1 习题

1.1 离散傅里叶变换对（10分）

若常数项包含在DFT里:

DFT: 

Inverse DFT: 

则有：

可求原函数的均值A，再调用函数接口求得F(0,0)的值，

若，则可确认常数项在DFT中。

同理：

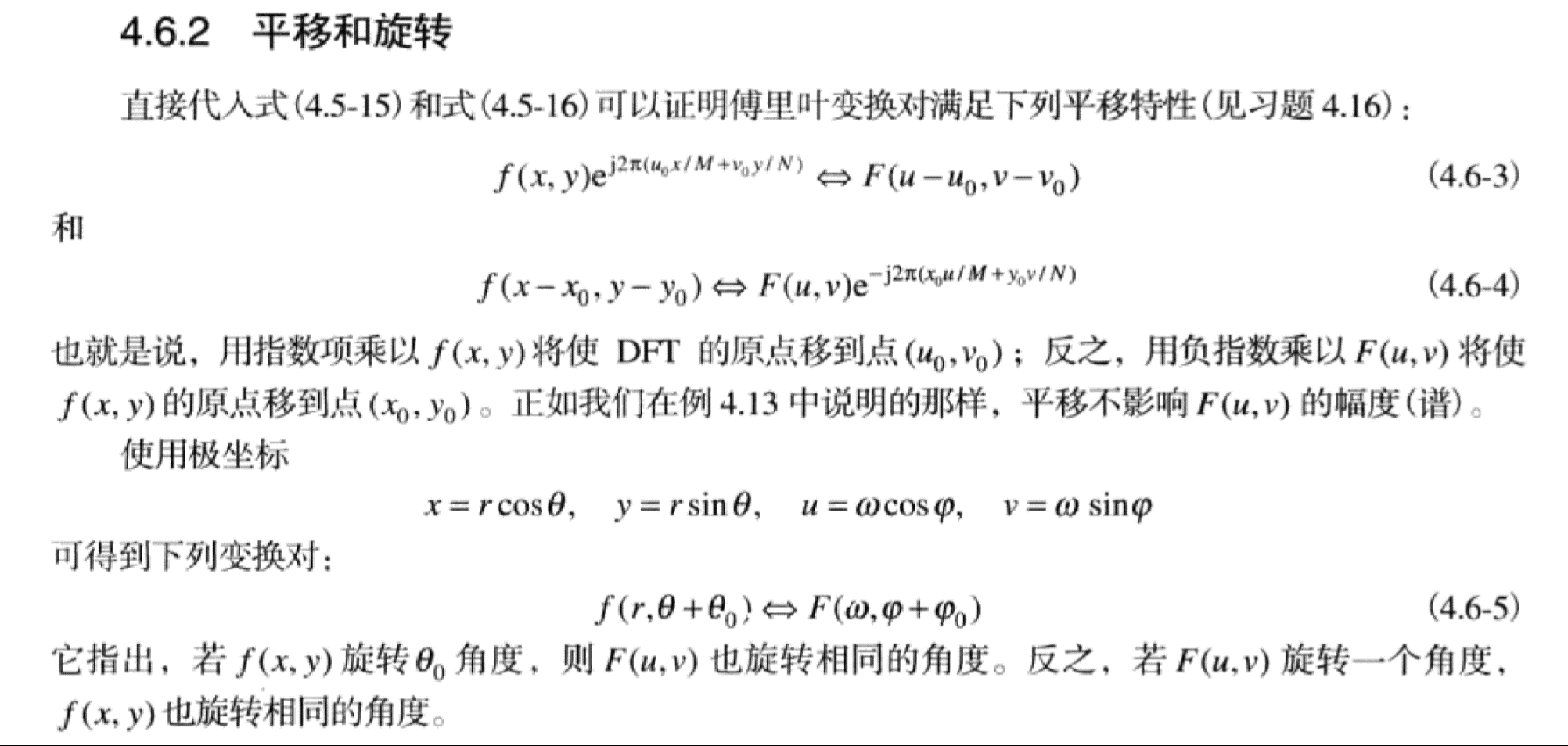
若，常数项在IDFT中。

若, 常数项分别在正变换和反变换前面。

1.2 傅里叶频谱（15分）

Ans: 对应的傅里叶频谱是一样的

理由：



由于指数项的绝对值是1，谱对图像平移是不敏感的。

由题，对(a)进行延拓，填充相同数量0，仅位置不同，即(c)图相当于对(b)图进行平移，可知对应的傅里叶频谱必然相同。

1.3 频率域滤波器

1.

