Problem 1

证明DFT的循环卷积定理，

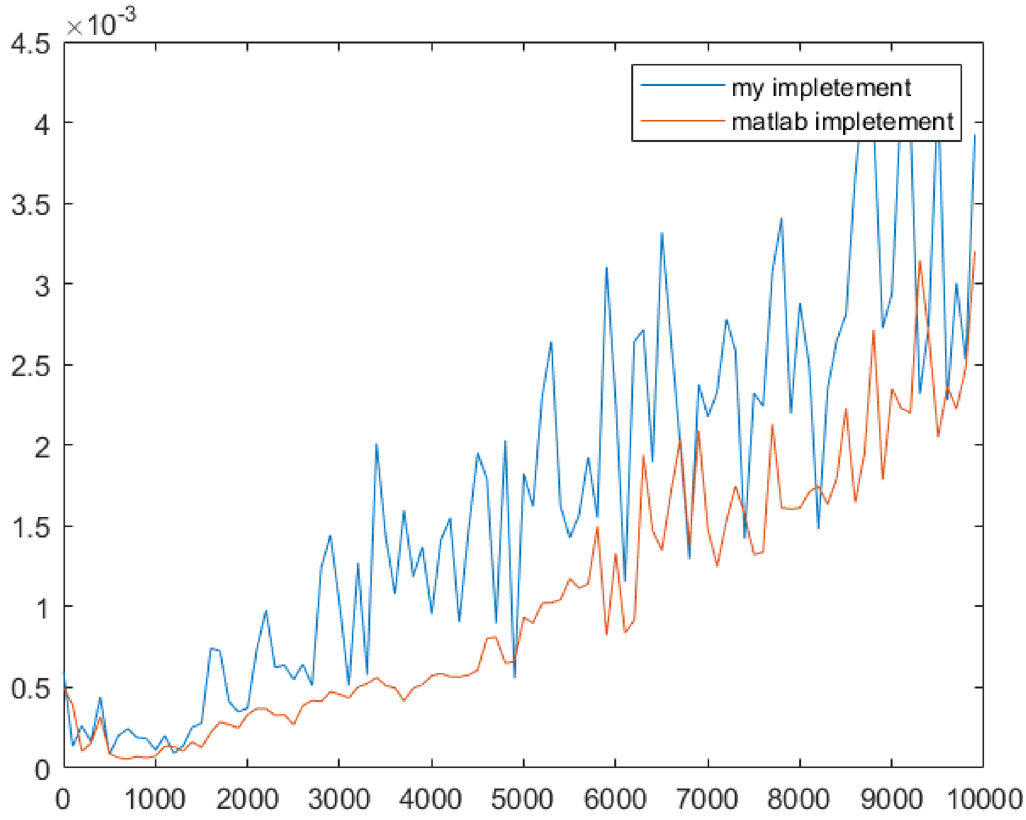
$$
\begin{aligned}
\mathrm{DFT}(u\otimes v)_k &= \sum_{j=0}^{n-1}\exp(-\frac{2\pi i}{n}jk)(u\otimes v)_j \\
&= \sum_{j=0}^{n-1}\exp(-\frac{2\pi i}{n}jk)\sum_{p=0}^{n-1}u_p v_{j-p} \\
&= \sum_{j=0}^{n-1}\exp(-\frac{2\pi i}{n}(j-p)k)\exp(-\frac{2\pi i}{n}pk)\sum_{p=0}^{n-1}u_p v_{j-p} \\
&= \sum_{p=0}^{n}\exp(-\frac{2\pi i}{n}pk)u_p\sum_{j=0}^{n-1}\exp(-\frac{2\pi i}{n}(j-p)k)v_{j-p} \\
&= \sum_{p=0}^{n}\exp(-\frac{2\pi i}{n}pk)u_p\sum_{q=0}^{n-1}\exp(-\frac{2\pi i}{n}qk)v_q \\
&= \mathrm{DFT}(v)_k\odot\mathrm{DFT}(u)_k
\end{aligned}
$$

