
Eigen ile Matris ve Vektör Aritmetiği

1. Eigen ile Matris ve Vektör Aritmetiği

Eigen ile matrislerde ve vektörlerde çeşitli işlemler yapılabilir. Bunlar basitçe $*$, $+$, $-$ vb. işlemler olabilir. Eigen C++ operatörlerinin bir kısmını overload etmiştir.

1.1. Toplama ve Çıkarma işlemleri

Toplanacak matris ya da vektörler aynı satır ve sütun sayısına sahip olmak zorundadır. Aynı zamanda aynı veri türünü tutmak zorundadırlar.

Aşağıdaki işlemler yapılabilir:

$a + b$

$a - b$

$-a$

$a += b$

$a -= b$

Örnek kod:

```
#include <iostream>
#include <Eigen>

using namespace Eigen;
using namespace std;

int main()
{
    matrix<int, 2, 2> a;
    a << 1,2,
        3,4;

    matrix<int, 2, 2> b;
```

```
b << 2,3,
    3,1;

cout << " a + b " << a + b << endl;
cout << " a - b " << a - b << endl;
cout << " -a = " << -a << endl;

return 0;
}
```

1.2. Matrisin Transpozu ve Konjuge Matris

Matrisin transpozu `.transpose()` , konjugesini `.conjugate()` ve adjointini `.adjoint()` fonksiyonları çağırılarak bulunabilir.

Gerçek sayılar için konjuge matris olmayacağı için değişiklik yapmaz ve adjoint transpoza eşittir.

Örnek kod:

```
Matrixxcf a = Matrixxcf::Random(2,2);
cout << "Here is the matrix a\n" << a << endl;
cout << "Here is the matrix a^T\n" << a.transpose() << endl;
cout << "Here is the conjugate of a\n" << a.conjugate() << endl;
cout << "Here is the matrix a^*\n" << a.adjoint() << endl;
```

Kodun çıktısı

```
Here is the matrix a
(-0.211,0.68) (-0.605,0.823)
(0.597,0.566) (0.536,-0.33)
Here is the matrix a^T
(-0.211,0.68) (0.597,0.566)
(-0.605,0.823) (0.536,-0.33)
Here is the conjugate of a
(-0.211,-0.68) (-0.605,-0.823)
(0.597,-0.566) (0.536,0.33)
Here is the matrix a^*
(-0.211,-0.68) (0.597,-0.566)
(-0.605,-0.823) (0.536,0.33)
```

Uyarı:

Bir matrisin transpuzunu alıp o matrise atama yapmak, hatalı bir sonuç almaya sebep olur.

```
Matrix2i a; a << 1, 2, 3, 4;
cout << "Here is the matrix a:\n" << a << endl;
a = a.transpose(); // !!! do NOT do this !!!
cout << "and the result of the aliasing effect:\n" << a << endl;
```

```
Here is the matrix a:
1 2
3 4
and the result of the aliasing effect:
1 2
2 4
```

Bu işlem `.transposeInPlace()` ile yapılmalıdır.

```
Matrixxf a(2,3); a << 1, 2, 3, 4, 5, 6;
cout << "Here is the initial matrix a:\n" << a << endl;
a.transposeInPlace();
cout << "and after being transposed:\n" << a << endl;
```

Kodun çıktısı:

```
Here is the initial matrix a:
1 2 3
4 5 6
and after being transposed:
1 4
2 5
3 6
```

1.3. Matris-Matris ve Matris-Vektör çarpımı

Kullanılabilecek operatörler

$a * b$ $a *= b$

```
#include <Eigen>
#include <iostream>
```

```

using namespace Eigen;
int main()
{
    Matrix2d mat;
    mat << 1, 2,
          3, 4;
    Vector2d u(-1,1), v(2,0);
    std::cout << "Here is mat*mat:\n" << mat*mat << std::endl;
    std::cout << "Here is mat*u:\n" << mat*u << std::endl;
    std::cout << "Here is u^T*mat:\n" << u.transpose()*mat << std::endl;
    std::cout << "Here is u^T*v:\n" << u.transpose()*v << std::endl;
    std::cout << "Here is u*v^T:\n" << u*v.transpose() << std::endl;
    std::cout << "Let's multiply mat by itself" << std::endl;
    mat = mat*mat;
    std::cout << "Now mat is mat:\n" << mat << std::endl;
}

```

Kodun Çıktısı

```

Here is mat*mat:
 7 10
15 22
Here is mat*u:
 1
 1
Here is u^T*mat:
 2 2
Here is u^T*v:
-2
Here is u*v^T:
-2 -0
 2 0
Let's multiply mat by itself
Now mat is mat:
 7 10
15 22

```

1.4. Basit aritmetik indirgeme operatörleri

```

.sum() Matris elemanlarının toplamını döndürür
.prod() Matris elemanlarının çarpımını döndürür

```

```
.mean() Matris elemanlarının ortalamasını döndürür
.minCoeff() Matrisin en küçük elemanını döndürür
.maxCoeff() Matrisin en büyük elemanını döndürür
.trace() Matrisin köşegen elemanlarının toplamını döndürür.
mat.diagonal().sum() ile aynı sonucu döndürür.
```

`minCoeff()` ve `maxCoeff()` fonksiyonlarının koordinat döndüren versiyonları da vardır. Dönüş değeri için argüman alırlar.

Örnek:

```
Matrix3f m = Matrix3f::Random();
std::ptrdiff_t i, j;
float minOfM = m.minCoeff(&i,&j);
cout << "Here is the matrix m:\n" << m << endl;
cout << "Its minimum coefficient (" << minOfM
    << ") is at position (" << i << "," << j << ")\n\n";
RowVector4f v = RowVector4f::Random();
int maxOfV = v.maxCoeff(&i);
cout << "Here is the vector v: " << v << endl;
cout << "Its maximum coefficient (" << maxOfV
    << ") is at position " << i << endl;
```

Programın çıktısı

```
Here is the matrix m:
 0.68  0.597 -0.33
-0.211 0.823 0.536
 0.566 -0.605 -0.444
Its minimum coefficient (-0.605) is at position (2,1)

Here is the vector v:  1  0  3 -3
Its maximum coefficient (3) is at position 2
```

1.5. İşlemlerin Geçerliliği

Eigen hataları mümkün oldukça derleme sırasında kontrol eder. Hatalar bazen korkutucu ve uzun gözükebilir. Eigen hatanın önemli kısmını büyük harflerle yazar.

örnek:

```
Matrix3f m;
```

```
Vector4f v;  
v = m*v;          // Compile-time error:  
YOU_MIXED_MATRICES_OF_DIFFERENT_SIZES
```