

华中科技大学

生物医学数字信号处理实验报告

|  |  |
| --- | --- |
| 学院 | 工程科学学院 |
| 班级 | 工程科学学院（生医）1701班 |
| 姓名 | 汪能志 |
| 学号 | U201713082 |
| 指导老师 | 全廷伟 |

2020年9月1日

目录

[1. 实验一：卷积与稀疏重建 1](#_Toc50105533)

[1.1. 实验任务 1](#_Toc50105534)

[1.2. 实验原理 1](#_Toc50105535)

[1.2.1. 一维卷积 1](#_Toc50105536)

[1.2.2. 稀疏逆卷积（稀疏重建） 1](#_Toc50105537)

[1.3. MATLAB程序实现 2](#_Toc50105538)

[1.3.1. 稀疏重建 2](#_Toc50105539)

[1.3.2. 高斯噪声 3](#_Toc50105540)

[1.3.3. 实验 3](#_Toc50105541)

[1.4. 实验结果 4](#_Toc50105542)

[1.4.1. 钙信号生成和稀疏重建 4](#_Toc50105543)

[1.4.2. 迭代过程中重建信号的变化 6](#_Toc50105544)

[1.4.3. 正则化系数对重建结果的影响 6](#_Toc50105545)

[1.4.4. 信噪比对重建结果影响 8](#_Toc50105546)

[1.5. 实验总结 10](#_Toc50105547)

[2. 实验二：傅里叶级数与傅里叶变换 11](#_Toc50105548)

[2.1. 实验目的 11](#_Toc50105549)

[2.2. 方波及其傅里叶级数逼近 11](#_Toc50105550)

[2.2.1. 实验原理 11](#_Toc50105551)

[2.2.2. MATLAB实现 11](#_Toc50105552)

[2.2.3. 实验结果 11](#_Toc50105553)

[2.3. 傅里叶变换与卷积 12](#_Toc50105554)

[2.3.1. 实验原理 12](#_Toc50105555)

[2.3.2. MATLAB实现 12](#_Toc50105556)

[2.3.3. 实验结果 13](#_Toc50105557)

[2.4. 一维离散傅里叶变换 14](#_Toc50105558)

[2.4.1. 实验原理 14](#_Toc50105559)

[2.4.2. MATLAB实现 14](#_Toc50105560)

[2.5. 二维离散傅里叶变换 15](#_Toc50105561)

[2.5.1. 实验原理 15](#_Toc50105562)

[2.5.2. MATLAB实现 16](#_Toc50105563)

[2.5.3. 实验结果 16](#_Toc50105564)

# 卷积与稀疏重建

## 实验任务

信号的产生：利用一连串的脉冲序列模拟动作电位，和指数下降函数进行卷积后得到钙信号，并对得到的钙信号添加高斯白噪音。

信号的重建：利用稀疏逆卷积，从钙信号中重建出动作电位。

## 实验原理

### 一维卷积

一维卷积的公式为：



其也可以写成线性方程组（矩阵乘法形式）如下：



### 稀疏逆卷积（稀疏重建）

稀疏重建即为求解如下的优化问题：



越大，则稀疏性的权重越大；越小，则准确性的权重越大。

记式3的最优解为：



以式1-4为基础，对式1-3进行一定的修改：



其中为一个非常小的常数，其目的是避免出现分母为0的情况。

由此可知，中的元素越小，则其在计算稀疏性时的权重越大。因此式1-5的解比式1-4的解更具有稀疏性。

则有迭代公式：



同时，在中所有元素的值相同时，式1-6退化为式1-3。

式1-6的解为：



可知，从初值开始，通过式1-7进行迭代，的稀疏性会逐渐增加，最终得到一个足够稀疏的解。在刚开始迭代时，可以使用等权重的计算范数。初值可以使用全零进行初始化。

## MATLAB程序实现

### 稀疏重建

|  |  |
| --- | --- |
| 1 | function x = deconv\_L1(y, h, lambda) |
| 2 | % 初始化 |
| 3 | % x, y & h should be column vector |
| 4 | lambda = 1e-3 \* lambda; |
| 5 | len\_y = size(y, 1); |
| 6 | len\_h = size(h, 1); |
| 7 | len\_x = len\_y - len\_h + 1; |
| 8 | x = zeros(len\_x, 1); |
| 9 | % 反转卷积模板 |
| 10 | h\_tilde = flipud(h); |
| 11 | % 初始化权重和迭代参数 |
| 12 | weight = lambda \* ones(len\_x, 1); |
| 13 | lr = 0.01; |
| 14 | epoch\_1 = 10; |
| 15 | epoch\_2 = 100; |
| 16 |  |
| 17 | for i\_1 = 1:epoch\_1 |
| 18 | for i\_2 = 1:epoch\_2 |
| 19 | % 梯度下降和软阈值迭代 |
| 20 |  |
| 21 | % 快速逆卷积，计算梯度 |
| 22 | grad = conv(h\_tilde, (conv(x, h, 'full') - y), 'full'); |
| 23 | grad = grad(len\_h:len\_y); |
| 24 | descent = lr .\* grad; |
| 25 | x = x - descent; |
| 26 | % 软阈值操作并引入非负性 |
| 27 | x = (abs(x) > weight) .\* (abs(x) - weight) .\* sign(x); |
| 28 | x = max(x, 0); |
| 29 | end |
| 30 | % 根据梯度下降结果，更新权重 |
| 31 | weight = lambda .\* (1 ./ (x + 1e-6)); |
| 32 | end |
| 33 | end |

### 高斯噪声

MATLAB中的randn函数可以生成符合标准正态分布（高斯分布）的矩阵。输入参数SNR代表信号的信噪比，单位为分贝（dB）。

|  |  |
| --- | --- |
| 1 | function Signal\_Noise = Add\_Noise(Signal, SNR) |
| 2 | % 计算信号功率 |
| 3 | Signal\_Power = sum(abs(Signal(:)).^2) / numel(Signal); |
| 4 | Signal\_dB = 10 \* log10(Signal\_Power); |
| 5 |  |
| 6 | % 计算噪声功率 |
| 7 | Noise\_dB = Signal\_dB - SNR; |
| 8 | Noise\_Power = 10^(Noise\_dB / 10); |
| 9 |  |
| 10 | % 生成噪声信号 |
| 11 | Rand\_Noise = randn(size(Signal)); |
| 12 | Noise = sqrt(Noise\_Power) \* Rand\_Noise; |
| 13 |  |
| 14 | Signal\_Noise = Signal + Noise; |
| 15 |  |
| 16 | end |

### 实验

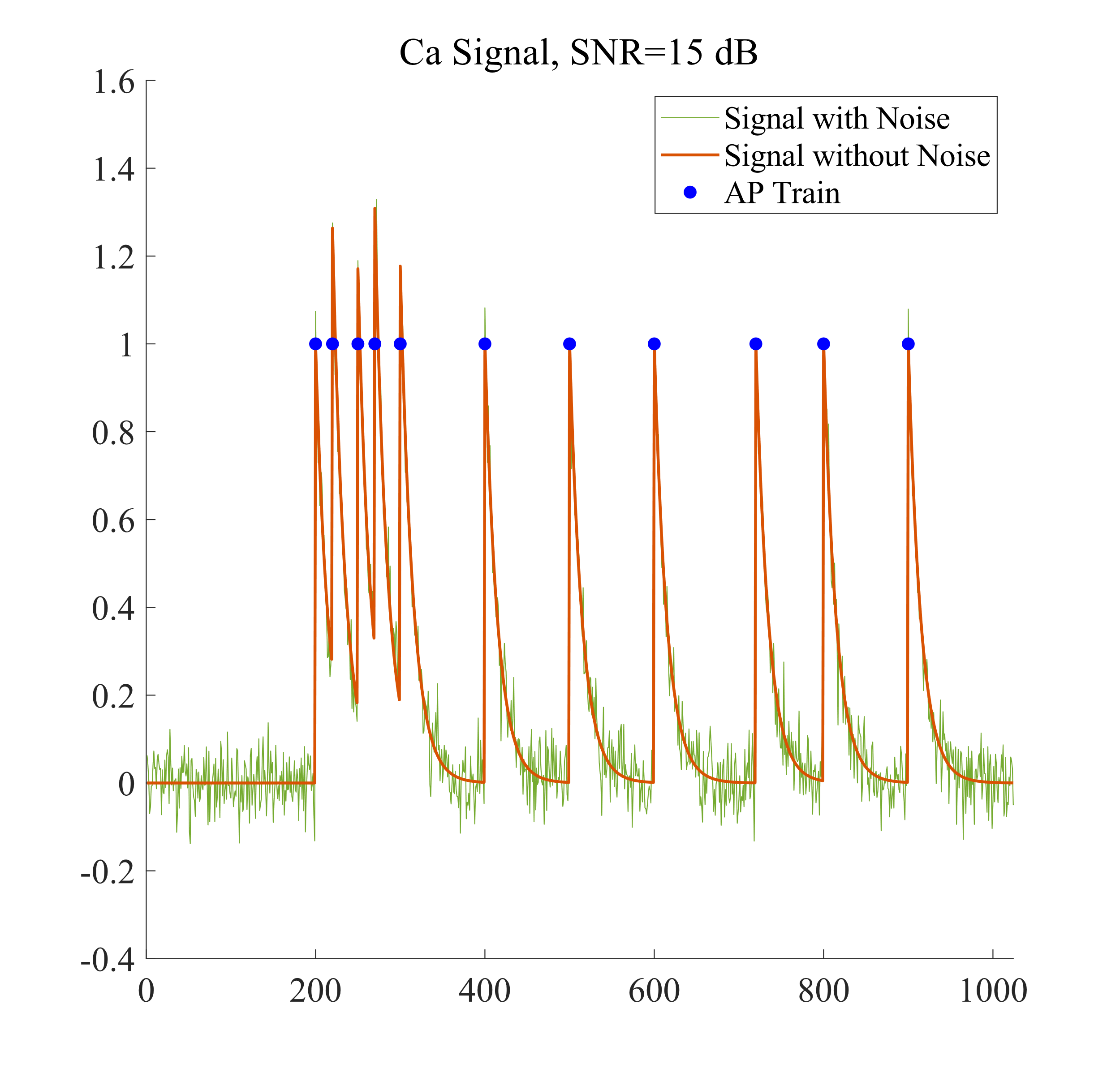
|  |  |
| --- | --- |
| 1 | %% |
| 2 | % 信号的产生 |
| 3 | % 动作电位脉冲 |
| 4 | x = zeros(1024, 1); |
| 5 | ap\_train = [200, 220, 250, 270, 300, 400, 500, 720, 600, 800, 900]; |
| 6 | x(ap\_train, 1) = 1; |
| 7 | % 指数下降模板 |
| 8 | t = 15; |
| 9 | h = exp(-(0:1:ceil(10 \* t)) / t)'; |
| 10 | % 无噪声的钙信号 |
| 11 | y = conv(x, h, 'full'); |
| 12 | % 信噪比dB |
| 13 | SNR = 15; |
| 14 | % 有噪声的钙信号 |
| 15 | y\_noise = Add\_Noise(y, SNR); |
| 16 | % 稀疏重建 |
| 17 | % 正则化系数（\*1e-3) |
| 18 | L = 0.5; |
| 19 | x\_deconv\_L1 = deconv\_L1(y\_noise, h, L); |
| 20 |  |
| 21 | %% |
| 22 | % 绘图 |
| 23 | figure(1) |
| 24 | plot(y\_noise, '-', 'color', [0.46, 0.67, 0.19]) |
| 25 | hold on |
| 26 | plot(y, '-', 'LineWidth', 1.5, 'color', [0.85, 0.32, 0.01]); |
| 27 | plot(ap\_train, 1, '.b', 'MarkerSize', 20); |
| 28 | hold off |
| 29 | title(sprintf('Ca Signal, SNR=%d dB', SNR)) |
| 30 | legend('Signal with Noise', 'Signal without Noise', 'AP Train') |
| 31 | xlim([0, 1024]) |
| 32 | ylim([-0.4, 1.6]) |
| 33 | box off |
| 34 |  |
| 35 | figure(2) |
| 36 | plot(x\_deconv\_L1, '-r'); |
| 37 | hold on |
| 38 | plot(ap\_train, 1, '.b', 'MarkerSize', 20); |
| 39 | legend('Sparse Reconstruction', 'AP Train') |
| 40 | hold off |
| 41 | title(sprintf('Sparse Reconstruction, SNR=%d dB, λ=%.5f', SNR, L \* 1e-3)) |
| 42 | xlim([0, 1024]) |
| 43 | ylim([-0.02, 1.3]) |
| 44 | box off |