H =

1.0000 + 0.0000i -0.1250 - 0.2165i -0.1250 + 0.2165i

-0.1250 - 0.2165i 0.2500 - 0.4330i -0.5000 + 0.0000i

-0.1250 + 0.2165i -0.5000 + 0.0000i 0.2500 + 0.4330i

2. Ans:

是低通滤波器。

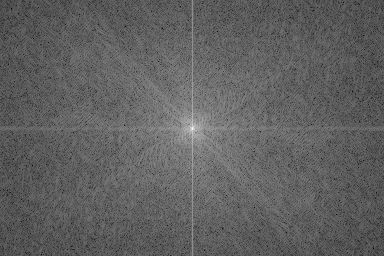
证明：

2. 编程题

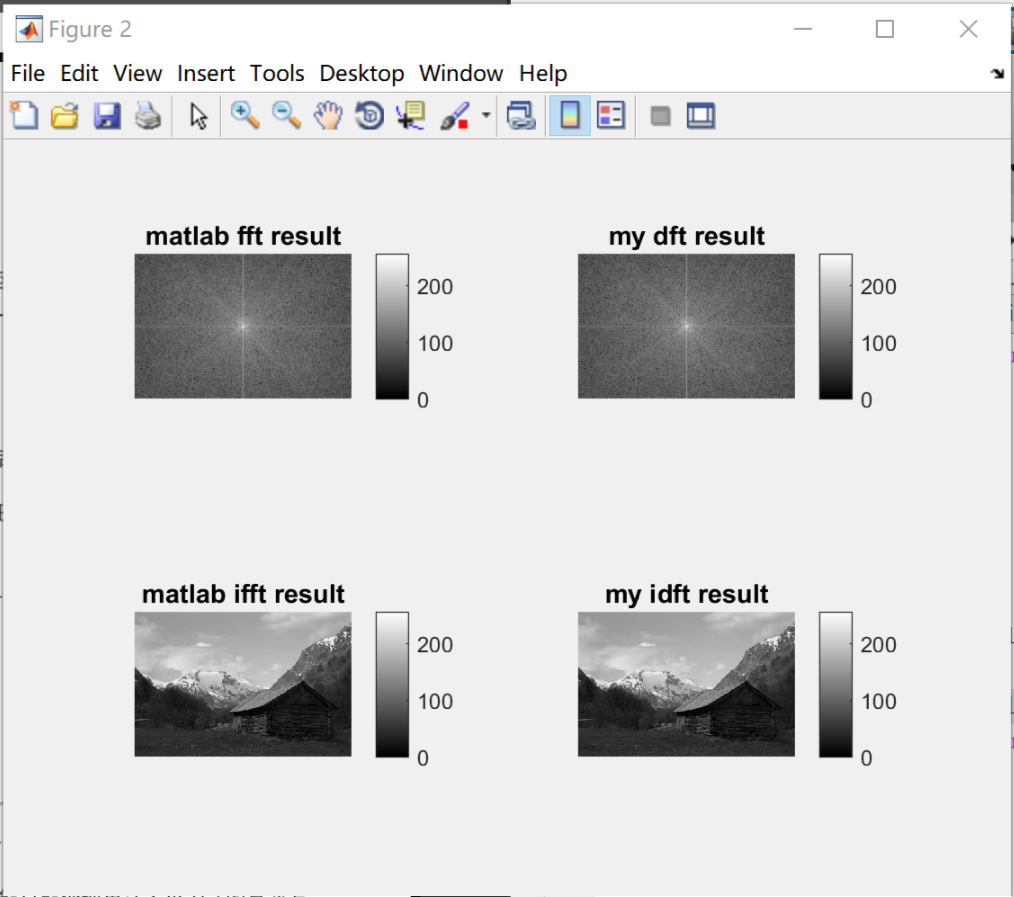
2.2 傅里叶变换（30 分）

1. （10分）对输入图像做离散傅里叶变换（DFT），并且把中心化后的傅里叶频谱粘贴到报告里。

中心化后的傅里叶频谱图：



运行效果对比图：



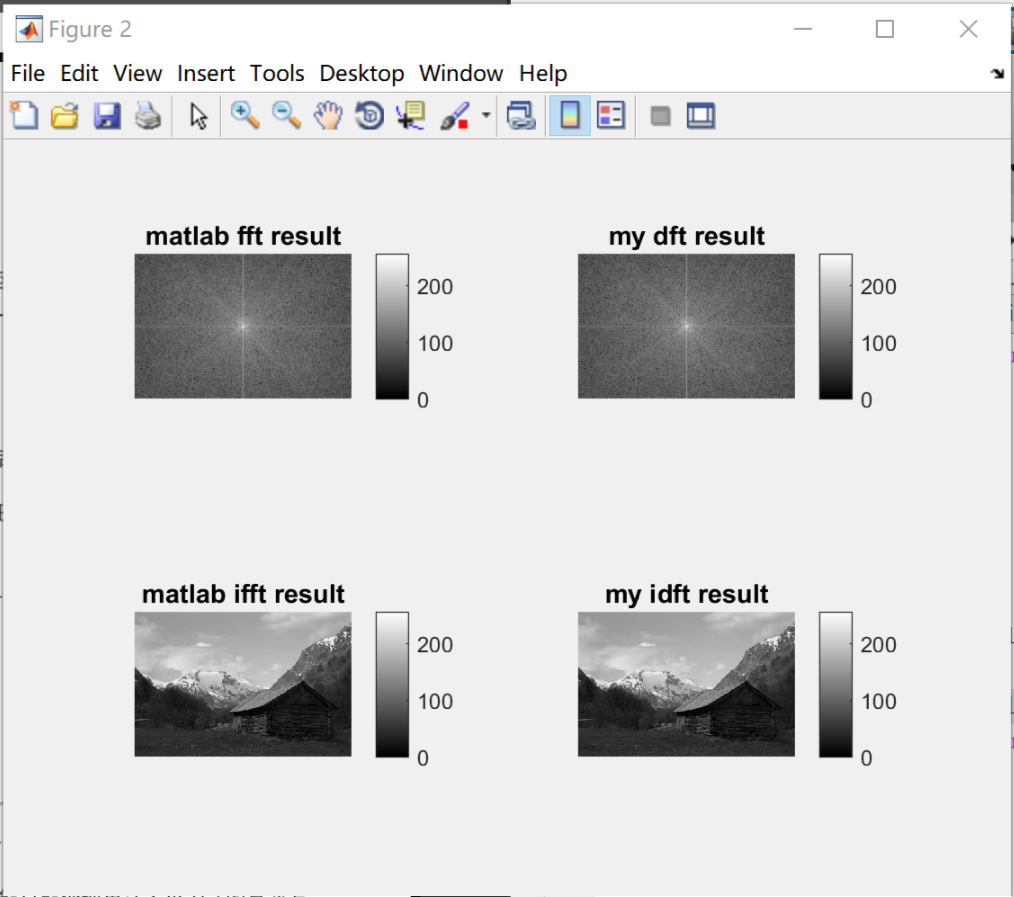
2. （5分）对上一个问题的结果做离散傅里叶反变换（IDFT），并把实部粘贴到报告里。 注意：实部与输入图像非常相似。（请思考其中的原因。）

