1. **实验环境**

开发环境：MATLAB R2022a

所调函数：conv 卷积函数，fft 快速傅里叶变换，ifft 快速傅里叶逆变换

1. **实验目的**
   1. 编程部分：

通过多重循环的方法，手写一个傅里叶变换及傅里叶逆变换

* 1. 分析对比部分：

对比你使用的平台下的快速傅里叶变化的库函数；比较对于特定长度序列，他们的时间效率。可以多尝试不同长度进行分析

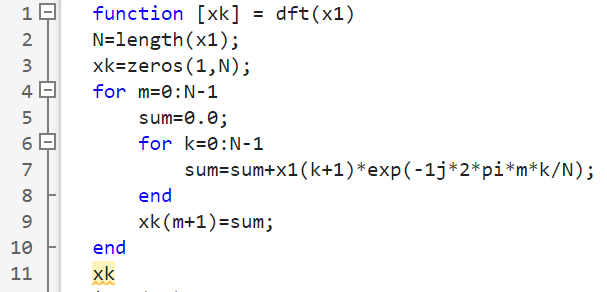
* 1. 验证卷积-傅里叶变化的关系

即对于时域的卷积等价于频域上的乘法；(请通过设计一个滤波的方法进行验证具体方法见频谱分析)

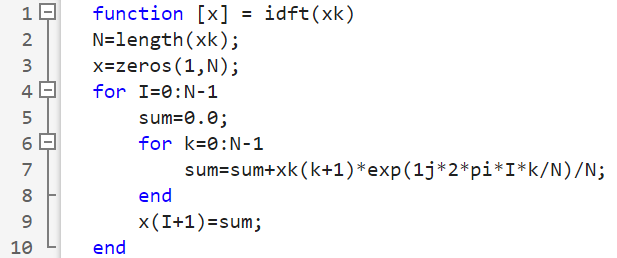
* 1. 频谱分析

验证对于有限长序列的频谱分析的方法画出输入序列的傅里叶频谱（谱线），并对经典的信号序列（例如特定频率正弦信号)进行频谱分析（参照频谱分析章节）

1. **实验内容**
2. 手写傅里叶变换及傅里叶逆变换
   1. dft



* 1. idft：

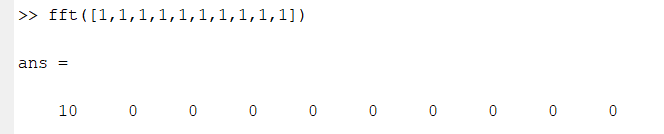


* 1. 测试运行：

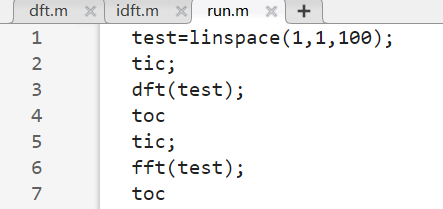
测试数据：[1,1,1,1,1,1,1,1,1,1]。执行dft后结果xk如下，后将xk执行idft，发现可以恢复回原序列：

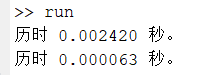


通过fft验证发现dft结果正确：



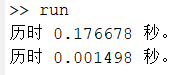
1. 分析对比
   1. 通过linspace函数构造序列并使用tic&toc计时进行测试，测试代码如下：