## 实验结果

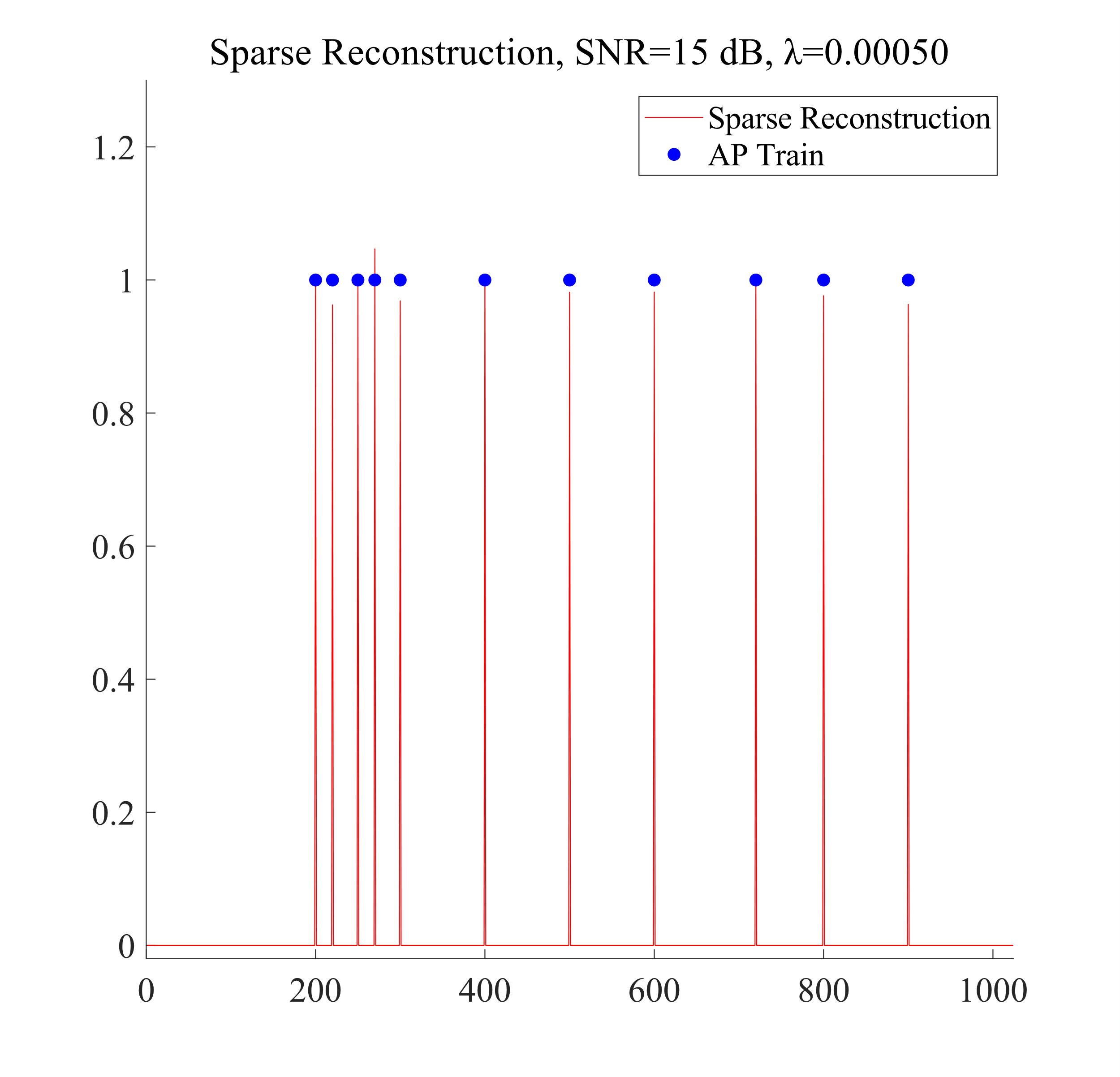
### 钙信号生成和稀疏重建

在200，220，250，270，300，400，500，600，720，800和900处共有11次动作电位。指数下降模板的时间常数为15。和动作电位信号卷积后得到钙信号，并添加信噪比为15dB的高斯噪声。

