

# EE4013 Assignment-1

Shaik Mastan Vali - EE18BTECH11039

Download all the codes from

[https://github.com/Mastan1301/EE4013/tree/main/assignment-1\\_v2/codes](https://github.com/Mastan1301/EE4013/tree/main/assignment-1_v2/codes)

and latex-tikz codes from

[https://github.com/Mastan1301/EE4013/tree/main/assignment-1\\_v2](https://github.com/Mastan1301/EE4013/tree/main/assignment-1_v2)

## 1 PROBLEM

Consider the following matrix:

$$R = \begin{bmatrix} 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \\ 1 & 5 & 25 & 125 \end{bmatrix}$$

Calculate the absolute value of the Eigen values of  $R$ .

## 2 SOLUTION-1

We use the following property -

$$\prod_{i=0}^{k-1} \lambda_i = \det(R)$$

where  $\lambda_i$  are the Eigen values and  $\det(\cdot)$  is the determinant operator.

To find the determinant, we use the following recurrence relation.

$$\det(R, n) = \sum_{j=0}^{n-1} (-1)^j \times R[0][j] \times \det(\text{cof}(R, 0, j), n-1)$$

where  $\text{cof}(i, j)$  is the cofactor matrix of the position  $(i, j)$  in the matrix  $R$ .

The recursion, for our example, is given in the flow chart.

To understand the process, consider  $C_1 = \text{cof}(R, 0, 0)$ .

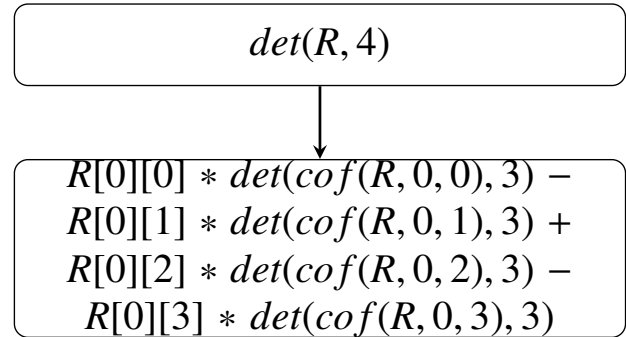


Fig. 0: The flow of recursion

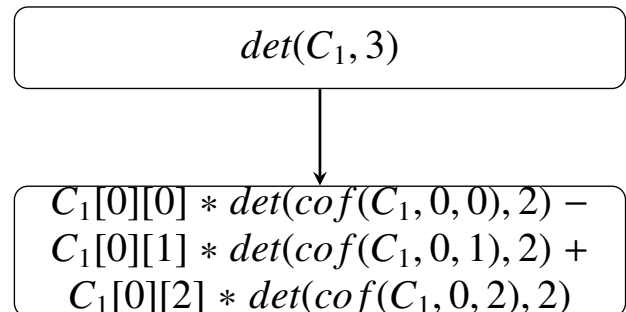


Fig. 0: A  $3 \times 3$  sub-problem

Now, consider  $C_2 = \text{cof}(C_1, 0, 0)$ .

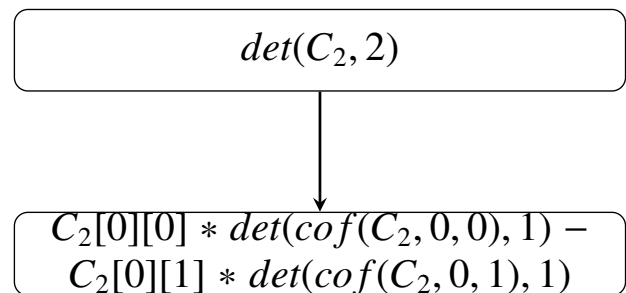


Fig. 0: A  $2 \times 2$  sub-problem

The determinant of a  $1 \times 1$  matrix is the value of the element itself. We use this property as the base case.

The absolute value of the determinant of the given matrix is 12. Hence, the answer is 12.

The algorithm is in the following file -

[https://github.com/Mastan1301/EE4013/tree/main/assignment-1\\_v2/codes/sol-1.c](https://github.com/Mastan1301/EE4013/tree/main/assignment-1_v2/codes/sol-1.c)

### 3 COMPLEXITY OF THE ALGORITHM

At each stage of recursion, there are  $O(n)$  recursive calls. In each of these function calls, we compute the co-factor matrix in  $O(n^2)$  time. So, the total time complexity is  $O(n^3)$ .