反变换得到的实部图像：



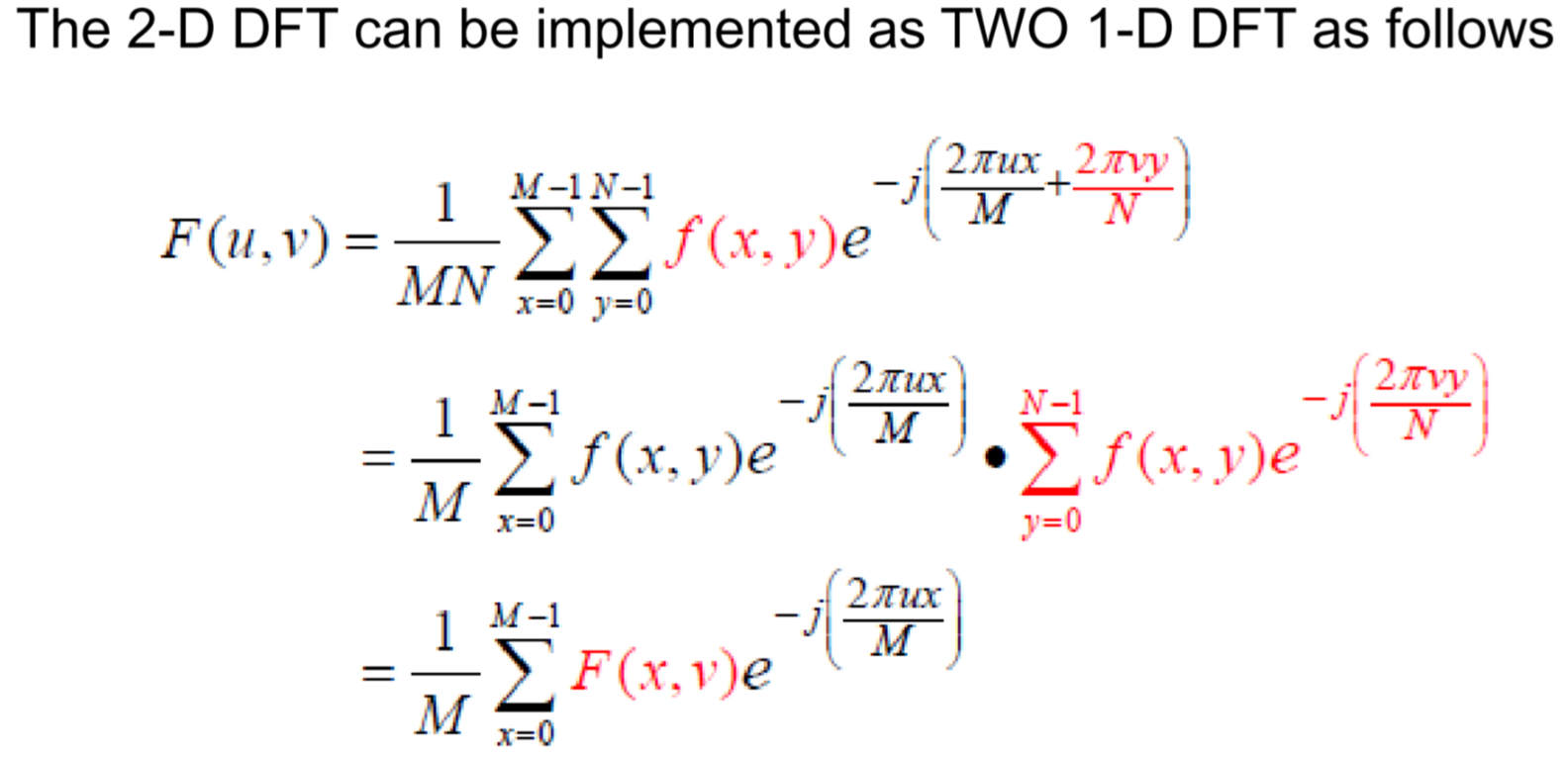
（选择实部忽略了由于计算不准确导致的寄生复分量，与输入图像十分相似）

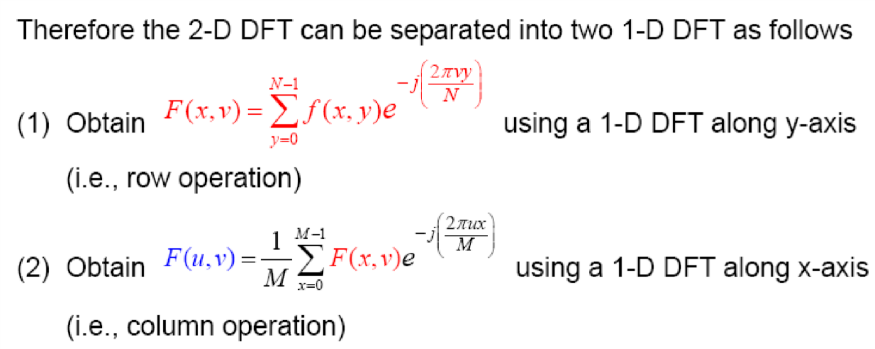