随后对带有噪声的钙信号进行稀疏重建。

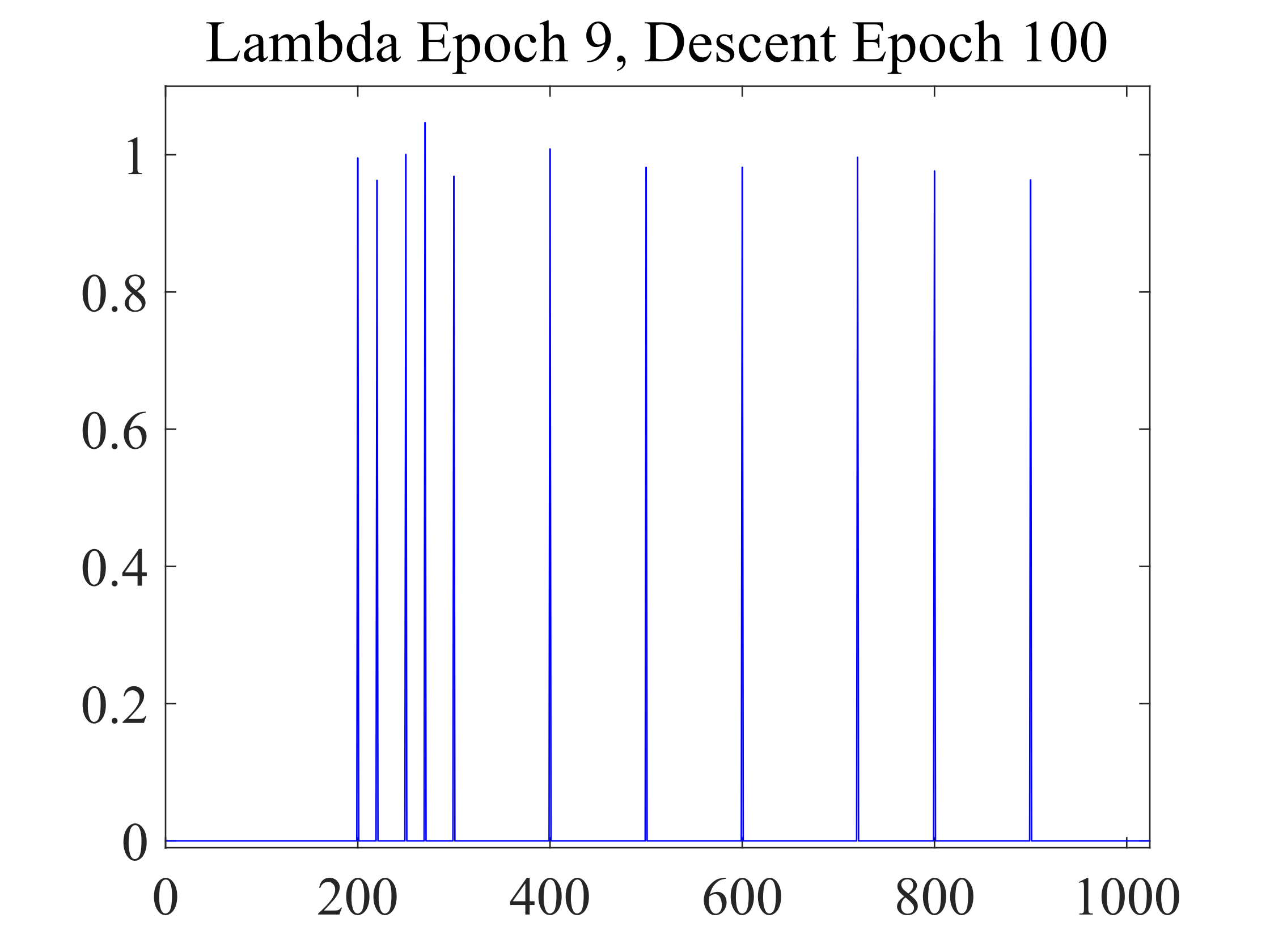
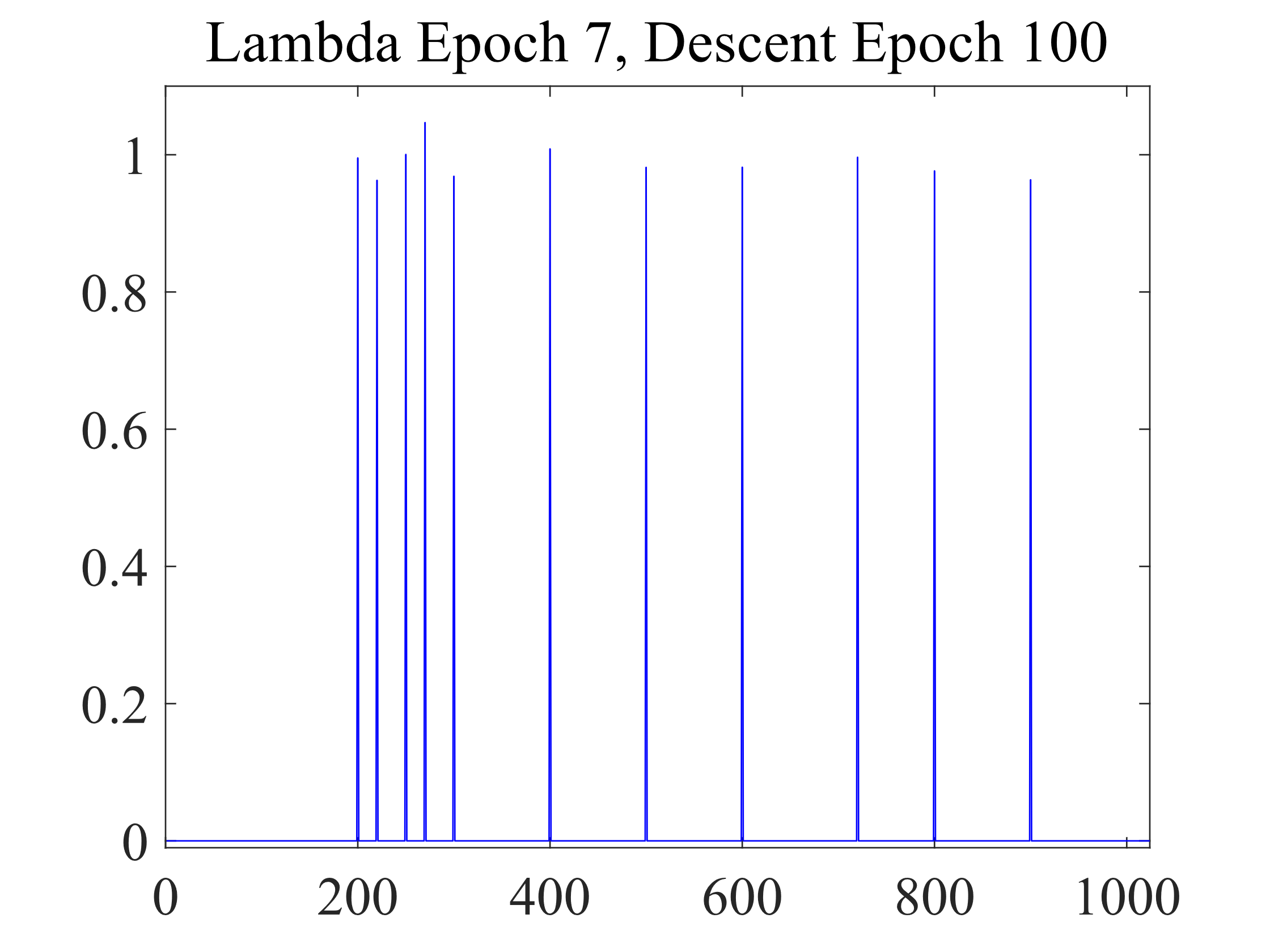
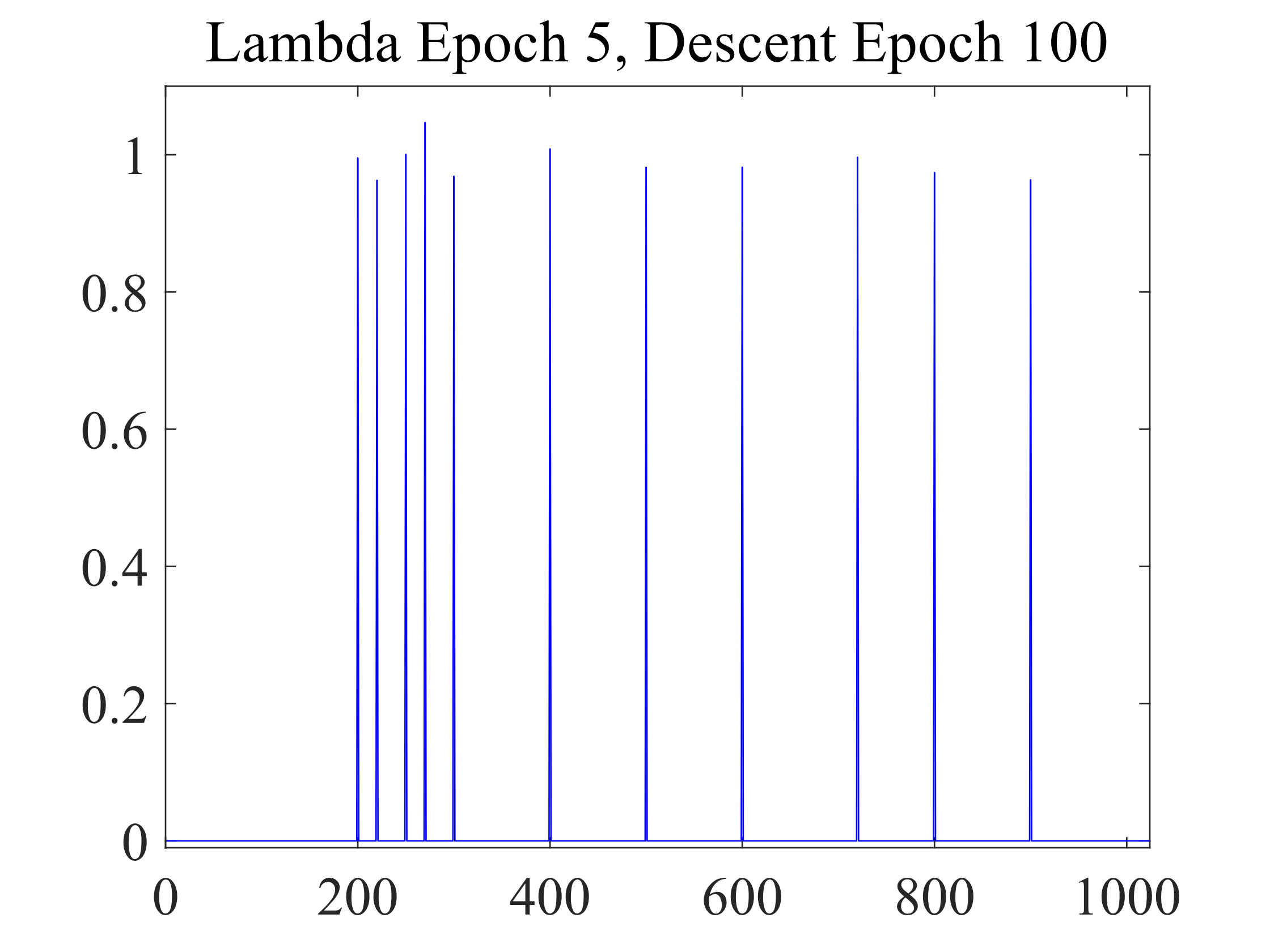
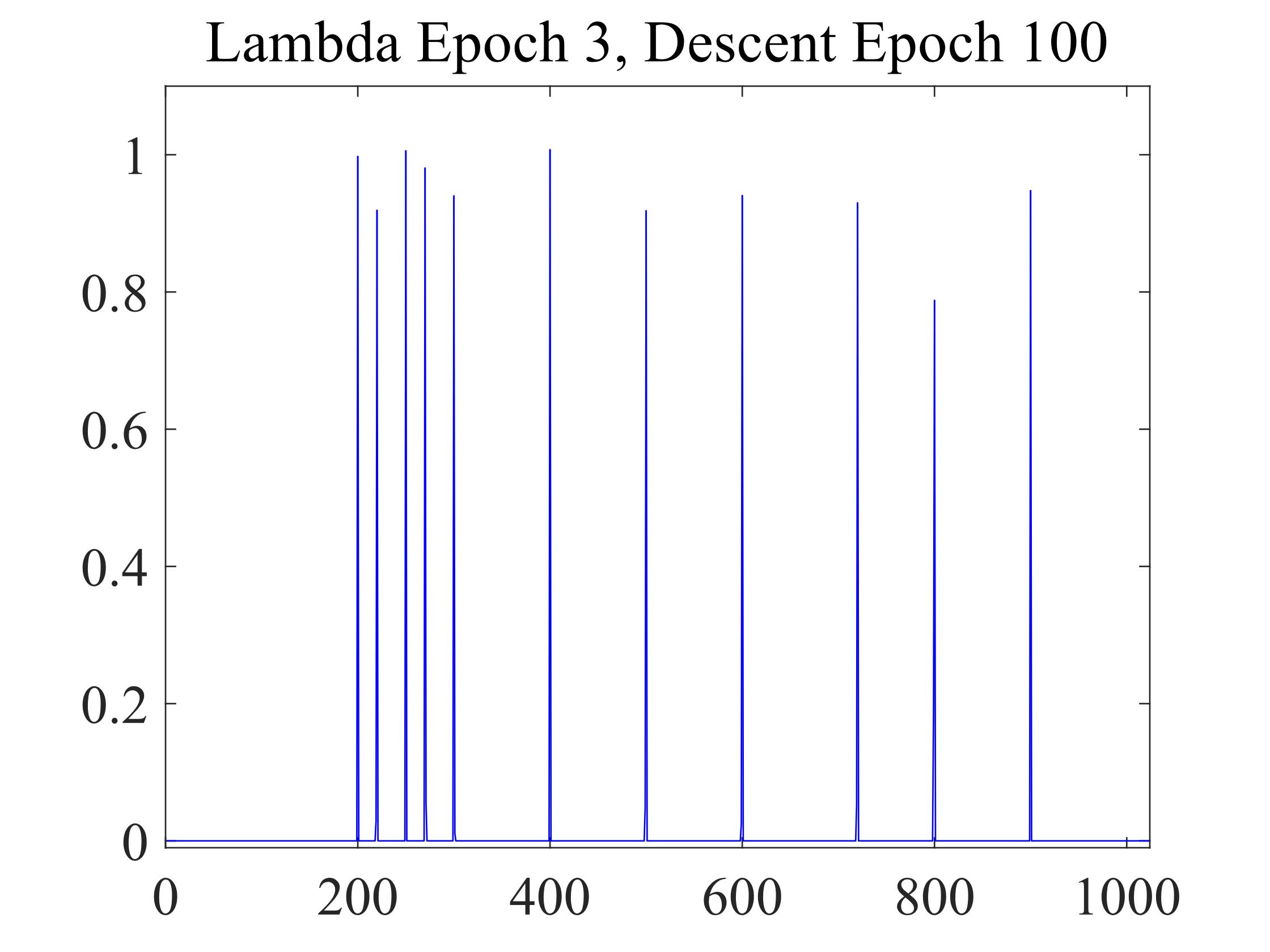
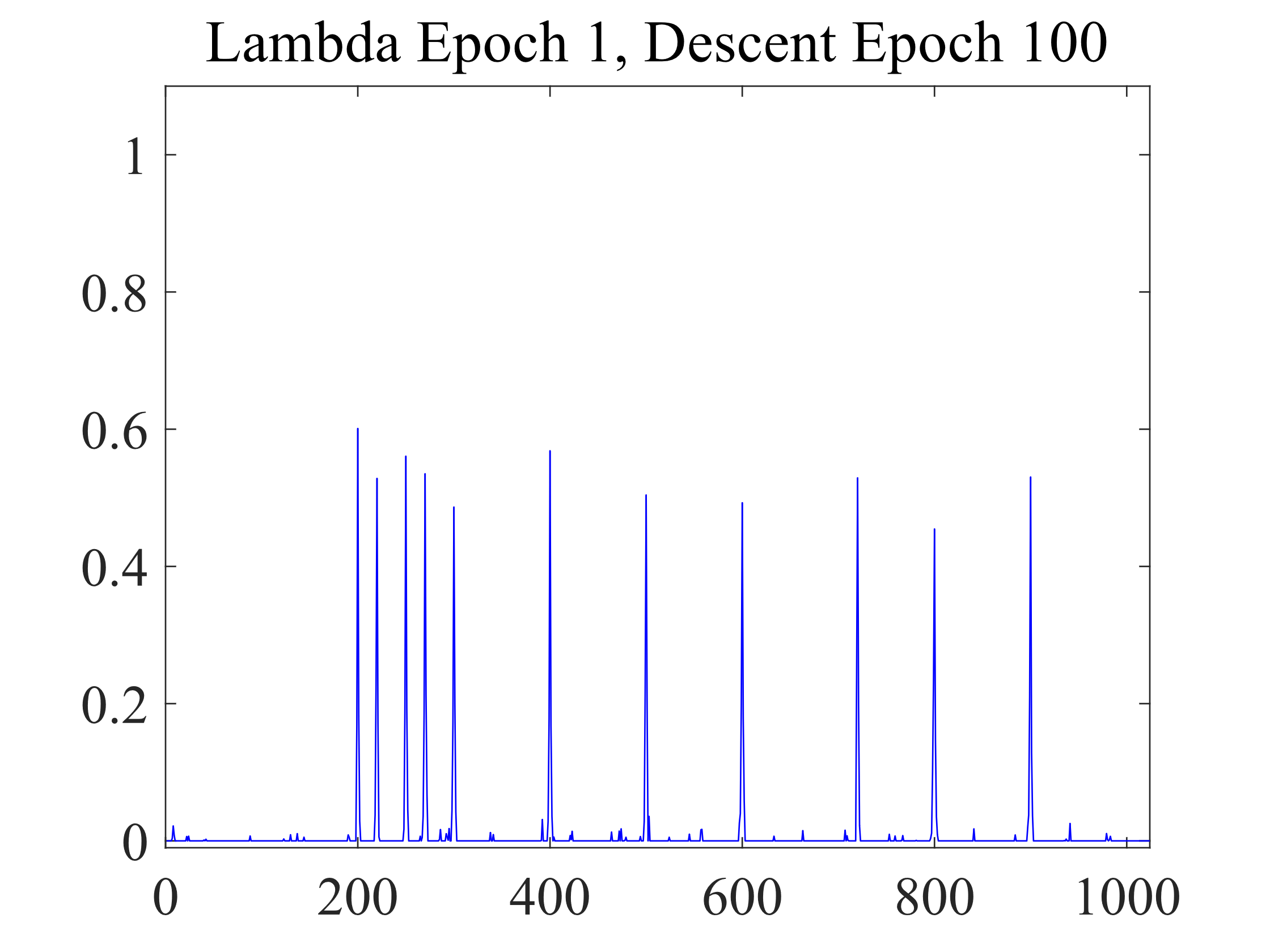
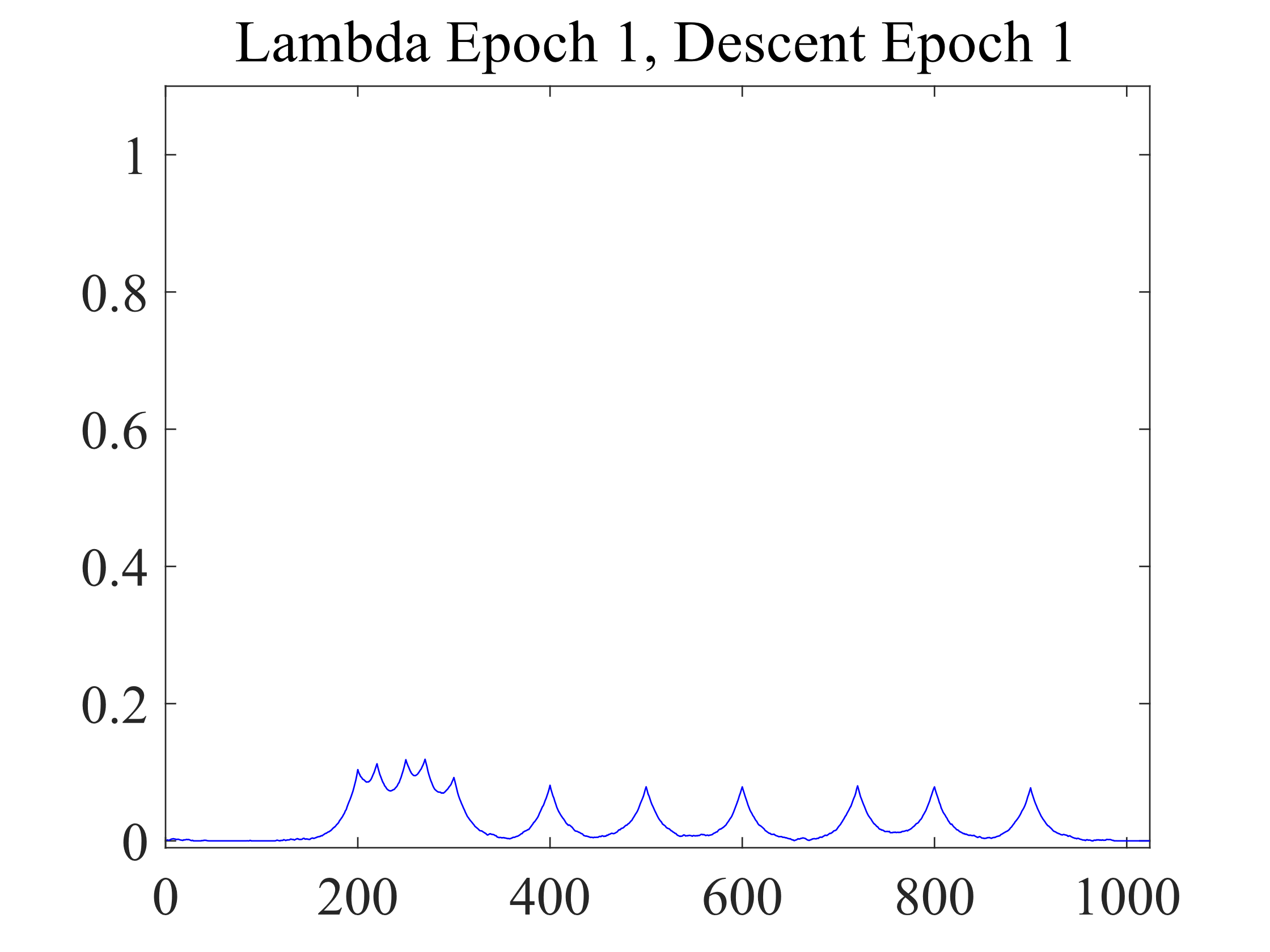