Problem 2

编写如下的函数实现多项式乘法，首先使用零填充，之后利用卷积定理即可。

```
function w = poly_multiple(u,v)
    lu = length(u);
    lv = length(v);
    u = [u, zeros(1,lv-1)];
    v = [v, zeros(1,lu-1)];
    w = ifft(fft(u) .* fft(v));
end
```

随机生成几组数据后可以发现效果与内置函数 `conv` 的效果完全相同，令 $n=1,\cdots,10000$ 测试时间如下，

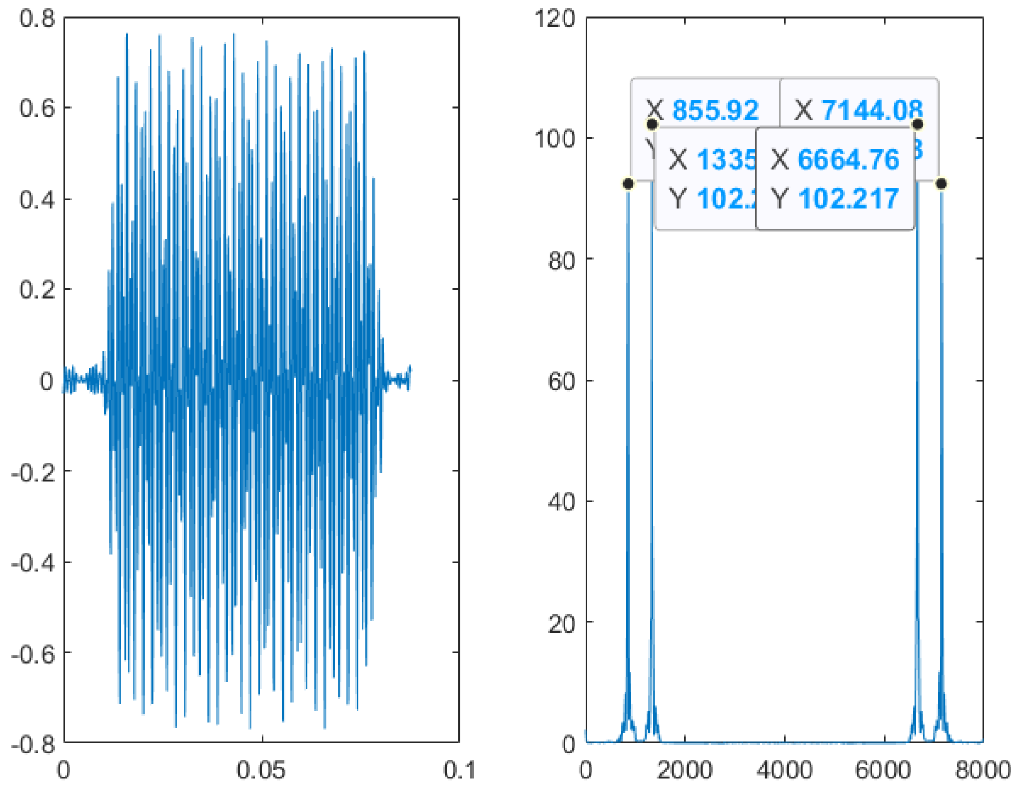


可以看到MATLAB自带的函数效率更高，但是总体的增长阶数是类似的。

分析上述代码的复杂度，FFT和IDFT的复杂度为 $\Theta(n\log n)$ 而逐元素相乘的复杂度为 $\Theta(n)$, 因此总复杂度为 $\Theta(n\log n)$.

