

# STL迭代器和算法

从 3.4 版本开始，Eigen 的密集矩阵和数组提供了 STL 兼容的迭代器。如下所示，这使得它们与 range-for-loops 和 STL 算法自然兼容。

## 迭代一维数组和向量

任何密集的一维表达式都会公开这对 `begin()/end()` 方法来迭代它们。

这直接为循环启用 c++11 范围：

例子：

```
1 VectorXi v = VectorXi::Random(4);
2 cout << "Here is the vector v:\n";
3 for(auto x : v) cout << x << " ";
4 cout << "\n";
```

输出：

```
1 Here is the vector v:
2 7 -2 6 6
```

一维表达式也可以很容易地传递给 STL 算法：

例子：

```
1 Array4i v = Array4i::Random().abs();
2 cout << "Here is the initial vector v:\n" << v.transpose() << "\n";
3 std::sort(v.begin(), v.end());
4 cout << "Here is the sorted vector v:\n" << v.transpose() << "\n";
```

输出：

```
1 Here is the initial vector v:
2 7 2 6 6
3 Here is the sorted vector v:
4 2 6 6 7
```

与类似 `std::vector`，一维表达式也公开了这对 `cbegin()/cend()` 方法来方便地获取非常量对象上的常量迭代器。

## 迭代二维数组和矩阵的系数

STL 迭代器本质上是为迭代一维结构而设计的。这就是 `begin()/end()` 为 2D 表达式禁用方法的原因。通过通过以下方式创建一维线性视图仍然可以轻松完成对二维表达式的所有系数的迭代 `reshaped()`：

例子：

```

1 Matrix2i A = Matrix2i::Random();
2 cout << "Here are the coeffs of the 2x2 matrix A:\n";
3 for(auto x : A.resaped())
4     cout << x << " ";
5 cout << "\n";

```

输出:

```

1 Here are the coeffs of the 2x2 matrix A:
2 7 -2 6 6

```

## 迭代二维数组和矩阵的行或列

也可以在二维表达式的行或列上获取迭代器。这些可以通过 `rowwise()` 和 `colwise()` 代理获得。这是对矩阵的每一行进行排序的示例:

例子:

```

1 ArrayXXi A = ArrayXXi::Random(4,4).abs();
2 cout << "Here is the initial matrix A:\n" << A << "\n";
3 for(auto row : A.rowwise())
4     std::sort(row.begin(), row.end());
5 cout << "Here is the sorted matrix A:\n" << A << "\n";

```

输出:

```

1 Here is the initial matrix A:
2 7 9 5 3
3 2 6 1 0
4 6 3 0 9
5 6 6 3 9
6 Here is the sorted matrix A:
7 3 5 7 9
8 0 1 2 6
9 0 3 6 9
10 3 6 6 9

```