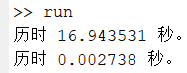


可知在长度为100的情况下，fft比所写dft快38.4倍。

* 1. 多次改变序列长度，数据如下：
     1. 序列长1000



* + 1. 序列长10000

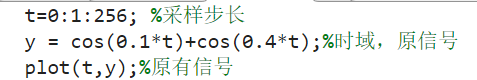


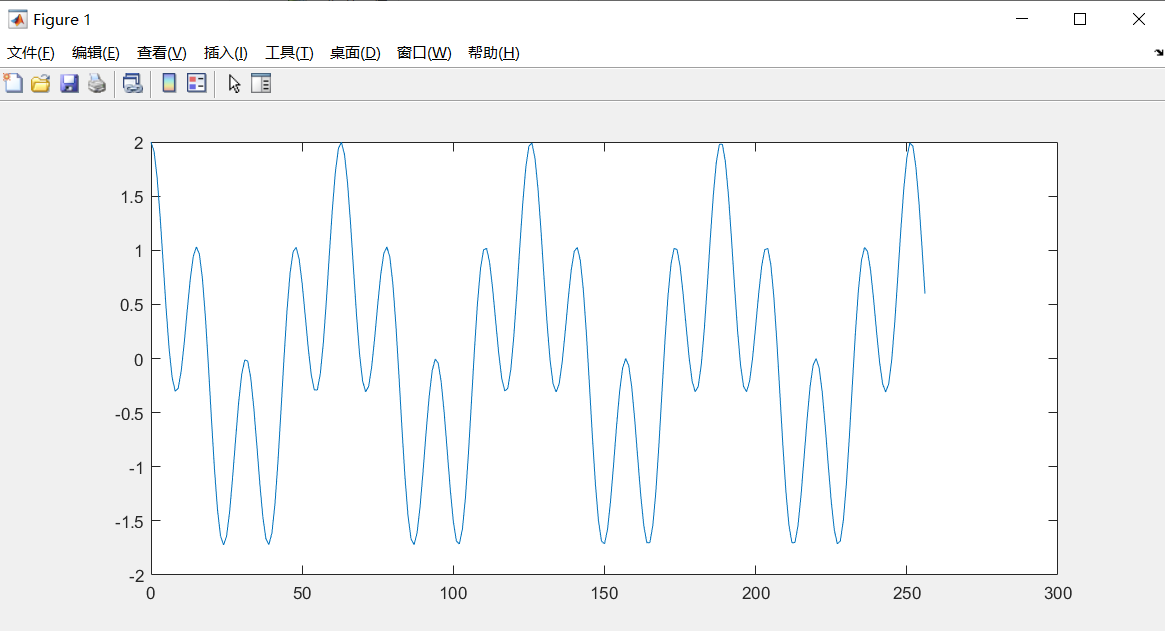
* + 1. 多次改变序列长度进行统计，得到以下结果

|  |  |  |  |
| --- | --- | --- | --- |
| 序列长度 | Dft所用时间/s | Fft所用时间/s | 时间比 |
| 100 | 0.002420 | 0.000063 | 38.4127 |
| 1000 | 0.176678 | 0.001498 | 117.9426 |
| 2500 | 1.074187 | 0.001347 | 797.4662 |
| 5000 | 4.337479 | 0.001772 | 2447.787 |
| 7500 | 9.603992 | 0.001861 | 5160.662 |
| 10000 | 16.943531 | 0.002738 | 6188.287 |
| 50000 | 433.586940 | 0.005831 | 74358.93 |