运行效果对比图：



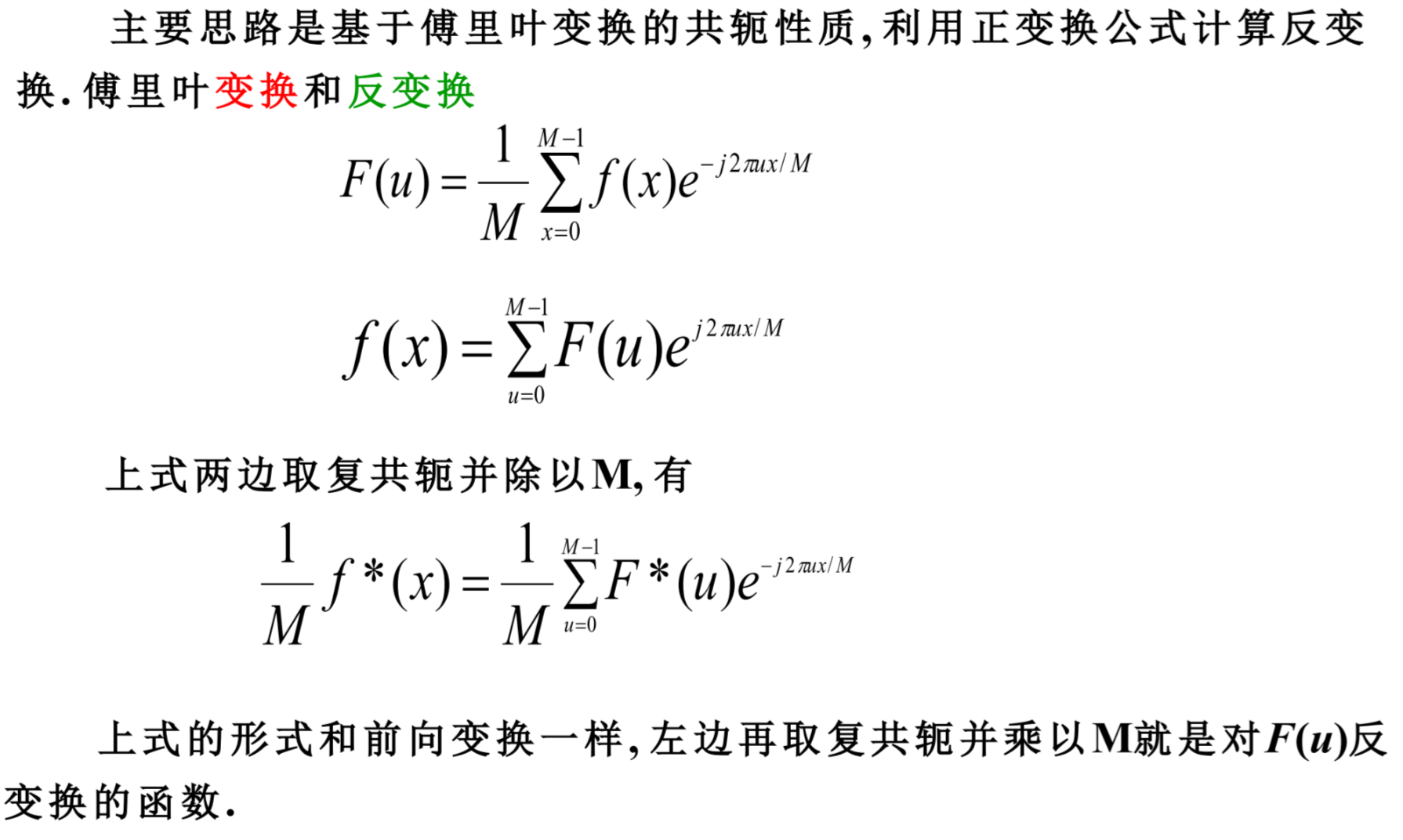
3. （15分）详细描述你是如何实现DFT / IDFT的，也就是说，针对“dft2d”函数进行算法说明，字数不能超过两页。请集中在算法方面，不要简单地复制/粘贴代码。