信噪比为15dB的钙信号



信噪比为15dB，正则化系数为时的稀疏重建结果

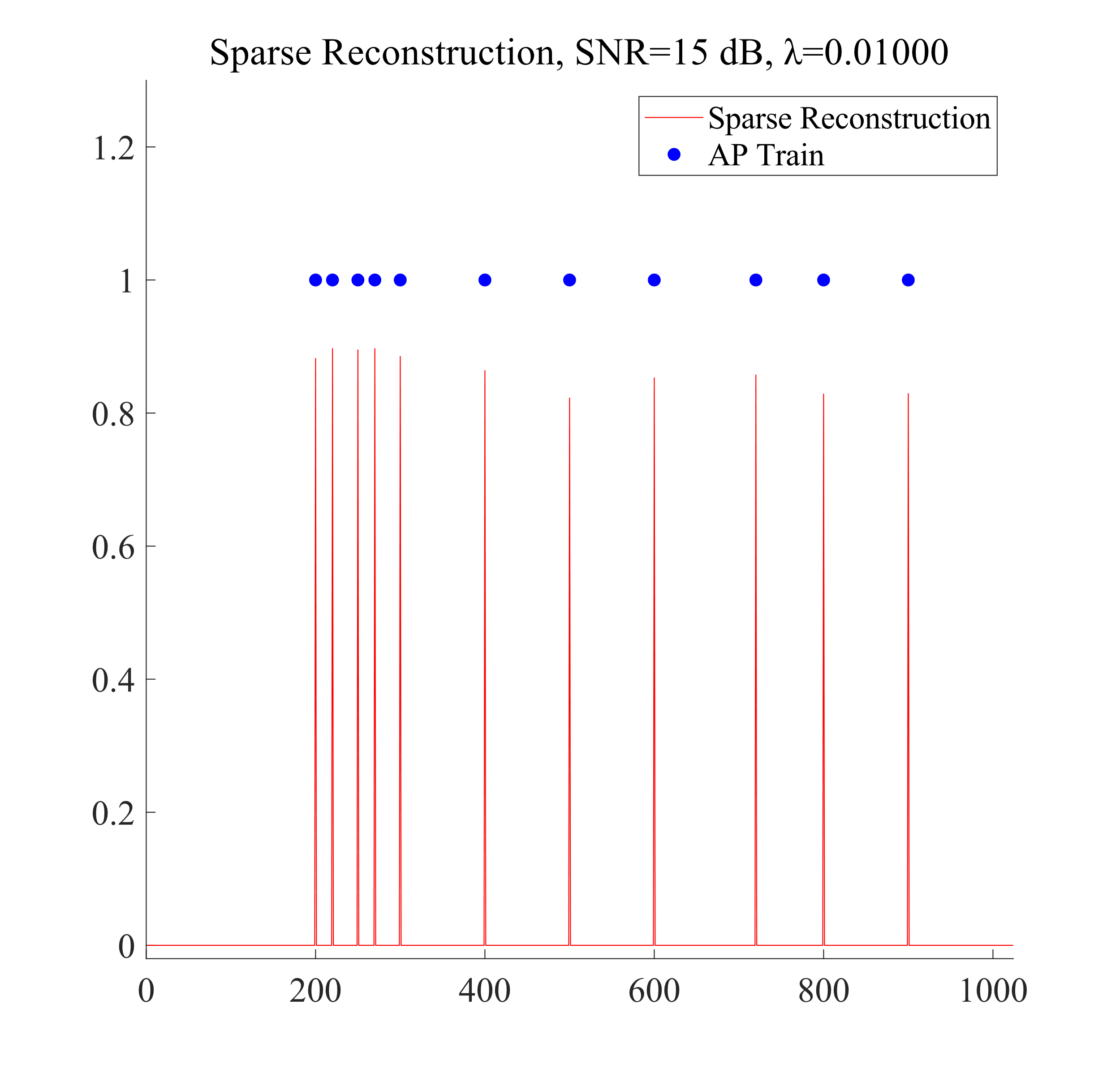
### 迭代过程中重建信号的变化



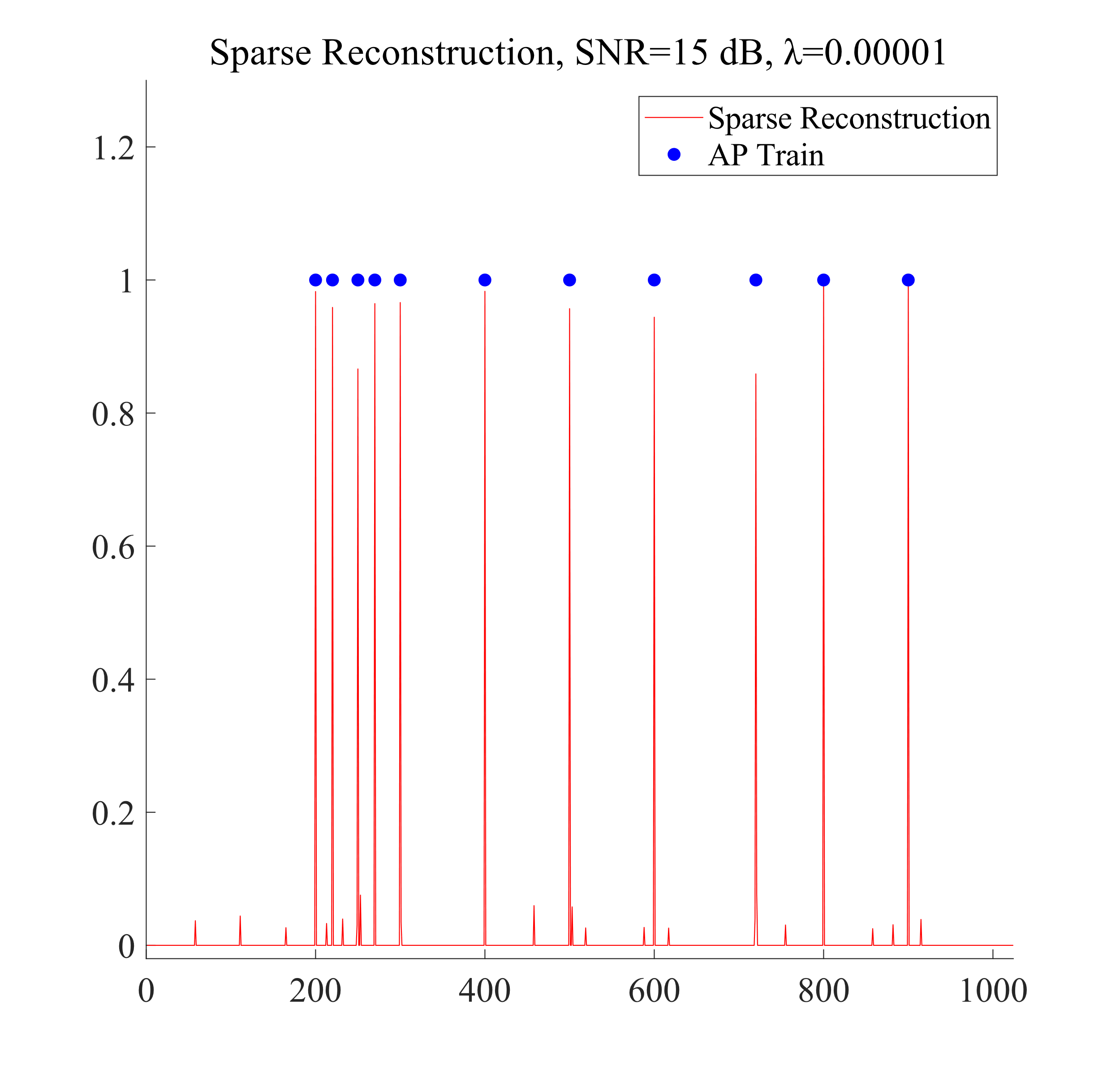
可见，在迭代刚开始时，动作电位序列就已经可以较为明显地被分辨出来。在根据梯度下降的结果调整范数权重后，动作电位序列会更明显地凸显出来，而因为噪声导致的信号会被抑制。

### 正则化系数对重建结果的影响

保持钙信号的信噪比为15dB，迭代次数和梯度下降率均保持不变，调整重建时的正则化系数。



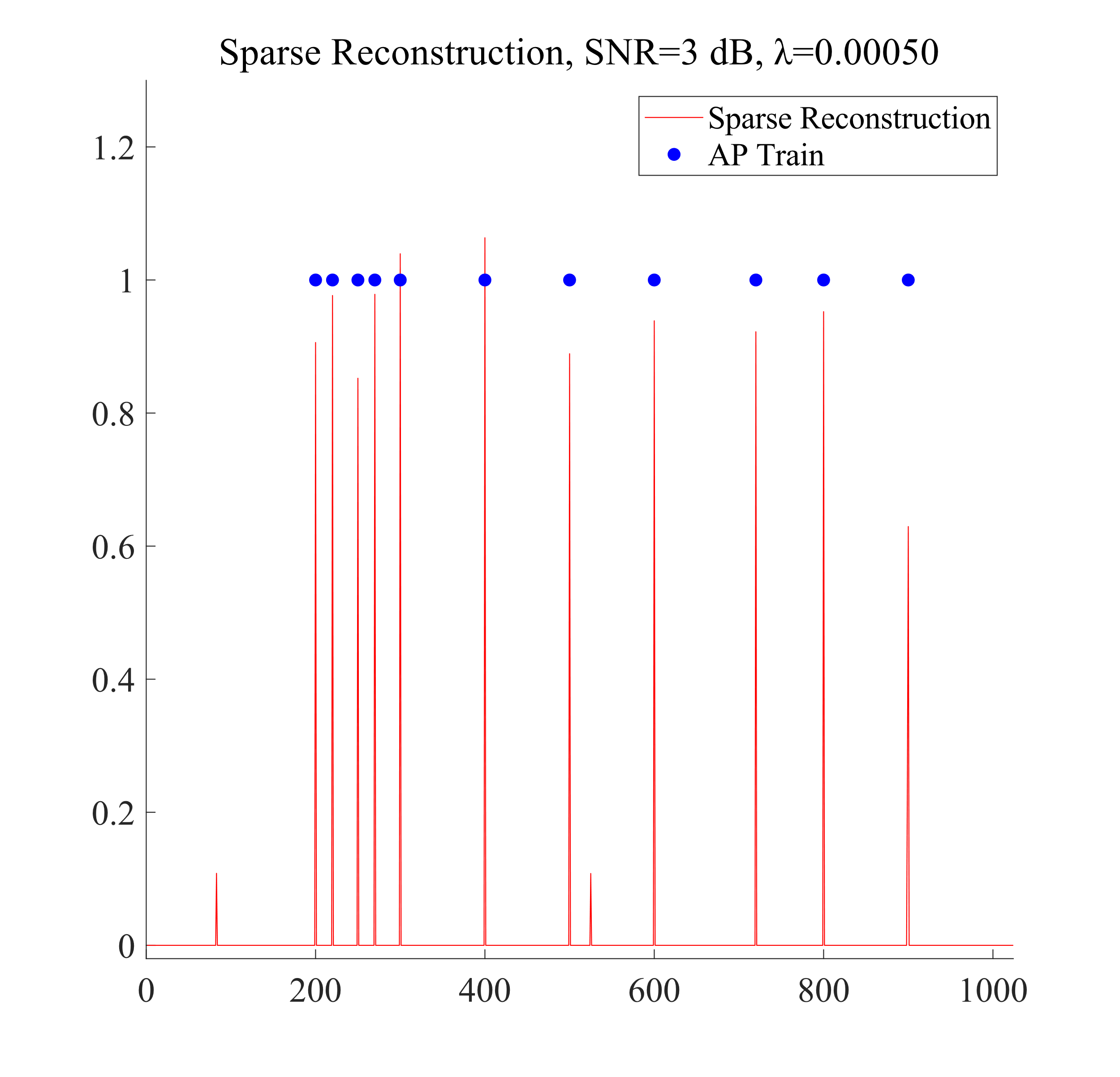
信噪比为15dB，正则化系数为时的稀疏重建结果



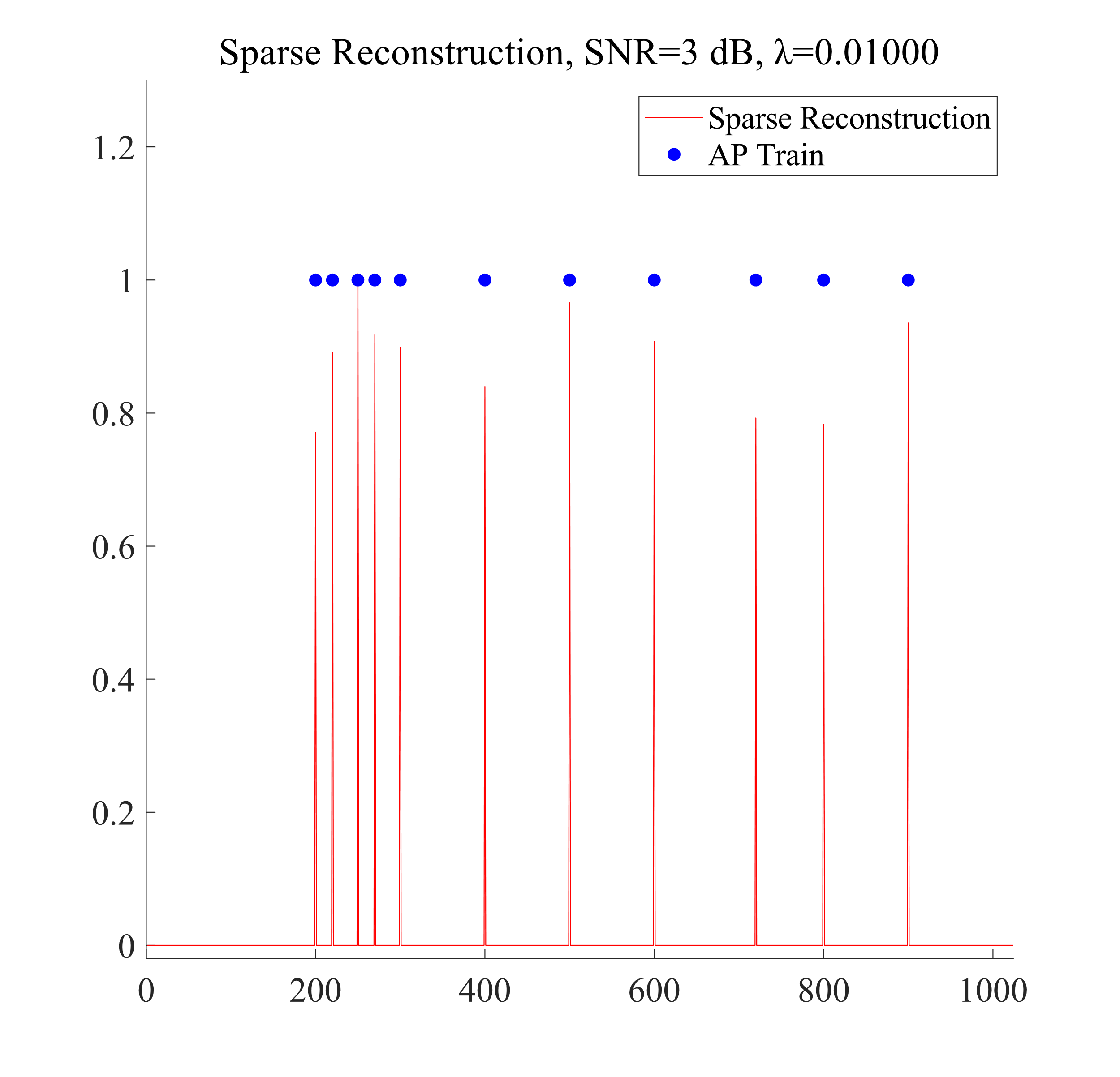
信噪比为15dB，正则化系数为时的稀疏重建结果

可见，在提高正则化系数（即提高稀疏性时），计算得到的动作电位值会下降。在继续提高正则化系数时，甚至会重建出全零的信号。而在降低正则化系数时，会导致噪声信号不能被很好地抑制。

