

# 复现matlab中的fft函数

## tips

1. 大家提前下载好matlab2021破解版(建议)或者临时使用在线matlab进行后续测试
2. 了解基本的Eigen数据类型，实数复数、向量矩阵等。如Eigen::VectorXd是列向量，Eigen::RowVectorXd才是行向量。
3. matlab中fft的帮助文档：<https://www.mathworks.com/help/matlab/ref/fft.html>
4. 利用多态，不使用模版函数（为了兼容性），实现图中所示功能，入参X、出参Y均使用Eigen类型而非std::vector容器，函数内部会涉及到Eigen类型与std::vector互相转换的过程。

### fft

快速傅里叶变换

### 语法

```
Y = fft(X)
Y = fft(X,n)
Y = fft(X,n,dim)
```

### 说明

Y = fft(X) 使用快速傅里叶变换 (FFT) 算法计算 X 的离散傅里叶变换 (DFT)。Y 与 X 的大小相同。

- 如果 X 是向量，则 fft(X) 返回该向量的傅里叶变换。
- 如果 X 是矩阵，则 fft(X) 将 X 的各列视为向量，并返回每列的傅里叶变换。
- 如果 X 是一个多维数组，则 fft(X) 将沿大小不等于 1 的第一个数组维度的值视为向量，并返回每个向量的傅里叶变换。

Y = fft(X,n) 返回 n 点 DFT。

- 如果 X 是向量且 X 的长度小于 n，则为 X 补上尾零以达到长度 n。
- 如果 X 是向量且 X 的长度大于 n，则对 X 进行截断以达到长度 n。
- 如果 X 是矩阵，则每列的处理与在向量情况下相同。
- 如果 X 为多维数组，则大小不等于 1 的第一个数组维度的处理与在向量情况下相同。

Y = fft(X,n,dim) 返回沿维度 dim 的傅里叶变换。例如，如果 X 是矩阵，则 fft(X,n,2) 返回每行的 n 点傅里叶变换。

### 5. 如果三种fft实现

```
Y = fft(X)
Y = fft(X,n)
Y = fft(X,n,dim)
```

类型过多，每种选择一种(矩阵或者向量)实现即可

6. 对于c++实现的函数与matlab的测试结果误差最好在千分之一以下
7. 采用头文件声明与cpp函数定义分离的形式
8. 函数大致结构如下，要求在命名空间内定义函数以防止命名冲突，函数入参在适当时候选择传引用以减小开销

```
#include <Eigen/Dense>
#include <unsupported/Eigen/FFT>
namespace FFTLibrary {
    Eigen::MatrixXcd fft(...) {
        Eigen::FFT<double> fft;
        ...;
    }
}
```