Problem 3

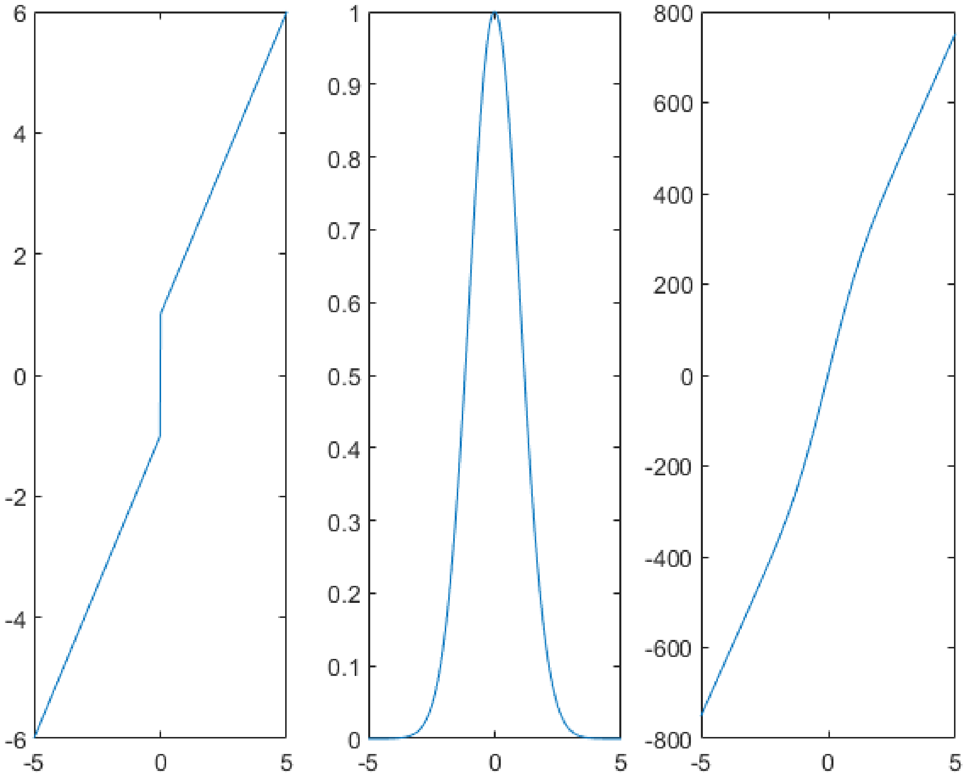
截取第 35 段音频，分别绘制出其时域和频域的图像如下，



对应峰值位置可以看出最接近的两个峰值对应于 1336Hz 与852Hz,查表可知该键为 8, 代码详见 `problem3.m` 文件。

Problem 4

生成分段函数，如下图左所示，以及高斯核函数，如下图中所示，使用卷积后生成的平滑后的函数如下图右所示，



Problem 5

首先推导导函数在频率域的形状，记 $u(t)$ 为时域图， $F(s)$ 为频域图，根据傅里叶变换公式，

$$u(x,y)=\iint \exp(i(xu+vy))F(u,v)\mathrm{d}u\mathrm{d}v=\mathrm{IFT}(F(u,v)).$$

两边同时对 x 求导可以得到，

$$\nabla_x u(x,y)=\iint iu\exp(i(xu+yv))F(u,v)\mathrm{d}u\mathrm{d}v=\mathrm{IFT}(iuF(u,v))$$

继续关于 x 求导，

$$\nabla^2_{xx}u(x,y)=-\iint u^2\exp(i(xu+yv))F(u,v)\mathrm{d}u\mathrm{d}v=-\mathrm{IFT}(u^2F(u,v)).$$

类似地，

$$\nabla^2_{yy}u(x,y)=-\mathrm{IFT}(v^2F(u,v)).$$

因此，

$$f(x,y)=\nabla^2_{xx}u(x,y)+\nabla^2_{yy}u(x,y)=-\mathrm{IFT}((u^2+v^2)F(u,v)).$$

当 $(u,v)\neq(0,0)$ 的时候，

$$F(u,v)=-\mathrm{FT}(f(x,y))/(u^2+v^2).$$

由于实际中使用的是离散傅里叶变换，作为对连续傅里叶变换的近似，但是需要注意 2π 的常数项，下面推导，

$$u(x,y)=\frac{1}{MN}\sum_{u=0}^{N-1}\sum_{v=0}^{M-1}\exp(2\pi i(xu/N+yv/M))F(u,v)=\mathrm{DFT}(F(u,v))$$

同样求导可以得到，

$$f(x,y)=\nabla^2_{xx}u(x,y)+\nabla^2_{yy}u(x,y)=-\mathrm{IFT}(4\pi^2(u^2/N^2+v^2/N^2)F(u,v)).$$