### 4 SOLUTION-2

In this solution, we use the QR-algorithm to compute the Eigen values. The algorithm is as follows-

- Apply the QR decomposition so that

$$X = Q_0 R_0$$

where  $Q_0$  is an orthogonal matrix and  $R_0$  is an upper-triangular matrix.

- Compute

$$E_0 = Q_0^{-1} X Q_0 = Q_0^T X Q_0$$

Note that the Eigen values of  $X$  and  $E_0$  are the same.

- Next, we decompose the  $E_0$  and repeat this procedure till we obtain a triangular matrix. The Eigen values of a triangular matrix are given by the diagonal entries of the matrix.

We compute the QR decomposition using the Householder transformation. Multiplying a given vector  $a$ , for example the first column of matrix  $A$ , with the Householder matrix  $H$ , which is given as

$$H = I - \frac{2}{u^T u} u u^T$$

reflects  $a$  about a plane given by its normal vector  $u$ . When the normal vector of the plane  $u$  is given as

$$u = a - \|a\|_2 e_1$$

then the transformation reflects  $a$  onto the first standard basis vector

$$e_1 = \begin{bmatrix} 1 & 0 & 0 & \dots \end{bmatrix}^T$$

which means that all entries but the first become zero. To avoid numerical cancellation errors, we should take the opposite sign of  $a_1$  :

$$u = a + \text{sign}(a_1) \|a\|_2 e_1$$

and normalize with respect to the first element:

$$v = \frac{u}{u_1}$$

The equation for  $H$  thus becomes:

$$H = I - \frac{2}{v^T v} v v^T$$

or, in another form

$$H = I - \beta v v^T$$

with

$$\beta = \frac{2}{v^T v}$$

Applying  $H$  on  $a$  then gives

$$H a = -\text{sign}(a_1) \|a\|_2 e_1$$

and applying  $H$  on the matrix  $A$  zeroes all subdiagonal elements of the first column:

$$H_1 A = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & * & * \\ 0 & * & * \end{pmatrix}$$

(denoted \*), not on the whole second column of  $A$ . To get  $H_2$ , we then embed the new  $H$  into an  $m \times n$  identity:

$$H_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & H & \\ 0 & & \end{pmatrix}$$

This is how we can, column by column, remove all subdiagonal elements of  $A$  and thus transform it into  $R$ .  $H_n \dots H_3 H_2 H_1 A = R$  The product of all the Householder matrices  $H$ , for every column, in reverse order, will then yield the orthogonal matrix  $Q$ .

$$H_1 H_2 H_3 \dots H_n = Q$$

Using this algorithm, the obtained Eigen values are 139.410, 5.003, 0.555, 0.0311. The absolute value of the product is 12.

The algorithm is in the following file-

[https://github.com/Mastan1301/EE4013/tree/main/assignment-1\\_v2/codes/sol-2.c](https://github.com/Mastan1301/EE4013/tree/main/assignment-1_v2/codes/sol-2.c)

The utility functions are in the following header file-

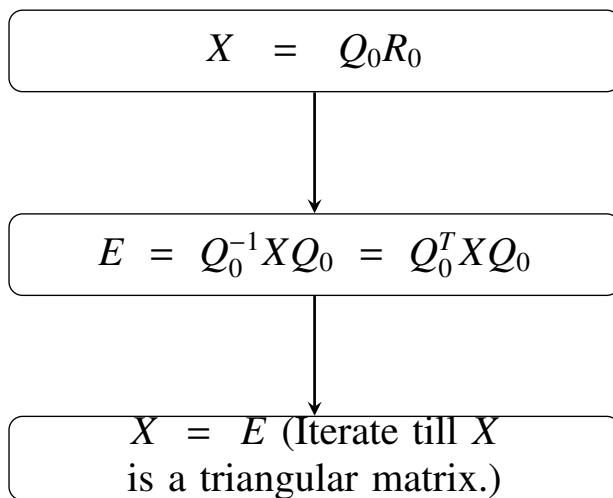


Fig. 0: The QR algorithm

[https://github.com/Mastan1301/EE4013/tree/main/assignment-1\\_v2/codes/utlis.h](https://github.com/Mastan1301/EE4013/tree/main/assignment-1_v2/codes/utlis.h)

Compile the code using-

```
gcc sol-2.c -o sol-2 -lm
```

Run the code using-

```
./sol-2
```

## 5 COMPLEXITY OF THE QR ALGORITHM

The QR decomposition step takes  $O(n^3)$  computations. The matrix convergence is achieved in  $O(|\lambda_i/\lambda_j|)$  steps. Hence the total time complexity is  $O(|\lambda_i/\lambda_j| \times n^3)$ .