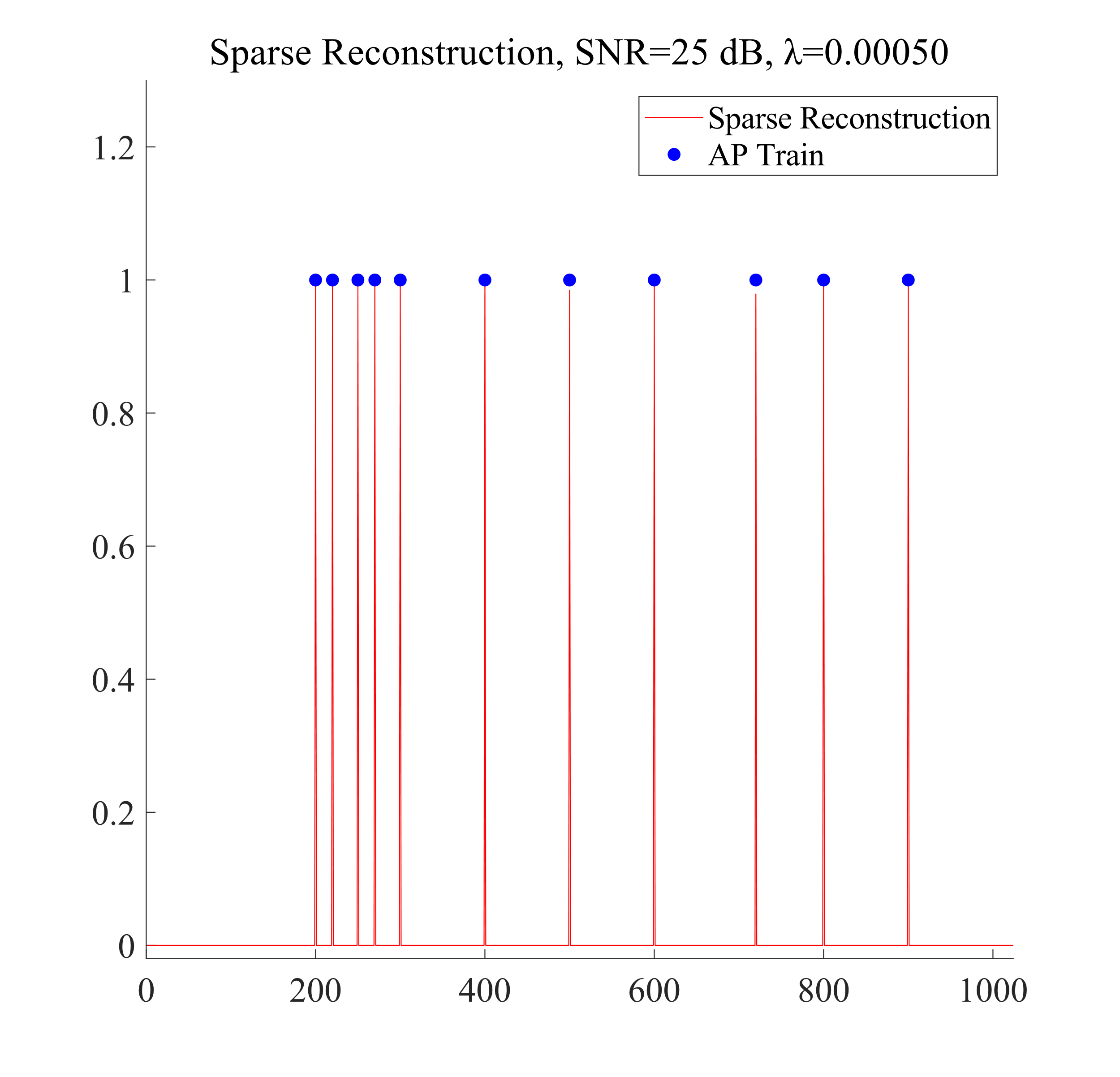
### 信噪比对重建结果影响



信噪比为3dB，正则化系数为时的稀疏重建结果



信噪比为3dB，正则化系数为时的稀疏重建结果



信噪比为25dB，正则化系数为时的稀疏重建结果

将钙信号的信噪比降低到3dB，仍然能较好地重建动作电位，但是动作电位的幅度波动较大，而且噪声信号并没有被很好地抑制。在提高正则化系数后，噪声信号得到了较好地抑制，但是动作电位幅度仍然波动较大，由于已知原始信号是二值信号，可以考虑对重建结果进行二值化来获得较好的重建结果。

将信噪比提高到25dB时，此时噪声已经非常小了，重建效果已经非常接近原始数据了。

## 实验总结

本次实验的主要内容是利用卷积模板生成钙信号，并对带有噪声的钙信号进行稀疏重建。生成信号的任务较为简单，

由于在之前的实验阶段，已经对二维图像的稀疏逆卷积操作有所了解，这次的稀疏重建算法也在一定程度上参考了之前实验的代码。同时对于计算过程进行了一定的修改，使其更简洁。较为复杂的部分是稀疏重建的超参数选择，如梯度下降率和正则化系数。需要多次寻找，才能找到一组合适的超参数，获得较好的重建结果。