因此当 $(u,v)\neq(0,0)$ 的时候，

$$F(u,v)=-\mathrm{FT}(f(x,y))/(4\pi^2(u^2/N^2+v^2/M^2)).$$

而 $(u,v)=(0,0)$ 的点，实际上对应于原函数的常数项，这可以从二维DFT的公式简单看出，

$$F(0,0)=\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\exp(-2\pi i(xu/N+vy/M))f(x,y)\Big|_{(u,v)=(0,0)}=\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}f(x,y)$$

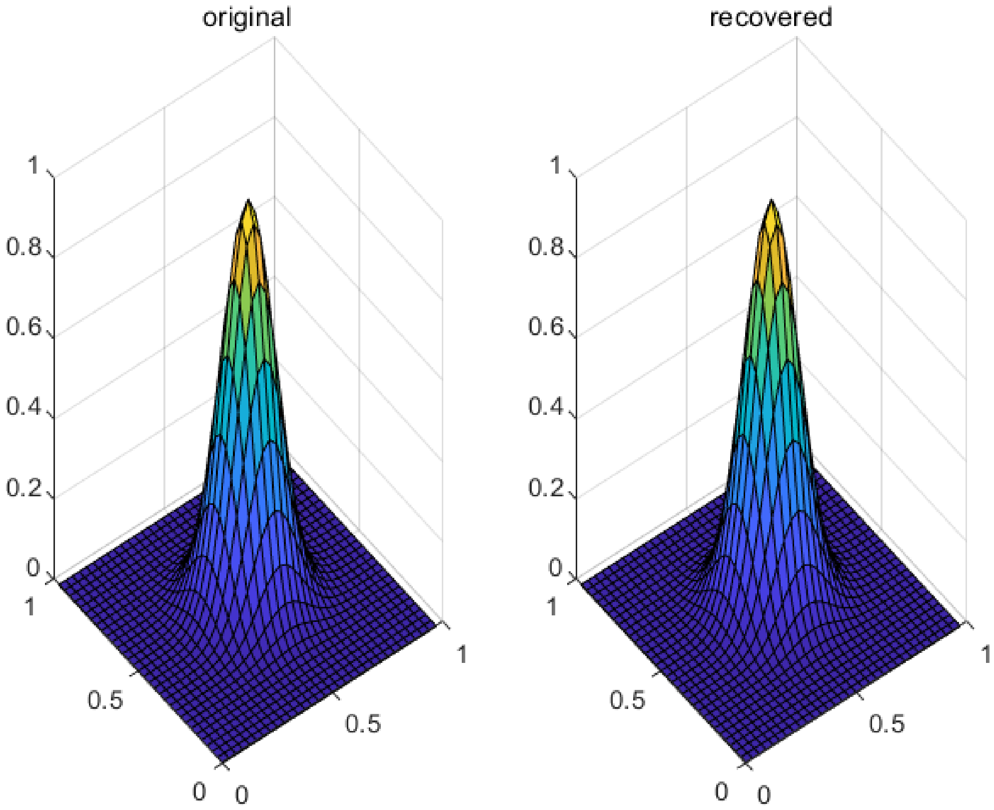
因此我们可以简单先设 $F(0,0)=0$, 之后根据逆傅里叶变换得到 $\hat{f}\,(x,y)$ ，但此时差了一个常数，可以由边界条件确定。

实际上很烦人的是离散傅里叶变换产生的实际上为 $[-N/2,N/2]$ 之间的频谱，因此编程时需要多留心处理。

采用高斯函数进行实验，

$$\begin{aligned}f(x,y)&=\exp(-((x-x_0)^2+(y-y_0)^2)/\sigma^2),\\ \nabla^2_{xx}f(x,y)+\nabla^2_{yy}f(x,y)&=\exp(-((x-x_0)^2+(y-y_0)^2)/\sigma^2)((x-x_0)^2+(y-y_0)^2)/\sigma^4-2/\sigma^2)\end{aligned}$$

当 σ 很小的时候边界条件近似为 0, 原函数和复原后的函数如下所示，


original
recovered
1
0.8
0.6
0.4
0.2
0
1
0.5
0
0
0.5
1