

Midterm Project

LINEAR ALGEBRA

Askhatkyzy Togzhan
Aristanov Ali

Kazakh-British
Technical
University



Matrix Operations in Python

**The program calculates the determinant and inverse of a given square matrix.
It utilizes basic linear algebra concepts and matrix manipulation techniques.**

GETTING THE MATRIX

- Function: `get_matrix()`
- Prompts the user for the size of the matrix.
- Collects matrix elements row by row.
- Stores the matrix in a list of lists format.

```
def get_matrix():  
    n = int(input("enter the size of matrix: "))  
    matrix = []  
    for i in range(n):  
        row = list(map(float, input(f"enter the row {i + 1} sep by a space: ").split()))  
        matrix.append(row)  
    return matrix
```

DETERMINANT CALCULATION

- Function: determinant(matrix)
- Base case for 2x2 matrices:
- Calculates determinant using the formula: $ad - bc$.
- For larger matrices:
- Uses recursion and the cofactor expansion method.
- Computes determinants of submatrices.

```
def determinant(matrix):  
    if len(matrix) == 2:  
        return matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0]  
  
    det = 0  
    for c in range(len(matrix)):  
        sub_matrix = [row[:c] + row[c + 1:] for row in matrix[1:]]  
        det += ((-1) ** c) * matrix[0][c] * determinant(sub_matrix)  
  
    return det
```

INVERSE CALCULATION

- Function: `inverse(matrix)`
- Checks the determinant; if zero, returns non-invertible.
- Creates an identity matrix of the same size.
- Applies row operations to convert the matrix into the identity matrix while modifying the identity matrix to get the inverse.

```

def inverse(matrix):
    n = len(matrix)
    det = determinant(matrix)

    if det == 0:
        return None #if the determinant is 0, the matrix is irreversible

    #a single matrix of the same size
    identity_matrix = [[float(i == j) for j in range(n)] for i in range(n)]

    #copy the original matrix so as not to change the original matrix during the algorithm. Instead, all changes are made to its copy
    mat = [row[:] for row in matrix]

    for i in range(n):
        #if the leading element is zero, we change the row to another one where the leading element is not zero
        if mat[i][i] == 0:
            for k in range(i + 1, n):
                if mat[k][i] != 0:
                    swap_rows(mat, i, k)
                    swap_rows(identity_matrix, i, k)
                    break
            else:
                return None #if the row for the permutation couldn't be found, the matrix is irreversible

        # Normalize the leading element to 1 by dividing the row.
        # This is essential for converting the matrix to the identity matrix
        # and for finding the inverse matrix.

        factor = mat[i][i]
        for j in range(n):
            mat[i][j] /= factor
            identity_matrix[i][j] /= factor

        # Process rows below and above the leading element.
        # This step creates zeros in the current column, simplifying the matrix.

        for k in range(n):
            if i != k:
                factor = mat[k][i]
                for j in range(n):
                    mat[k][j] -= factor * mat[i][j]
                    identity_matrix[k][j] -= factor * identity_matrix[i][j]

    return identity_matrix

```

MATRIX OUTPUT

- Function: print_matrix(matrix)
- Prints the matrix in a formatted way.
- Ensures that the output is readable with two decimal points.

```
def print_matrix(matrix):  
    for row in matrix:  
        print(" ".join(map(lambda x: f"{x:.2f}", row)))  
  
def main():  
    print("what do you want to find")  
    print("1 - determinant")  
    print("2 - inverse matrix")  
    choice = input("enter your choice (1 or 2): ")  
  
    matrix = get_matrix()  
  
    if choice == '1':  
        det = determinant(matrix)  
        print(f"determinant of the matrix: {det:.2f}")  
    elif choice == '2':  
        inverse_matrix = inverse(matrix)  
        if inverse_matrix:  
            print("inverse matrix:")  
            print_matrix(inverse_matrix)  
        else:  
            print("the matrix is non-invertible.")  
    else:  
        print("invalid choice. Please select 1 or 2.")  
  
main()
```

USAGE EXAMPLE

```
what do you want to find?  
1 - determinant  
2 - inverse matrix  
enter your choice (1 or 2): 1  
enter the size of matrix: 4  
enter the row 1 sep by a space: 1 2 3 4  
enter the row 2 sep by a space: 2 3 4 1  
enter the row 3 sep by a space: 3 4 1 2  
enter the row 4 sep by a space: 4 1 2 3  
determinant of the matrix: 160.00
```

```
what do you want to find?  
1 - determinant  
2 - inverse matrix  
enter your choice (1 or 2): 2  
enter the size of matrix: 3  
enter the row 1 sep by a space: 1 2 -1  
enter the row 2 sep by a space: -2 0 1  
enter the row 3 sep by a space: 1 -1 0  
inverse matrix:  
1.00 1.00 2.00  
1.00 1.00 1.00  
2.00 3.00 4.00  
determinant of the matrix: 1.00
```