# 傅里叶级数与傅里叶变换

## 实验目的

用傅里叶级数逼近方波信号；

利用傅里叶变换从钙信号中重建出动作电位序列；

理解并实现fft，ifft，fft2和ifft2这四个函数的功能。

## 方波及其傅里叶级数逼近

### 实验原理

对于离散信号，可以使用累加的方法来近似计算积分：

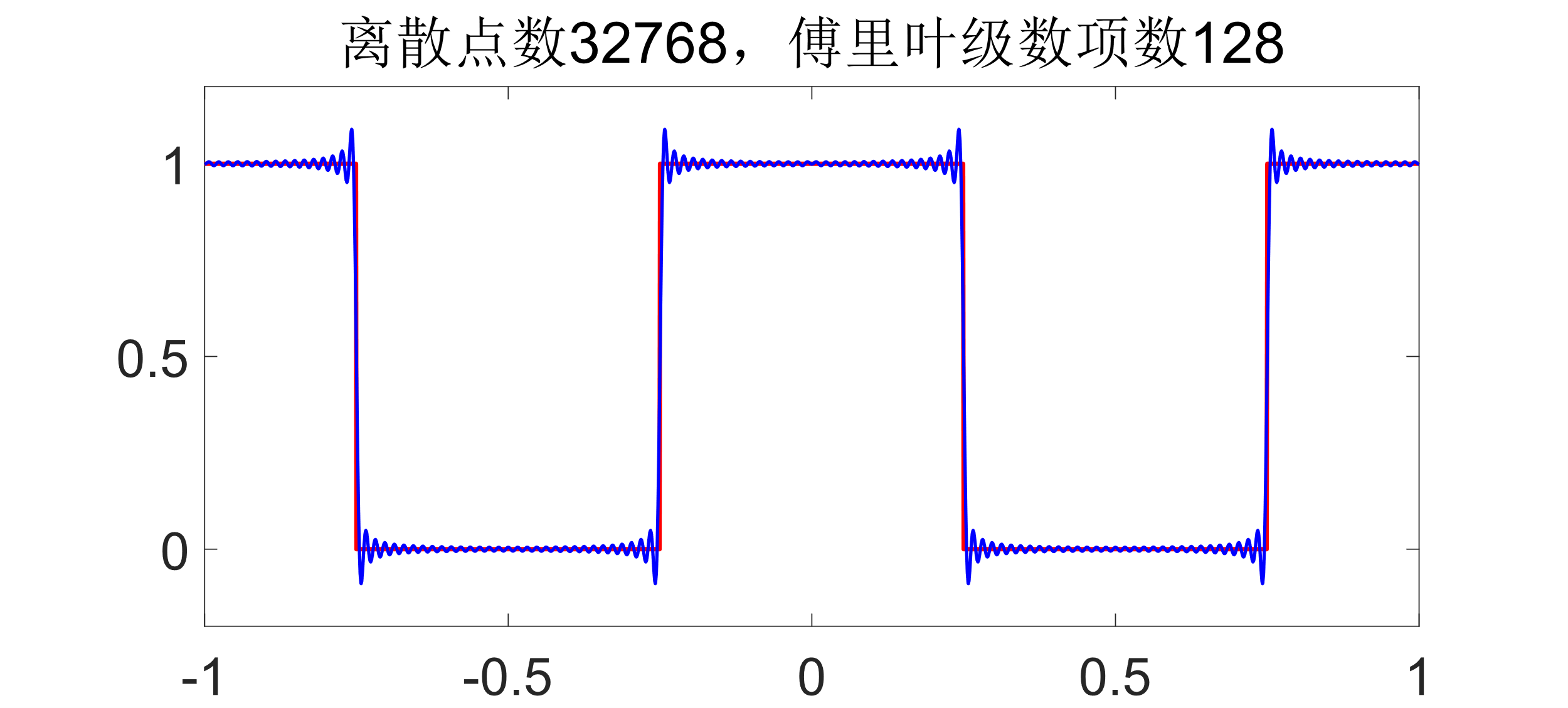
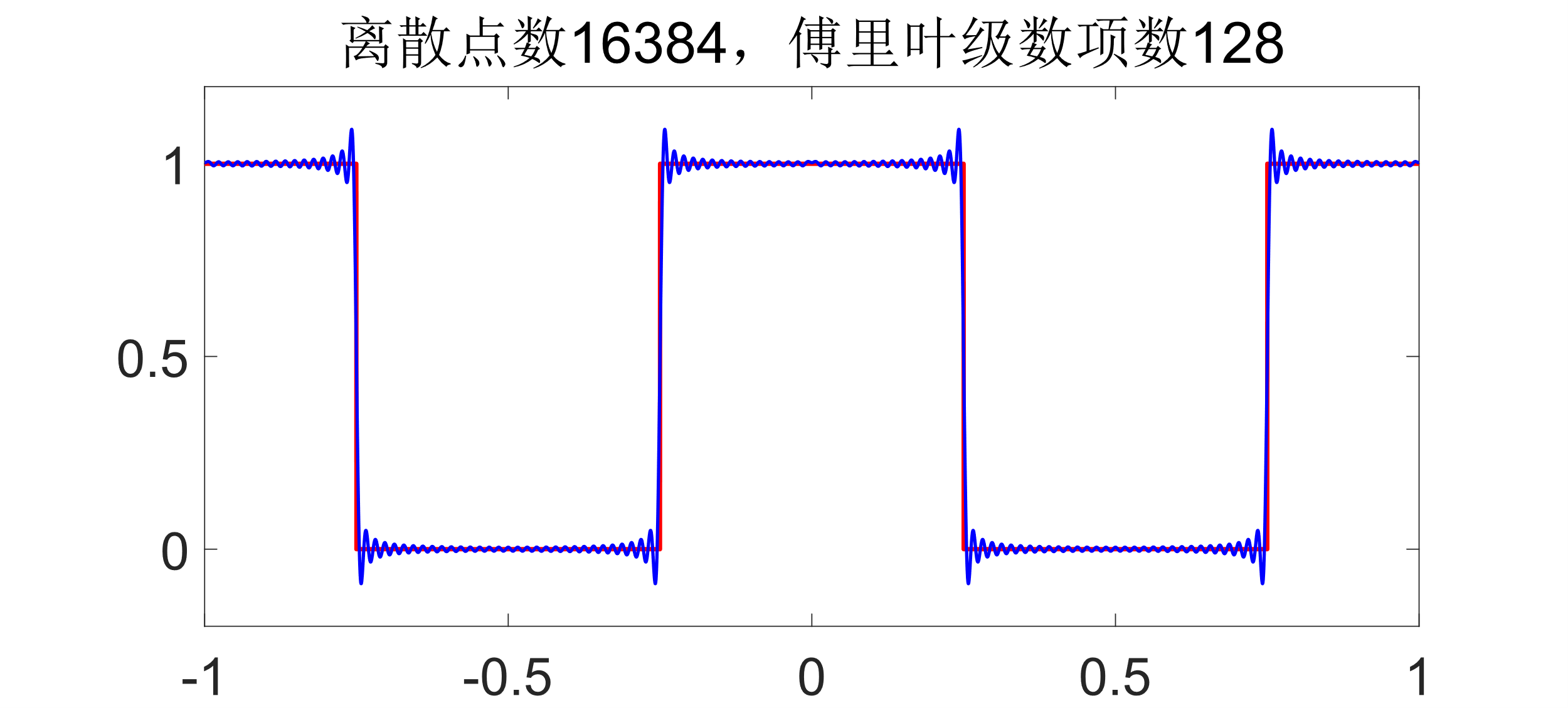
