

Modern cpp 30 lectures — linear algebra

Friday, December 25, 2020

10:07 AM

Armadillo

假设对一个向量进行旋转

$$P = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$P' = R \cdot P$$

现代 C++ 的写法:

```
#include <armadillo>
```

```
#include <cmath>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{  
    arma::vec pos{1.0, 0.0};
```

代表位置的向量.

```
    auto& pi = arma::datum::pi;
```

```
    double angle = pi/2;
```

```
    arma::mat rot = {  
        {cos(angle), -sin(angle)},  
        {sin(angle), cos(angle)}};
```

旋转矩阵

```
    cout << "Current position:\n"
```

```
        << pos;
```

```
    cout << "Rotating "
```

```
        << angle * 180/pi
```

```
        << " deg\n";
```

```
    arma::vec new_pos = rot * pos;
```

```
    cout << "New position:\n"
```

```
        << new_pos;
```

```
}
```

Output:

Current position:

1.000

0

Rotating 90 deg

New position:

6.1232e-17

1.0000e+00

- 所有的 Armadillo 的类型和函数都在 arma 空间名下.
- Armadillo 在 arma::datum 下定义了 pi, e 等常量.
- vec 是向量类型, mat 是矩阵类型, 这两个类型是 Col<double> 和 Mat<double> 的缩写别名.
- Armadillo 支持使用 C++11 的列表初始化语法来初始化对象.
- Armadillo 支持使用流来输出对象.

表达式模板

```
arma::vec new_pos = rot * pos;
```

没有使用 auto 声明, 部分原因是 rot * pos 的类型不是 vec, 而是

```
const Glue<Mat<double>, Col<double>, glue-times>
```

结果是一个表达式, 而并没有进行实际的计算, 如果用 auto, 每次输出这个结果, 都会重新进行一次矩阵的乘法, 用 arma::vec 接收, 构造时就直接进行了计算, 存储了表达式的结果.