ERROR HANDLING

- Error 1: Zero Division Error
- Cause: Occurs when the leading element of a row is zero during matrix inversion.
- Solution:
- Implemented a row-swapping function `swap_rows()` to handle cases where the leading element is zero.
- If no suitable row is found, the matrix is declared non-invertible.

```
def swap_rows(mat, row1, row2):  
    #a function for rearranging two rows in a matrix if the leading element is zero  
    mat[row1], mat[row2] = mat[row2], mat[row1]
```

ERROR HANDLING

- Error 2: Invalid Input
- Cause: User may input non-numeric values or mismatched matrix dimensions.
- Solution:
- Input is parsed using float() and handled carefully.
- Ensured that the matrix is square by requesting size n x n.

```
enter the row 1 sep by a space: 1 2
enter the row 2 sep by a space: 4 5 6
enter the row 3 sep by a space: 7
Traceback (most recent call last):
  File "c:\Users\ASUS\Desktop\python2\mid\full_without_inter.py", line 94, in <module>
    main()
  File "c:\Users\ASUS\Desktop\python2\mid\full_without_inter.py", line 85, in main
    inverse_matrix = inverse(matrix)
                      ~~~~~^
File "c:\Users\ASUS\Desktop\python2\mid\full_without_inter.py", line 26, in inverse
    det = determinant(matrix)
          ~~~~~^
File "c:\Users\ASUS\Desktop\python2\mid\full_without_inter.py", line 16, in determinant
    det += ((-1) ** c) * matrix[0][c] * determinant(sub_matrix)
              ~~~~~^
File "c:\Users\ASUS\Desktop\python2\mid\full_without_inter.py", line 11, in determinant
    return matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0]
           ~~~~~^
IndexError: list index out of range
PS C:\Users\ASUS\Desktop\python2> █
```

```
enter your choice (1 or 2): 2
enter the size of matrix: 3
enter the row 1 sep by a space: 1 2 three
Traceback (most recent call last):
  File "c:\Users\ASUS\Desktop\python2\mid\full_without_inter.py", line 94, in <module>
    main()
  File "c:\Users\ASUS\Desktop\python2\mid\full_without_inter.py", line 79, in main
    matrix = get_matrix()
              ~~~~~^
File "c:\Users\ASUS\Desktop\python2\mid\full_without_inter.py", line 5, in get_matrix
    row = list(map(float, input(f"enter the row {i + 1} sep by a space: ").split()))
              ~~~~~^
ValueError: could not convert string to float: 'three'
PS C:\Users\ASUS\Desktop\python2> █
```

ERROR HANDLING

Error 3: Non-Invertible Matrix

- Cause: Matrix with a zero determinant cannot be inverted.

Solution:

- The program checks the determinant before attempting to compute the inverse.
- If the determinant is zero, it prints "The matrix is non-invertible."

```
what do you want to find?
1 - determinant
2 - inverse matrix
enter your choice (1 or 2): 2
enter the size of matrix: 3
enter the row 1 sep by a space: 1 2 3
enter the row 2 sep by a space: 4 5 6
enter the row 3 sep by a space: 7 8 9
the matrix is non-invertible.
PS C:\Users\ASUS\Desktop\python2>
```

ERROR HANDLING

Error 4: Invalid Menu Selection

- Cause: User may enter an invalid option (neither 1 nor 2).

Solution:

- Implemented a check in the main menu to handle incorrect inputs.
- Prompts the user to select either 1 (Determinant) or 2 (Inverse).

```
what do you want to find?
1 - determinant
2 - inverse matrix
enter your choice (1 or 2): 5
enter the size of matrix: 3
enter the row 1 sep by a space: 1 2 3
enter the row 2 sep by a space: 4 5 6
enter the row 3 sep by a space: 9 8 7
invalid choice. Please select 1 or 2.
PS C:\Users\ASUS\Desktop\python2>
```

AND FINALLY, ALWAYS BE POSITIVE LIKE ALI.

детерминант табу 22:16

тоже функция емеспа по идее 22:16

Togzhan ➔ Togzhan

у нас 20 строк

бызде зато качественный код 22:16

Thanks for your attention!