Dft通过二重循环进行累加，所用时间呈2次幂上升，而fft上升幅度较小。

1. 通过滤波验证卷积、傅里叶变化
   1. 选取课上ppt中的滤波例子进行编程验证，输入信号为，设置采样频率fs=256进行采样形成待操作的初始序列：

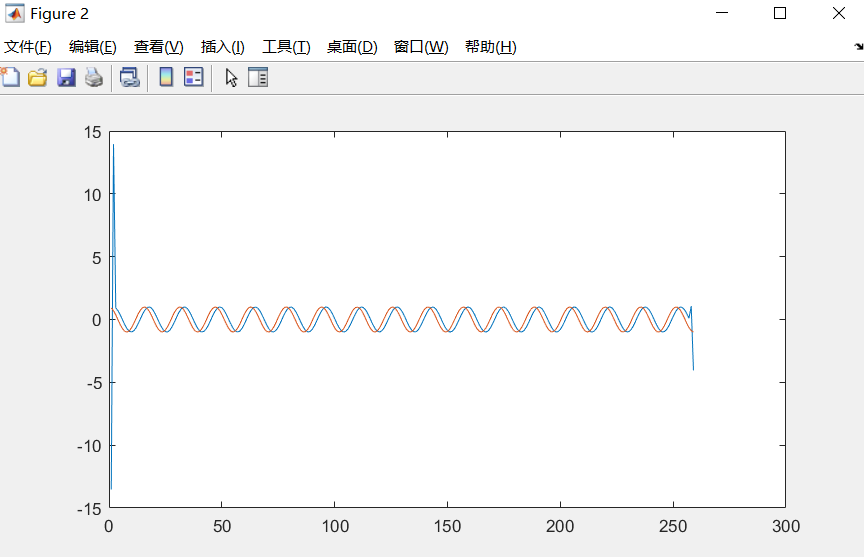




* 1. 时域卷积部分（希望过滤cos0.4t，留下cos0.1t）

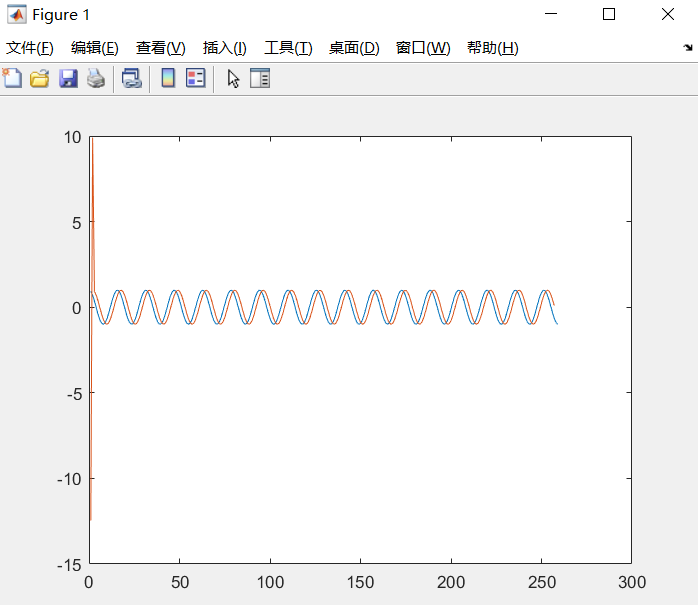
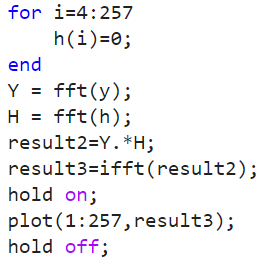
初始化序列h为课上算出的卷积核，h=[-6.76195,13.456335,-6.76195]，将原序列y与该序列h进行卷积，得到result1。result1应该过滤掉了cos0.4t，只剩cos0.1t，故另外设置y1=cos0.1t，将卷积结果和y1画在同一张图中，代码及结果如下（橘色线为y1=cos0.1t，蓝色线为卷积结果result1）：





* 1. 频域乘法部分

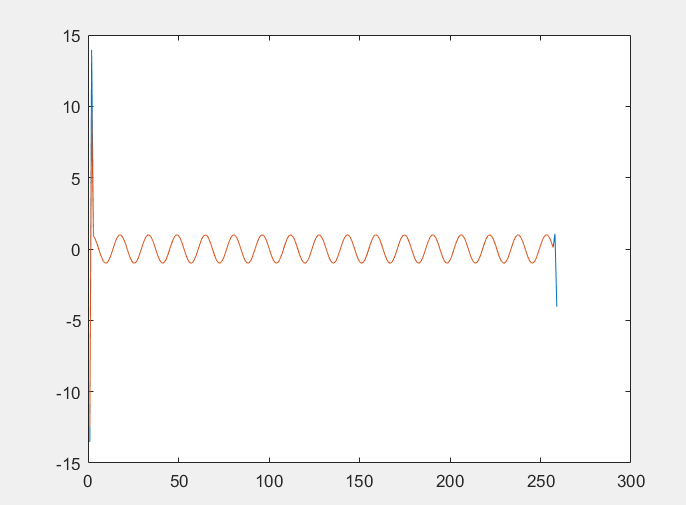
首先将两个序列y、h通过fft转换到频域，分别对应为Y和H，计算Y与H的乘积，并将乘积做傅里叶逆变换得到result3，代码及结果如下（蓝色线为y1=cos0.1x，橘色线为频域乘积结果result3）：



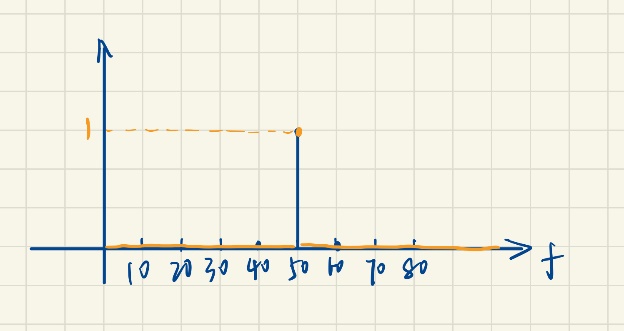
注意：由于y与h的序列长度不同，傅里叶结果不能直接对位相乘，所以将y和h看作无限长序列，其他值即为0，所以将h进行补零操作，使两个序列等长，后进行乘法操作。

