, where the strength of a part is to be maximized by adding material to it, but it increases the cost of the part.
- In Chemical Engineering, wherein a continuous process, there needs to be an optimal rate of reactant to be supplied for the reaction to give maximum yield, but not so much for the reactant to be wasted, which lowers its cost-competitiveness, optimization is inevitably used.
- It is a crucial aspect of the market economy, where companies compete to increase profits and reduce costs.

**Task:**
You are given a set of ($x$, $y$, $z$) coordinates of points in `points.csv` (Program files are available in `/var/home/Jan24/assignments/assignment_4`). Your task is to find the optimal equation of the plane, fitting the behavior of the system of points, to the best possible extent.

The equation of the plane is taken to be $ax + by + cz = 1$, for the rest of the discussion.

Here in the following discussion,

1. $\boldsymbol{A}$ refers to matrix A, and $\boldsymbol{b}$ refers to a vector b,
2. The superscript $t$ refers to the $t^{\text{th}}$ time step,
3. $\boldsymbol{\theta}^t$ refers to vector ($a$, $b$, $c$) and $\Delta\boldsymbol{\theta}^t$ refers to vector ($\Delta a$, $\Delta b$, $\Delta c$).

You need to do the following:

[2 marks] (a) Initialise random values to $\boldsymbol{\theta}^0$. Report this value when the program starts executing,

[5 marks] (b) Compute the objective function ($\mathcal{L}$) which is to be minimised. You can use sum of squares of distances of the plane from the points to be that objective (error) function,

[5 marks] (c) Find $\Delta\boldsymbol{\theta}^t$ using Newton's steepest descent method[7]

$$\Delta\boldsymbol{\theta}^t = \left(\boldsymbol{H^{-1}}\right)^{\boldsymbol{t}} \boldsymbol{g^t}$$

where $\boldsymbol{H}$ is the Hessian and $\boldsymbol{g}$ is the Gradient which are computed as

$$H_{ij} = \frac{\partial^2 \mathcal{L}}{\partial\theta_i \partial\theta_j}$$
$$g_i = \frac{\partial \mathcal{L}}{\partial\theta_i},$$

[6 marks] (d) Update the parameters $\boldsymbol{\theta}^t$ using the following update rule,

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \Delta\boldsymbol{\theta}^t$$

[2 marks] (e) Keep track of the error at $t^{\text{th}}$ step. This can be computed in a similar manner as the objective function,

[10 marks] (f) Repeat the following steps until this error is lesser than a threshold, say $\delta$. Then print the optimal parameters with the number of steps (epochs) taken for solution to converge to $\boldsymbol{\theta}^*$.

**Input:**[8]

[7]**Bonus:** This is theoretically the best second order method, which converges to the required solution in less number of iterations, but almost never used. You may put your thoughts on it in your report.

[8]Pretty printed for convenience. Real file is separated with commas only.

```
$ head -n 6 points.csv
x,       y,         z
53.03,   12.059,    -174.943
17.1,    53.707,    93.721
13.289,  49.104,    95.156
-9.41,   34.472,    142.056
-6.761,  14.219,    70.701
```

and so on ...

**Usage:**

```
./question_2.sh points.csv
```

**Sample Output:**

```
Initialised: 0 0 0
Optimal: 2.001 3.000 1.002
Epochs: 15
```

[40 marks] 3. **Reflection:**

- Write a LaTeX report (named `<roll_number>_A3.pdf`), which contains your observations and the overall conclusion you got from doing.
- Try not to repeat the same point multiple times in the report.
- Make use of math expressions effectively.
- As and when possible, write your points in ordered or unordered bullet points. Images of intermediate steps are appreciated.
- Remember that the more details you project in the report, the more marks you would be reflected with.

[20 marks] (a) Task 1

[20 marks] (b) Task 2