

# 将 Eigen 对象按值传递给函数

[密集矩阵和数组操作](#)»[对齐问题](#)

在 C++ 中按值传递对象几乎总是一个非常糟糕的主意，因为这意味着无用的副本，而应该通过引用传递它们。

对于 Eigen，这一点更为重要：按值传递[固定大小的可矢量化 Eigen 对象](#)不仅效率低下，而且可能是非法的或使您的程序崩溃！原因是这些 Eigen 对象具有按值传递时不受尊重的对齐修饰符。

例如，像这样的函数，v 是按值传递的：

```
1 void my_function( Eigen::Vector2d v);
```

需要重写如下，通过 v const 引用传递：

```
1 void my_function( const Eigen::Vector2d & v);
```

同样，如果您有一个具有 Eigen 对象作为成员类：

```
1 struct Foo
2 {
3     Eigen::Vector2d v;
4 };
5 void my_function(Foo v);
```

这个函数也需要像这样重写：

```
1 void my_function( const Foo& v);
```

请注意，另一方面，按值返回对象的函数没有问题。