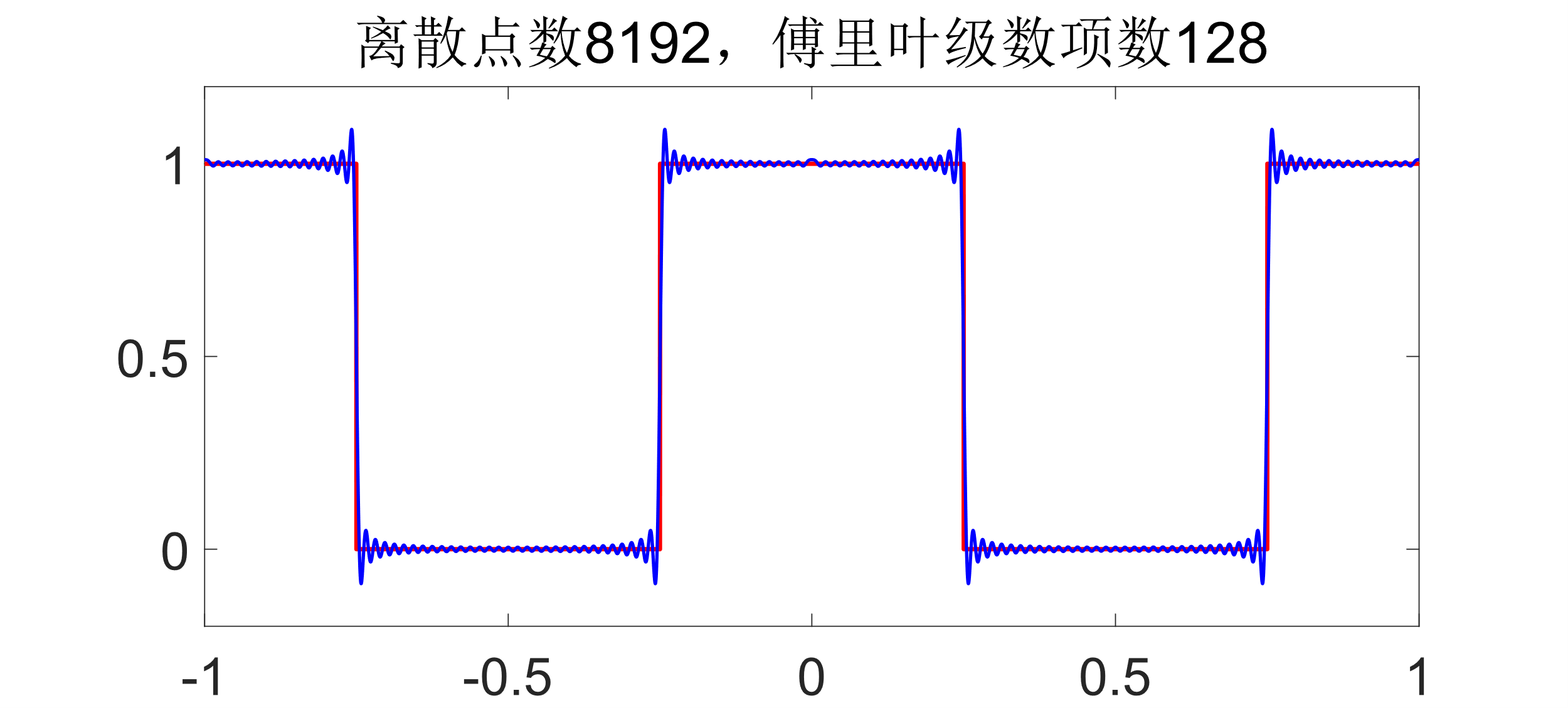
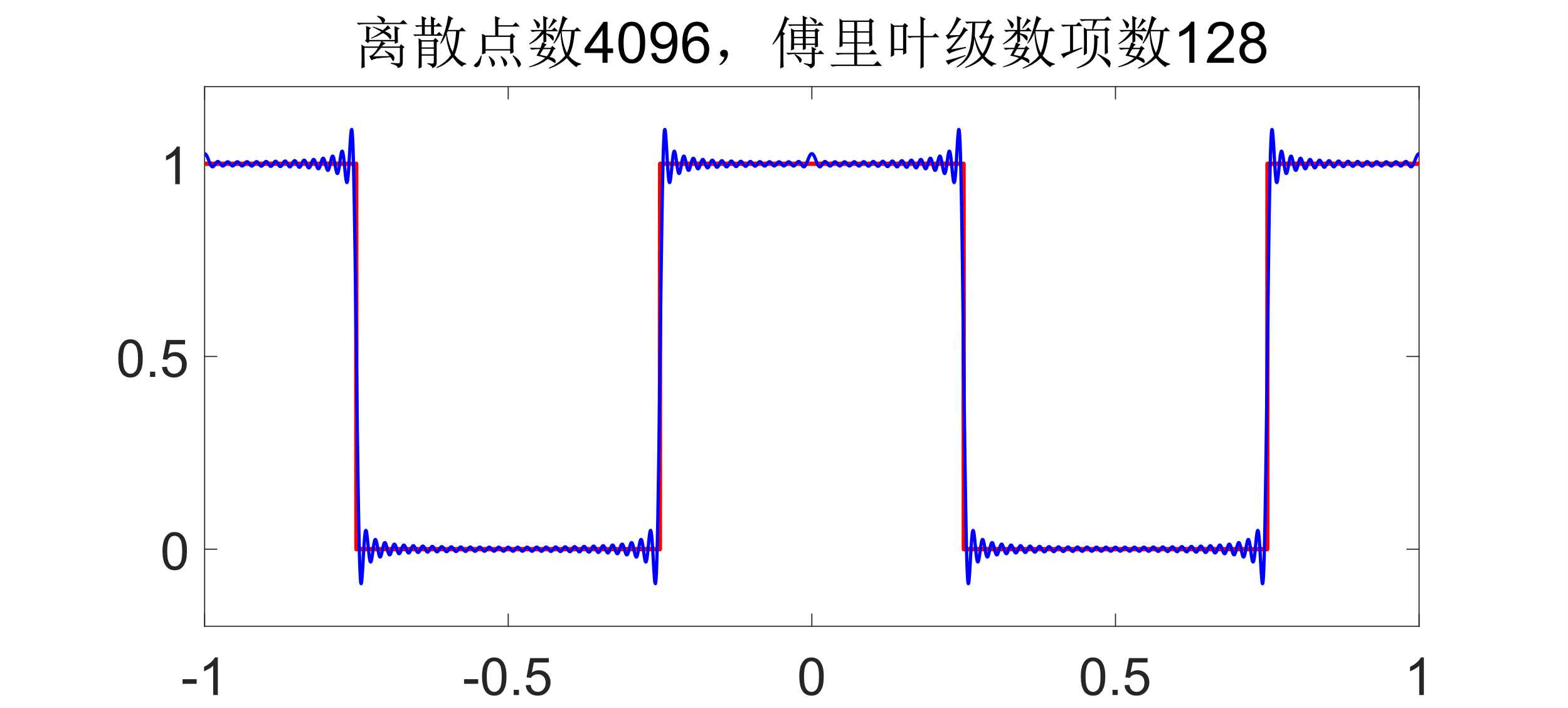
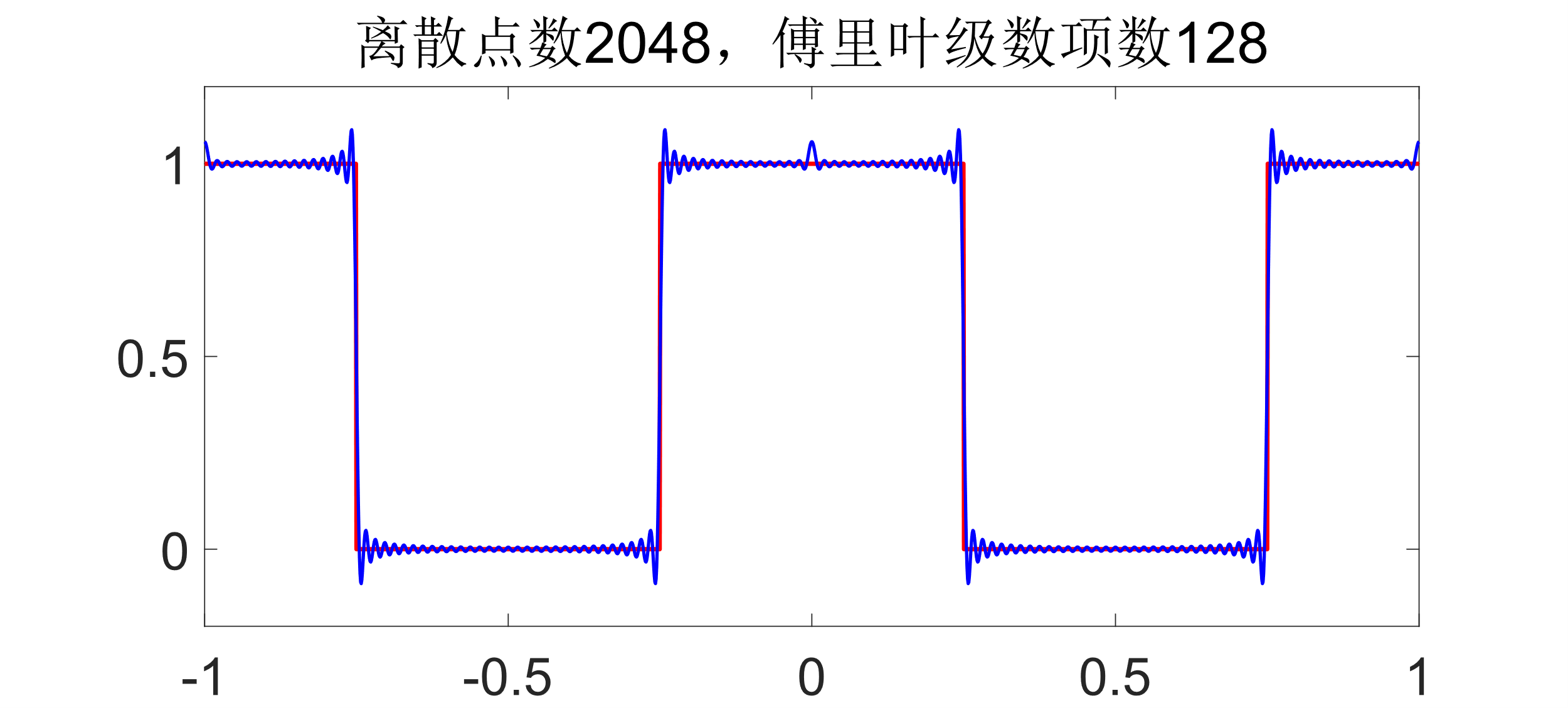
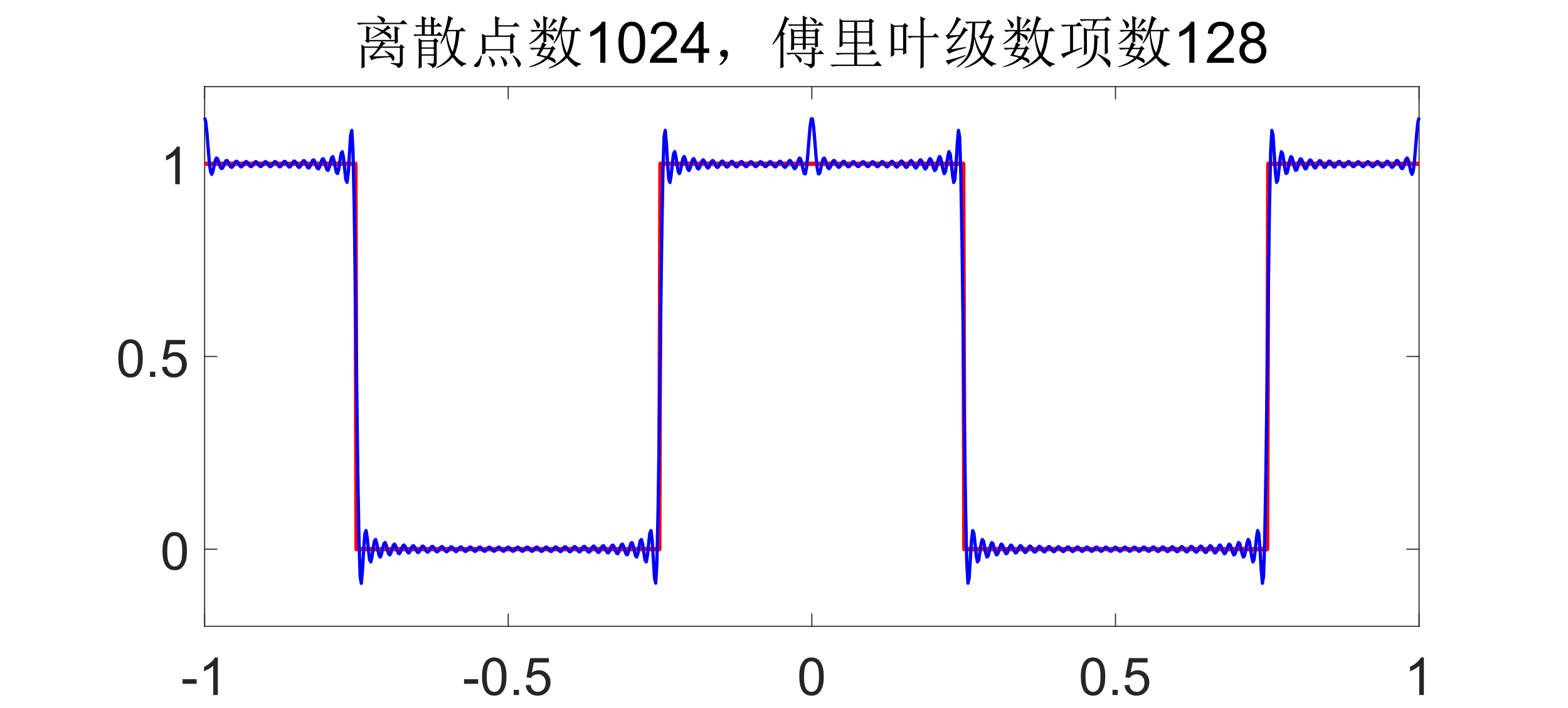
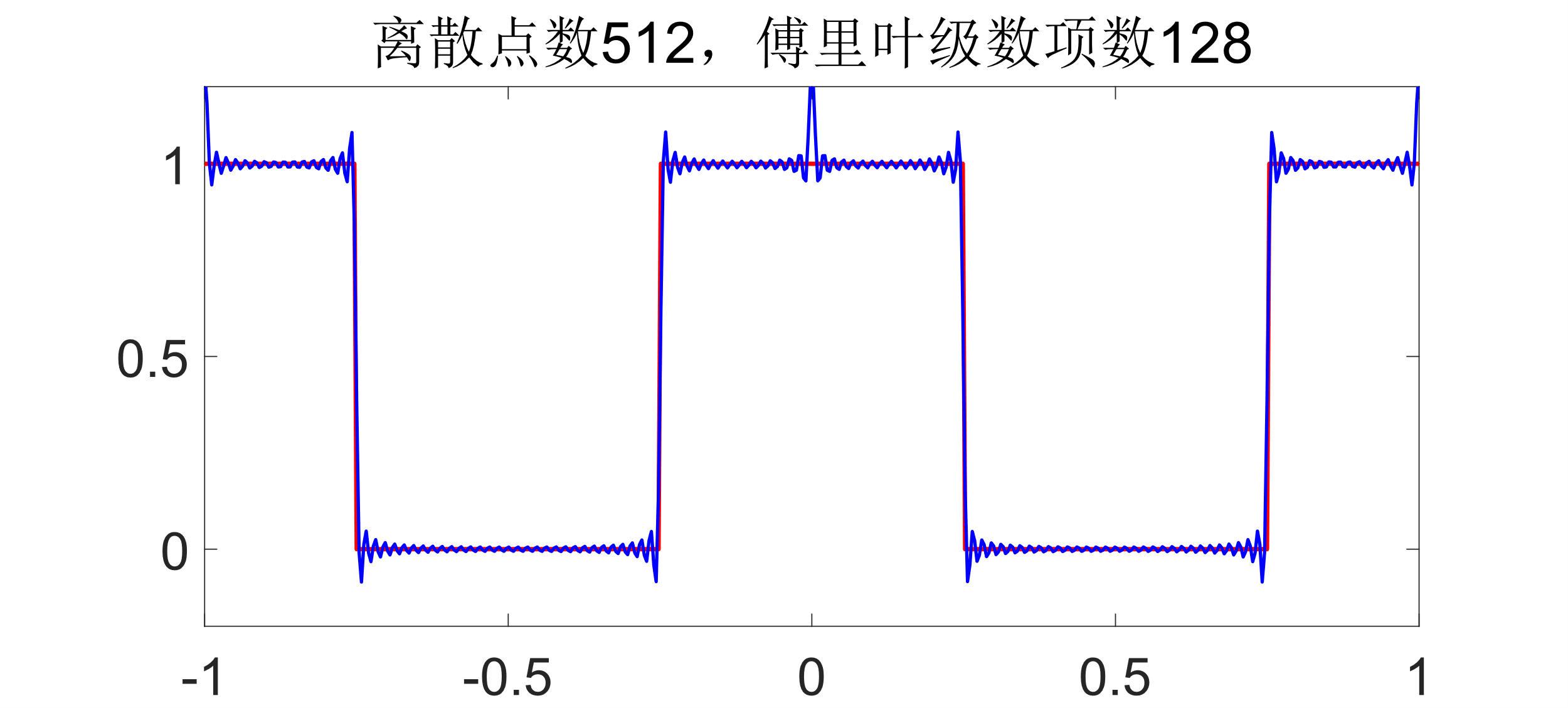
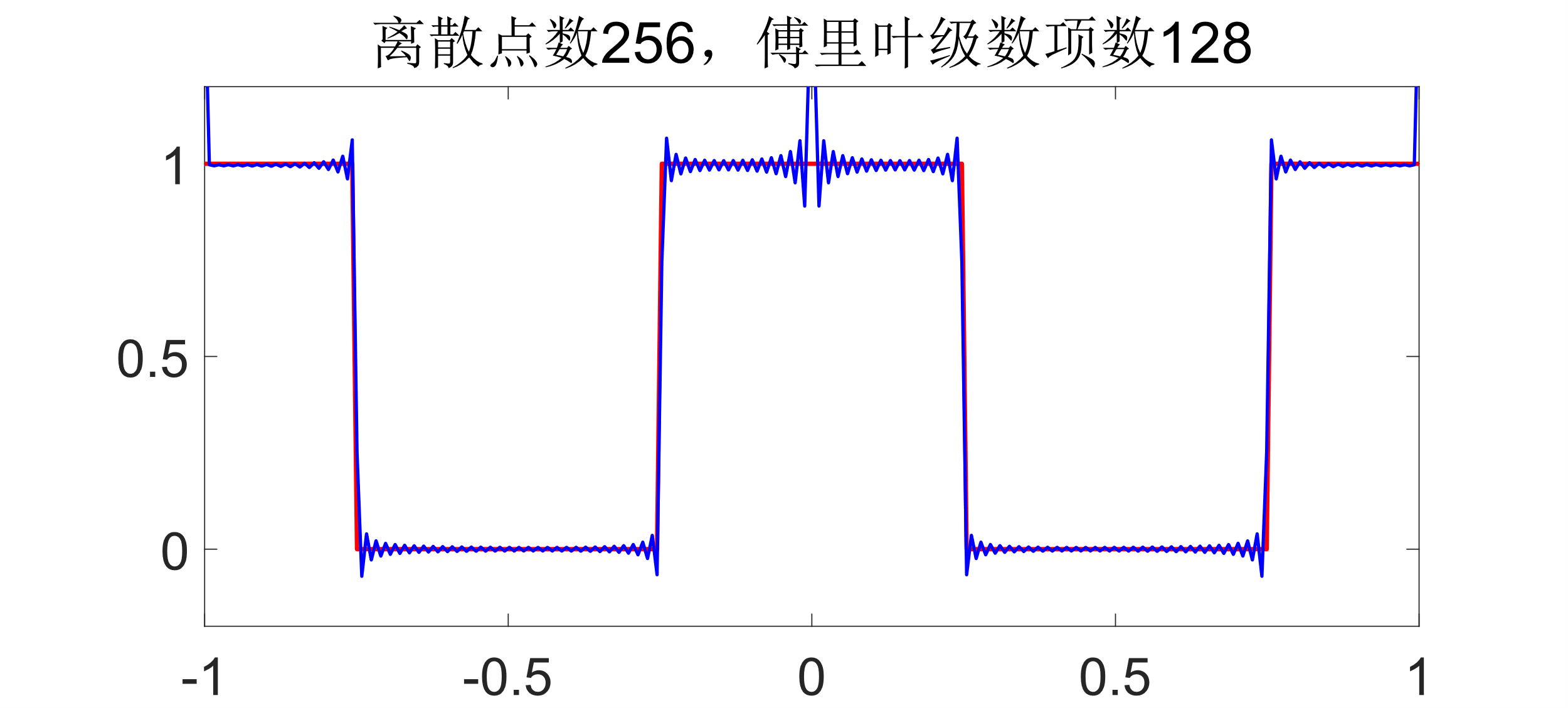
