

README: How to Use the Matrix Class

September 19, 2024

1 Introduction

This project provides a class for working with matrices, including basic operations such as addition, subtraction, multiplication, and transposition. This document explains how to use the `Matrix` class and lists the available functions.

2 Usage

2.1 Step 1: Include Necessary Files

To use the `Matrix` class, you need to include the following files in your `main.cpp`:

```
#include "Matrix.h"
#include "Matrix.cpp"
```

These files contain the definitions and implementations of all the functions for working with matrices.

2.2 Step 2: Creating a Matrix

To create a matrix, call the constructor of the `Matrix` class, passing the number of rows and columns:

```
Matrix matrix1(3, 3); // Creating a 3x3 matrix
```

2.3 Step 3: Setting Matrix Values

To set the values of a matrix, you can use the `Set_values_from_vector` method. This method takes a 2D vector of values:

```
std::vector<std::vector<double>> values = {
    {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9}
};
matrix1.Set_values_from_vector(values);
```

2.4 Step 4: Basic Operations

The `Matrix` class supports the following basic operations:

- `Matrix operator+(Matrix& other)`: Adds two matrices.
- `Matrix operator-(Matrix& other)`: Subtracts one matrix from another.
- `Matrix operator*(Matrix& other)`: Multiplies two matrices.
- `void Transpose()`: Transposes the matrix.
- `double Determinant()`: Calculates the determinant (only for square matrices).
- `double Trace()`: Calculates the trace (sum of the diagonal elements) of the matrix (only for square matrices).

Examples:

```
// Multiplying two matrices
Matrix matrix2(3, 3);
Matrix result = matrix1 * matrix2;

// Transposing a matrix
matrix1.Transpose();
```

3 Additional Features

- The matrix class also supports adding and subtracting scalars:

```
Matrix matrix3 = matrix1 + 5.0; // Adds a scalar to each element
```

- The `Set_values_manually` method allows you to input matrix values manually from the console:

```
matrix1.Set_values_manually();
```

4 Conclusion

The `Matrix` class provides convenient methods for working with matrices. You can perform basic mathematical operations such as addition, subtraction, multiplication, transposition, and calculate the determinant and trace of the matrix. To use the class, simply include `Matrix.h` and make use of the available functions.