1. 根据二维傅里叶函数的可分性





1. 利用傅里叶变换的共轭性质，由正变换公式计算反变换



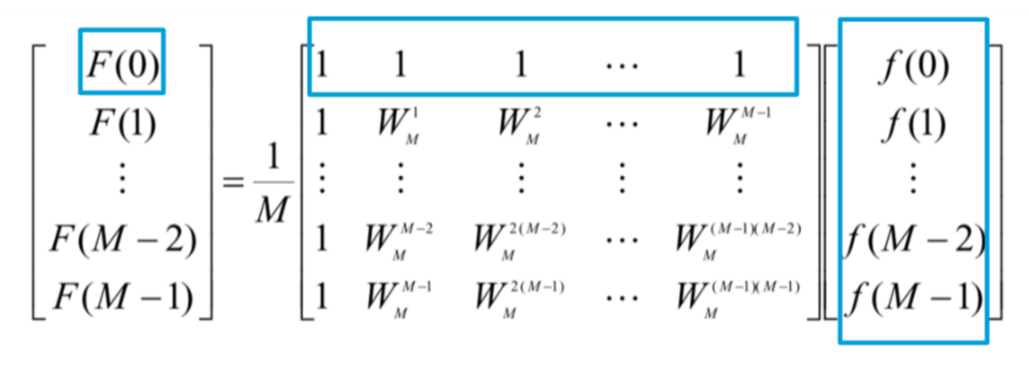
1. 一维傅里叶变换的实现(dft1d.m)

由(1)(2)可知，最重要的部分是实现一维傅里叶变换，而后只需逐行变换再用变换结果逐列变化即可得到傅里叶变换后的系数矩阵。

1d-DRT： 

对外层循环0到M-1范围内的每个u，都有内层循环x范围0到M-1使满足上述求和公式得到对应结果。

例：u = 0时：



1. 由一维傅里叶变换扩展到二维(dft2d.m)

1) 遍历复共轭矩阵的每一行，调用一维傅里叶变换函数，用变换结果代替原行。

2) 遍历矩阵每一列，调用一维傅里叶变换函数，用变换结果代替原列

1. 由二维傅里叶正变换得到反变换(dft2d.m)

1) 获得傅里叶变换所得结果的复共轭矩阵

2) 对此矩阵做正变换

3) 对正变换结果再取复共轭，得到反变换结果

注：

dft2d.m对应函数格式为function output\_img = dft2d( input\_img,flag )