* 1. 验证时域的卷积等价于频域上的乘法

将result1和result3和理论结果y1画在同一张图里，发现两条线重合，成功验证：

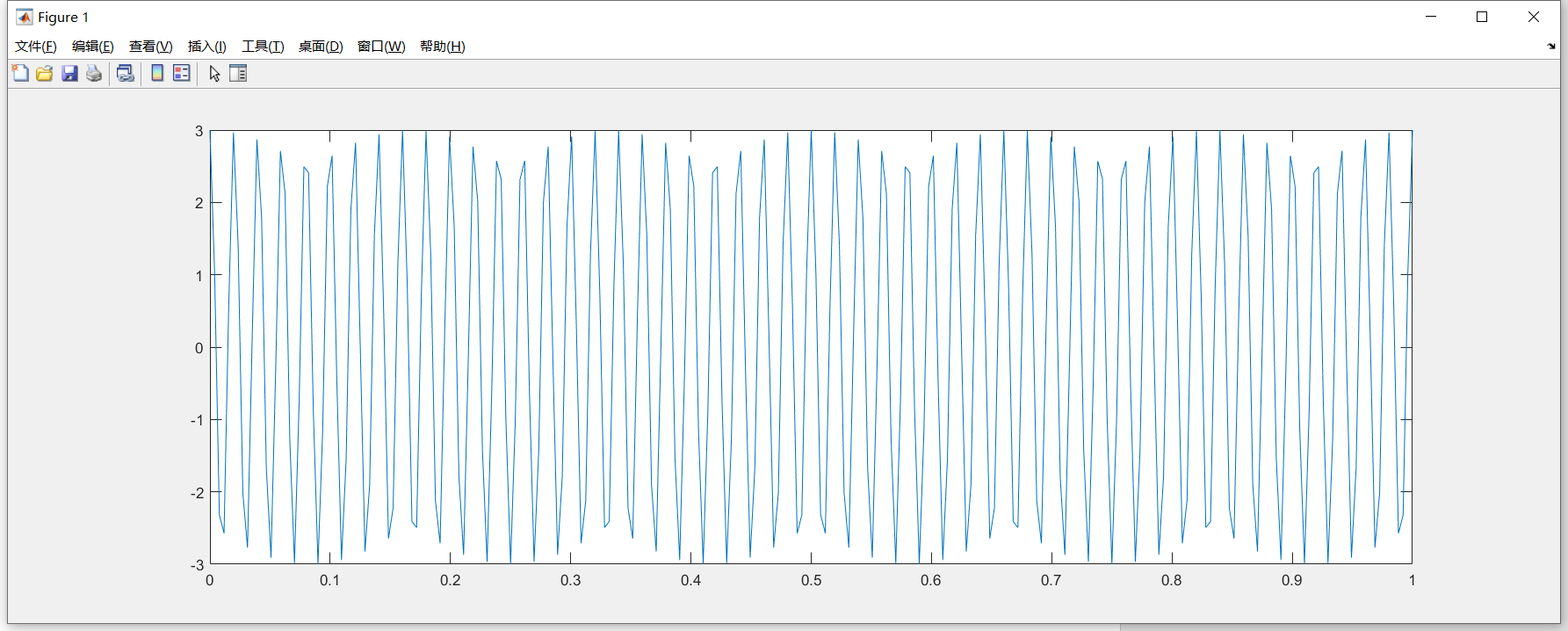


1. 对经典信号cos10πt进行频谱分析
   1. 理想频域图（仅在频率为50时不为0，其余频率连续均为0）



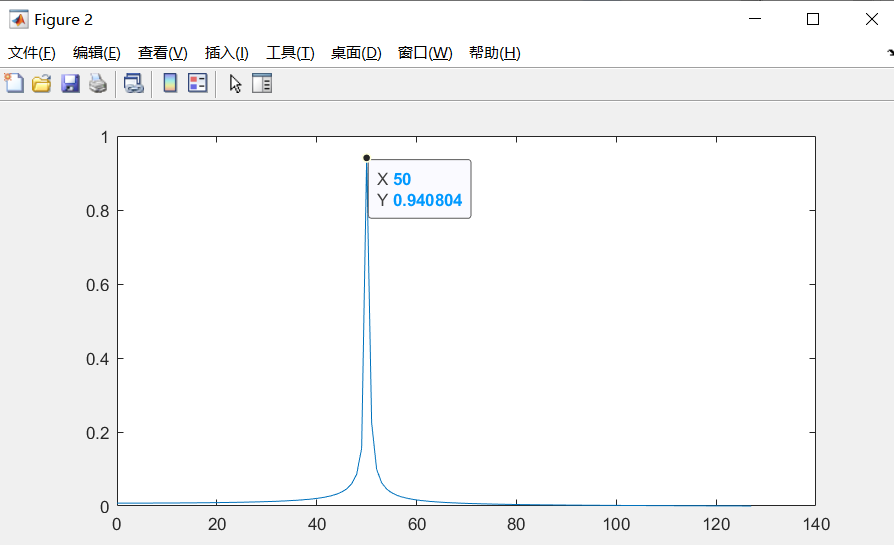
* 1. 验证过程

选取256个点，并以fs=256的采样频率进行采样，构建有限长序列，得到信号图像如下：



通过fft实现傅里叶变换，根据资料所得频谱的幅值大小与选取的N有关，且变换后幅值的大小需要除以后才是真实的幅值。

有效频率仅在范围内，所以通过plot(f(1:N/2),abs(Y(1:N/2)))只显示有效频率范围：



仅在频率为50时有峰值，与原信号cos100πt相符合。

代码如下：

