User

I compiled a cpp file like this in msys:
g++ -I /mingw64/include/eigen3 sample.cpp -o sample
Then I ran ./sample. It yielded output:

Here is mat3i:
-16343 10116 -4906
2083 2785 12974
-10050 -660 10578
The sum of its coefficients is 6577

Source is:
#include <iostream>
#include <Eigen/Dense>

int main() {
// Create a 3x3 matrix of integers.
Eigen::Matrix3i mat3i = Eigen::Matrix3i::Random();

// Print the matrix to the console.

std::cout << "Here is mat3i:\n" << mat3i << std::endl;

// Compute and print the sum of its coefficients.

std::cout << "The sum of its coefficients is " << mat3i.sum() << std::endl;

return 0;

}
Can you explain the output and where eigen has been applied?

Bing

Sure, I'd be happy to explain!

The Eigen library is a high-level C++ library of template headers for linear algebra, matrix and vector operations, numerical solvers and related algorithms. In your code, Eigen is used to create and manipulate a 3x3 matrix of integers.

Let's break down the code:

1. `Eigen::Matrix3i mat3i = Eigen::Matrix3i::Random();`
   * Here, `Eigen::Matrix3i` is a type provided by the Eigen library. It represents a 3x3 matrix of integers.
   * `Eigen::Matrix3i::Random()` generates a 3x3 matrix with random coefficients. The `Random()` function is a static member of `Matrix3i` that returns a randomly initialized matrix.
2. Here, `Eigen::Matrix3i` is a type provided by the Eigen library. It represents a 3x3 matrix of integers.
3. `Eigen::Matrix3i::Random()` generates a 3x3 matrix with random coefficients. The `Random()` function is a static member of `Matrix3i` that returns a randomly initialized matrix.
4. `std::cout << "Here is mat3i:\n" << mat3i << std::endl;`
   * This line prints the string "Here is mat3i:" followed by the contents of the matrix `mat3i`. The `<<` operator is overloaded by Eigen to provide pretty-printing of matrices.
5. This line prints the string "Here is mat3i:" followed by the contents of the matrix `mat3i`. The `<<` operator is overloaded by Eigen to provide pretty-printing of matrices.
6. `std::cout << "The sum of its coefficients is " << mat3i.sum() << std::endl;`
   * This line prints the string "The sum of its coefficients is " followed by the sum of all coefficients in the matrix. The `sum()` function is provided by Eigen and calculates the sum of all elements in the matrix.
7. This line prints the string "The sum of its coefficients is " followed by the sum of all coefficients in the matrix. The `sum()` function is provided by Eigen and calculates the sum of all elements in the matrix.

The output you see is the result of these operations. The 3x3 matrix printed is the randomly generated matrix, and the sum is calculated from this matrix. The specific numbers you see will vary each time you run the program because Random() generates different numbers each time.

User

$ ./sample
Here is mat3i:
-16343 10116 -4906
2083 2785 12974
-10050 -660 10578
The sum of its coefficients is 6577
(.venv)
j1304@dhawal-laptop MSYS ~/Downloads
$ ./sample
Here is mat3i:
-16343 10116 -4906
2083 2785 12974
-10050 -660 10578
The sum of its coefficients is 6577

As you can see above, I have run the exe file twice bit randomization is not there in matrix. All numbers are repeated.

Bing

The Eigen library's Random() function generates pseudo-random numbers. However, it seems like the random number generator is not being properly seeded, which is why you're seeing the same numbers each time you run your program.

In C++, random number generators typically need to be seeded with a unique value (like the current time) to ensure different sequences of random numbers. Without a unique seed, the generator will produce the same sequence of numbers each time the program runs.

Unfortunately, Eigen's Random() function doesn't provide a way to seed the random number generator. If you need different random matrices each time you run your program, you might need to use a different method to generate random numbers.

One possible solution is to use C++11's random number generation facilities in combination with Eigen. Here's an example:

#include <random>

#include <Eigen/Dense>

int main() {

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<> dis(-10000, 10000);

Eigen::Matrix3i mat3i;

for(int i=0; i<mat3i.rows(); ++i)

for(int j=0; j<mat3i.cols(); ++j)

mat3i(i,j) = dis(gen);

std::cout << "Here is mat3i:\n" << mat3i << std::endl;

std::cout << "The sum of its coefficients is " << mat3i.sum() << std::endl;

return 0;

}

This code uses C++11's <random> library to generate random integers, which are then placed into an Eigen matrix. The std::random\_device and std::mt19937 objects are used to seed the random number generator. This should give you a different matrix each time you run your program.

User

errors in compilation of above code:
sample.cpp: In function 'int main()':
sample.cpp:14:10: error: 'cout' is not a member of 'std'
14 | std::cout << "Here is mat3i:\n" << mat3i << std::endl;
| ^~~~
sample.cpp:3:1: note: 'std::cout' is defined in header '<iostream>'; did you forget to '#include <iostream>'?
2 | #include <Eigen/Dense>
+++ |+#include <iostream>
3 |
sample.cpp:15:10: error: 'cout' is not a member of 'std'
15 | std::cout << "The sum of its coefficients is " << mat3i.sum() << std::endl;
| ^~~~
sample.cpp:15:10: note: 'std::cout' is defined in header '<iostream>'; did you forget to '#include <iostream>'?

Bing

I apologize for the oversight. The error messages indicate that the <iostream> library, which contains the definition for std::cout, is not included in your code. Here's the corrected code:

#include <random>

#include <Eigen/Dense>

#include <iostream>

int main() {

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<> dis(-10000, 10000);

Eigen::Matrix3i mat3i;

for(int i=0; i<mat3i.rows(); ++i)

for(int j=0; j<mat3i.cols(); ++j)

mat3i(i,j) = dis(gen);

std::cout << "Here is mat3i:\n" << mat3i << std::endl;

std::cout << "The sum of its coefficients is " << mat3i.sum() << std::endl;

return 0;

}

This should resolve the compilation errors. Please try compiling and running this updated code. Let me know if you encounter any other issues! 😊

User

In the output I see:

Here is mat3i:
6505 -8021 9843
-1282 3054 -2096
-7581 -1894 -4509
The sum of its coefficients is -5981

What does last line mean?

Bing

The last line of the output, "The sum of its coefficients is -5981", is showing the sum of all the elements (coefficients) in the matrix.

In this case, if you add up all the numbers in the matrix:

6505 -8021 9843

-1282 3054 -2096

-7581 -1894 -4509

you will get -5981. This is calculated by the mat3i.sum() function in your code. The .sum() function is a method provided by Eigen that calculates the sum of all elements in the matrix. So, it's adding up all the numbers in your 3x3 matrix and printing out the result.