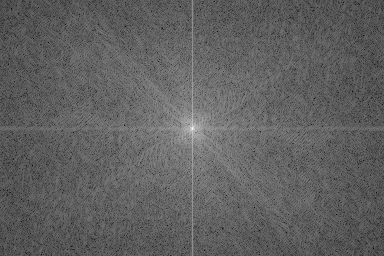
当flag = 0时，做傅里叶正变换

当flag = 1时，做傅里叶反变换

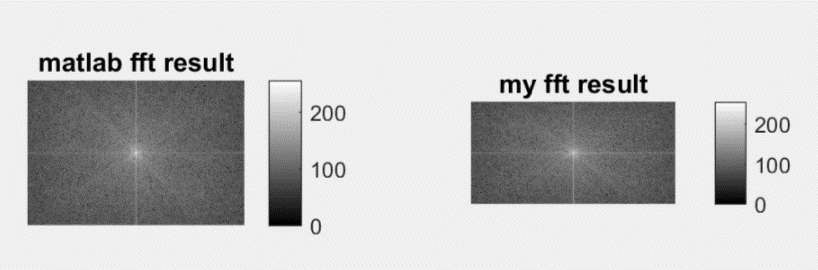
2.3 附加题：快速傅里叶变换（50 分）

1. （15分）对输入图像做快速傅里叶变换（FFT），并且把中心化后的傅里叶频谱粘贴到报告里。

中心化后的傅里叶频谱图：



运行效果对比图：

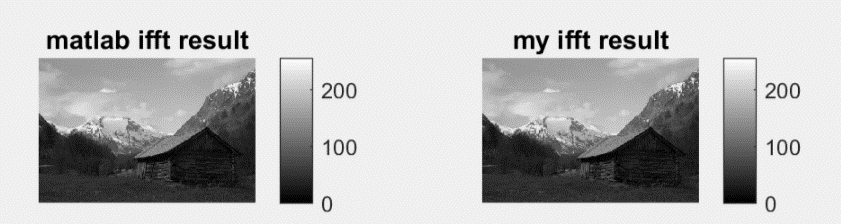


2. （10分）对上一个问题的结果做快速傅里叶反变换（IFFT），并把实部粘贴到报告里。

快速傅里叶反变换结果：



运行效果对比图：



3. （25分）详细描述你是如何实现 FFT / IFFT的，也就是说，针对“fft2d”函数进行算法说明，字数不能超过三页。

算法描述

与DFT同理，二维DFT可以使用逐次通过一维变换的方法来执行，重点考虑如何实现一维FFT。



其中 ，假设 ，可表示为

由旋转因子的性质可知：

定义，

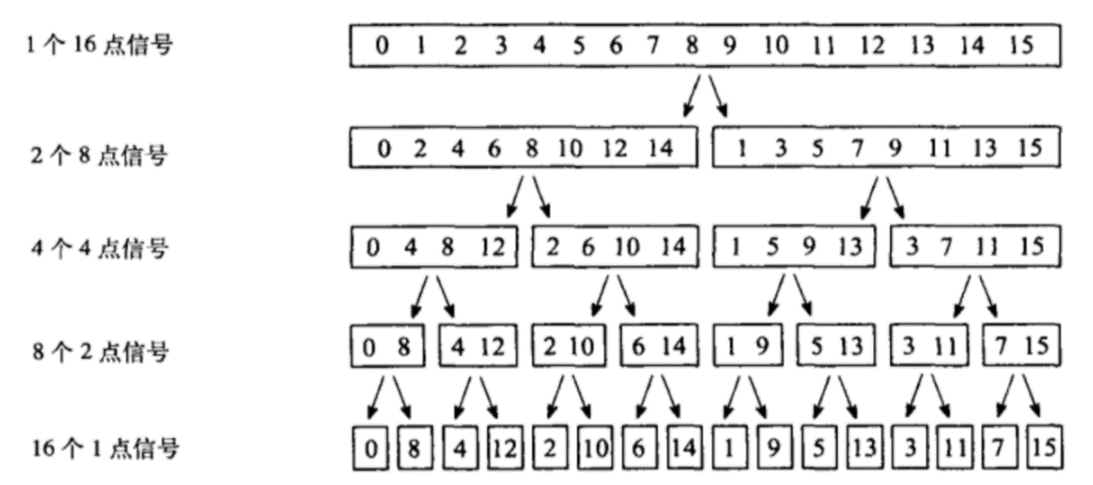
可得： ，

(1) 步分解

以M=16的16点信号为例：

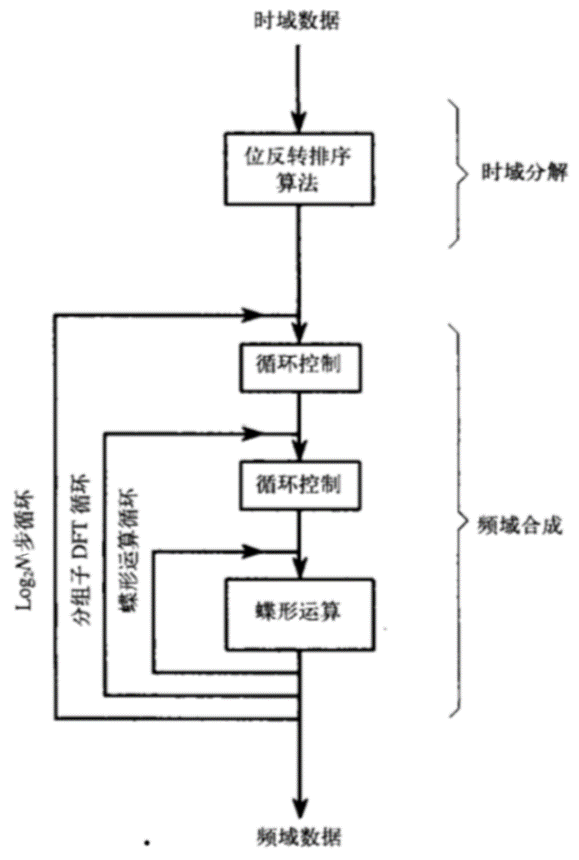
一个16点信号要经过4次分解，先分为2个8点信号，再分为4个4点信号，如此进行下去，直到得到M=16个由1点组成的信号。

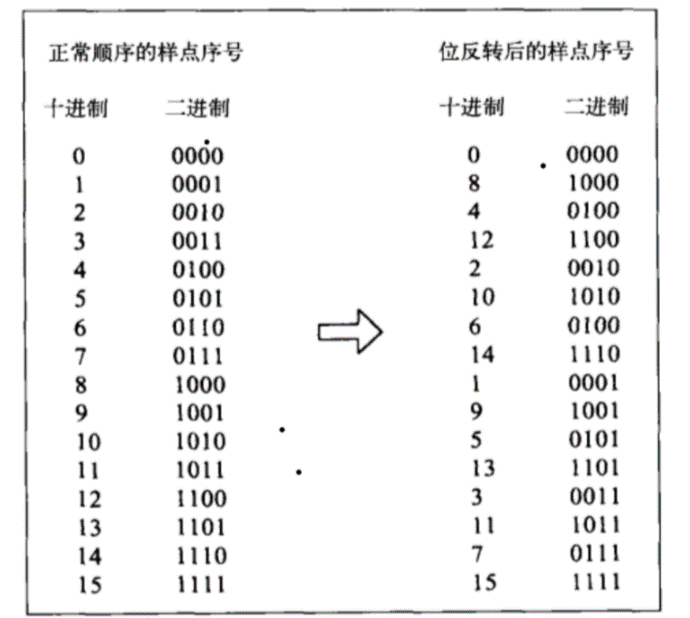
每次信号一分为二时使用交错分解，即把信号分为偶数抽样点和奇数抽样点。



(2) 二进制位反转

由分解的结构可以知道此分解实际上是对信号抽样点的重新排列，左边和右边分别为原始信号抽样点和重新排列的信号点及其对应的二进制。

FFT的时域分解可以通过二进制反转算法（把M个时域信号抽样点序号的二进制数值从左到右反转）实现。



(3) 找到1点时域信号的频谱

最底层1点信号的频谱等于它本身，将此时域数据转换到频域不需进行任何操作。（此刻其本身可看作频谱）

(4) 从底层逐阶向上合并

FFT流程分三步：①把一个M点时域信号分解为M个单点信号②计算这M个单点信号频谱（等于它本身）③把这M个频谱合成一个频谱

1) 两点蝶形算法

2）联系实现方式描述：8点DFT合成

此过程合成频域信号获得傅里叶变换的最终结果，需要三层循环。