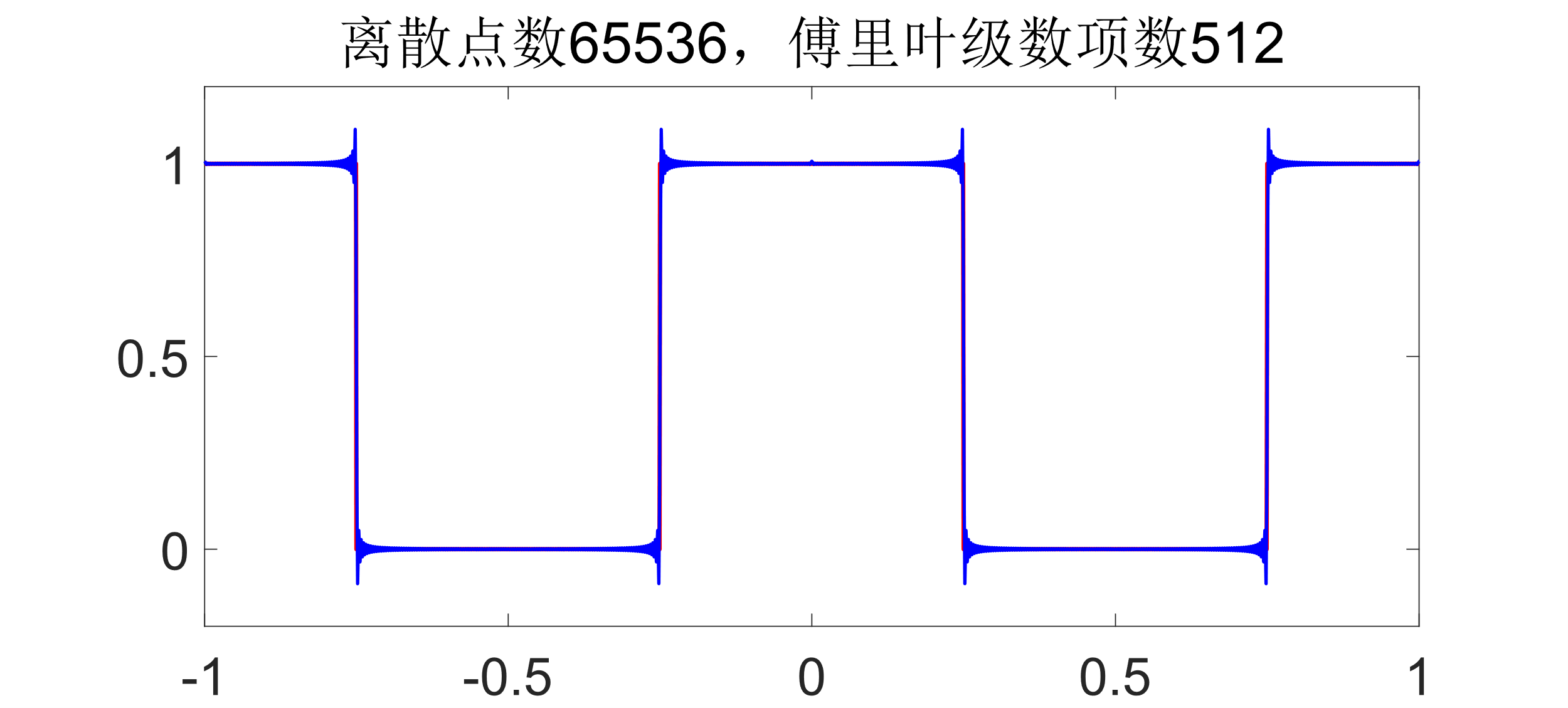
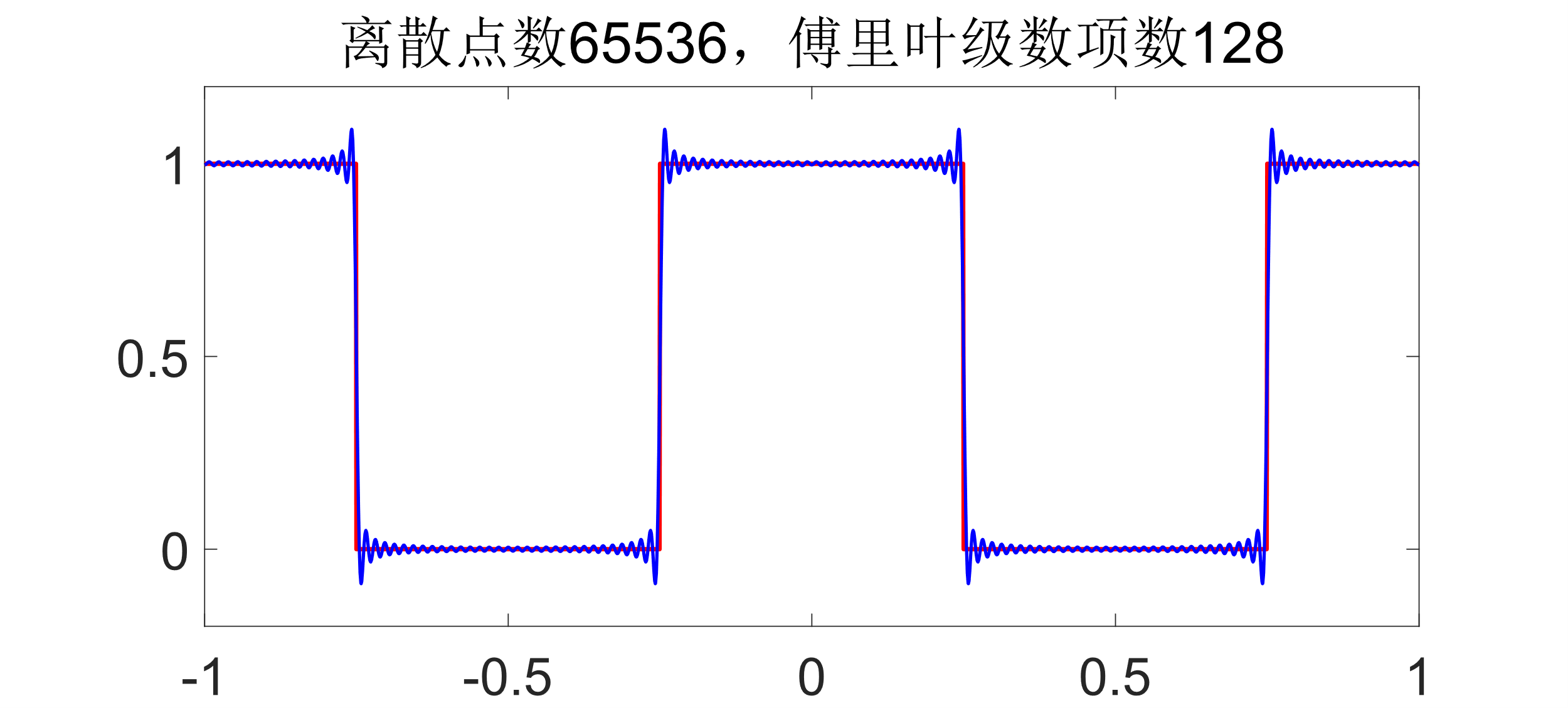
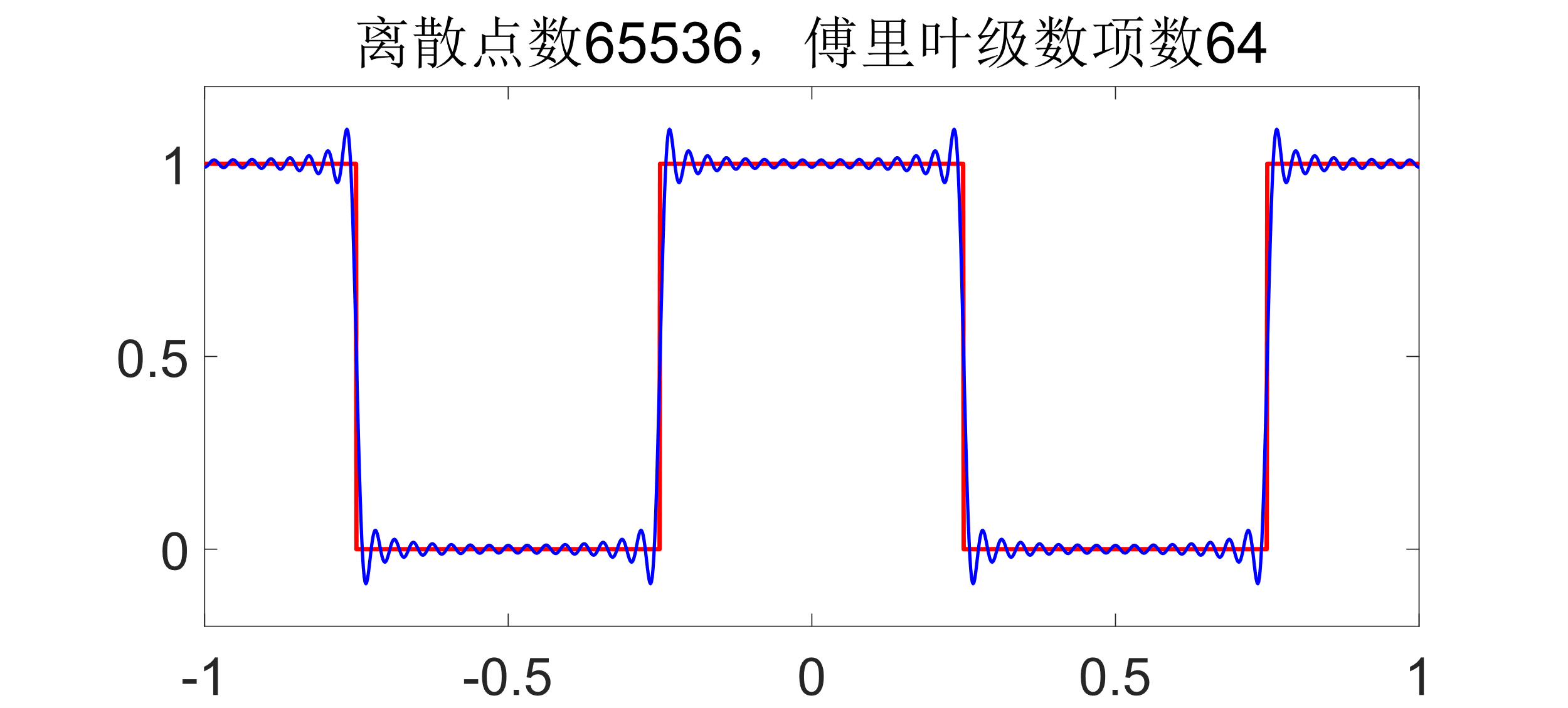
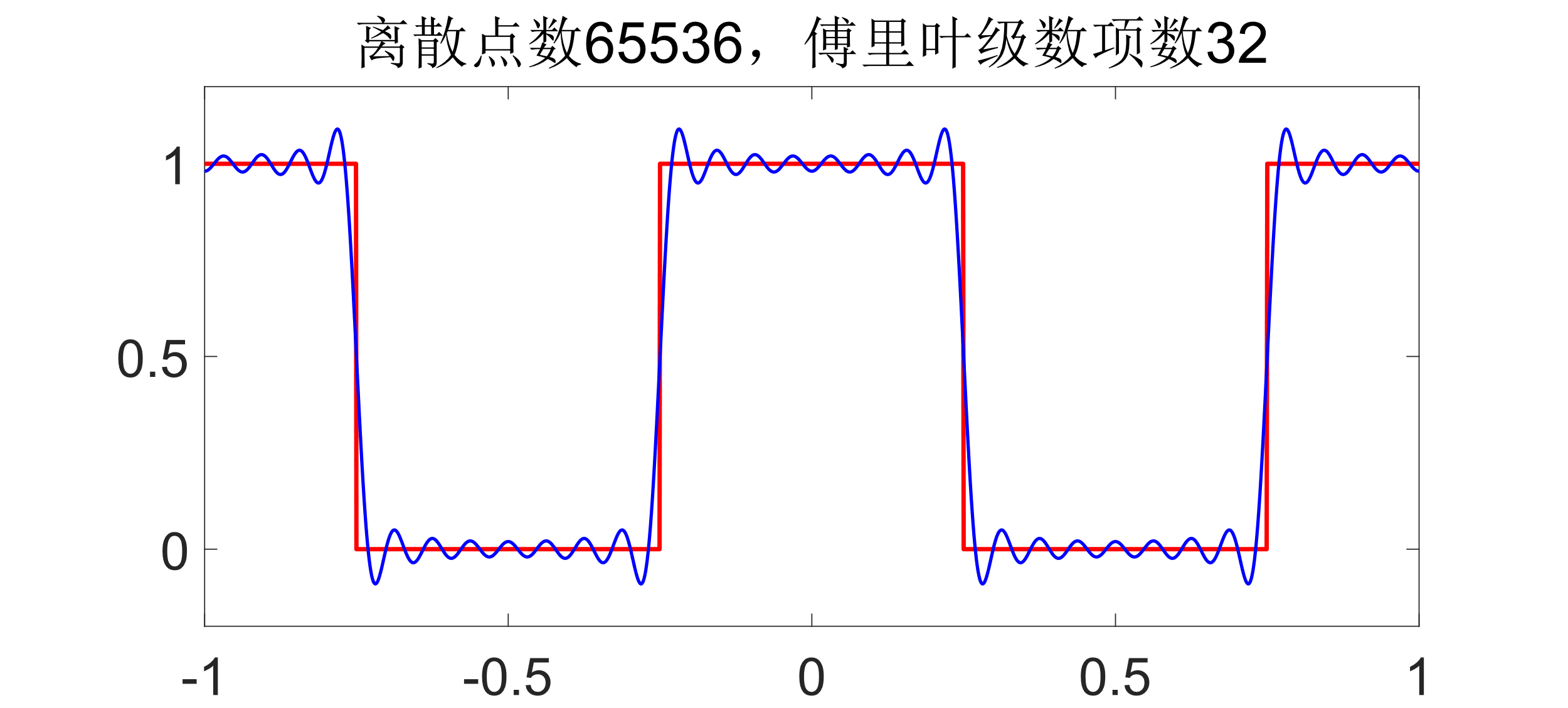
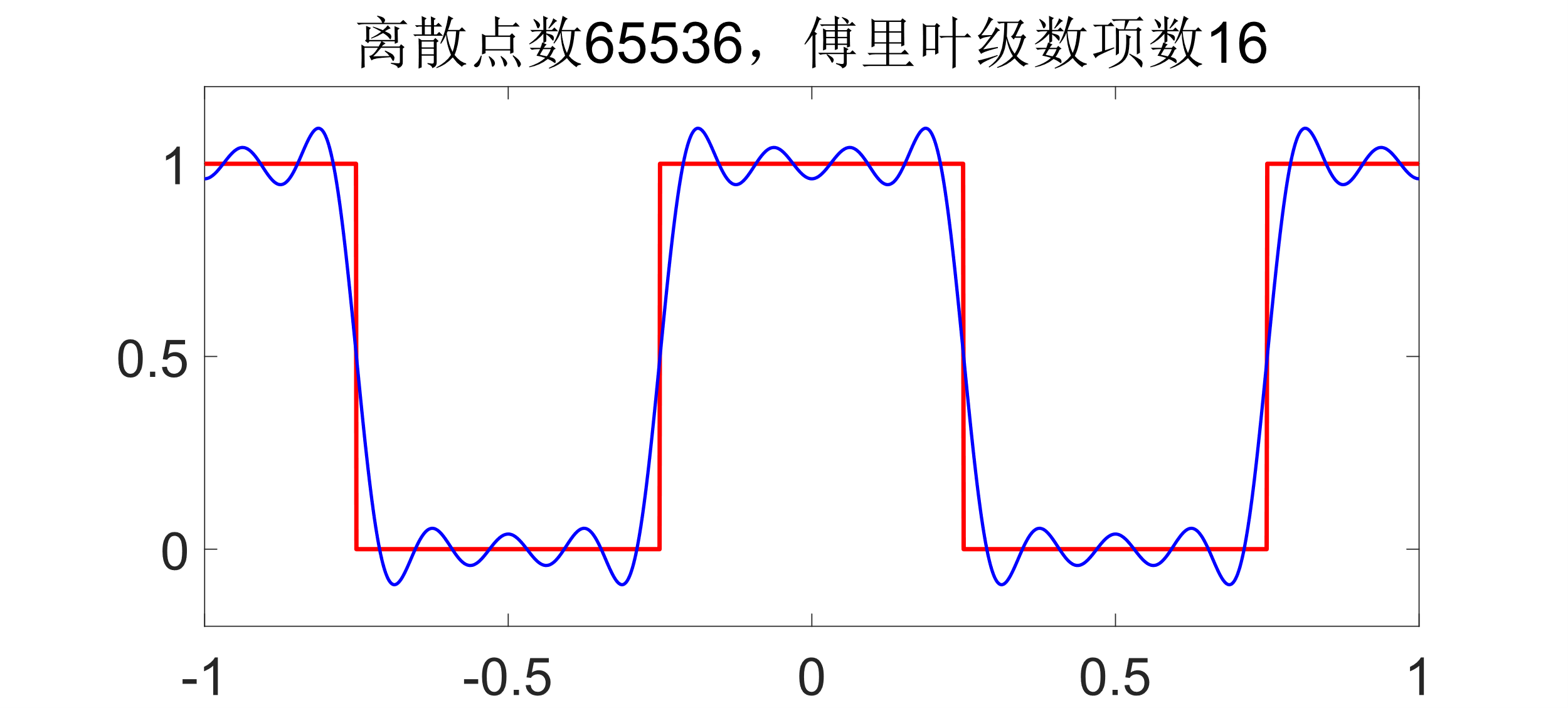
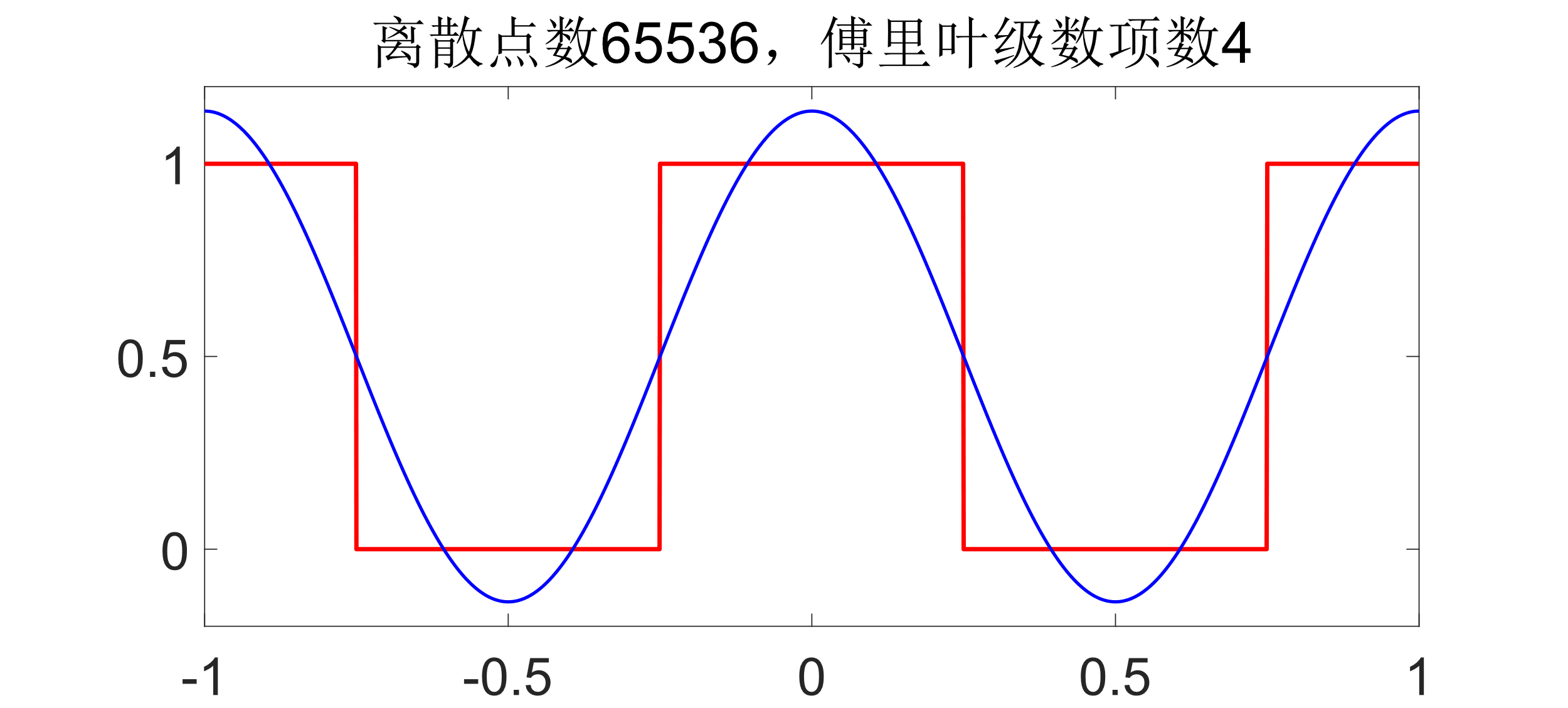


则，对于傅里叶级数：



### MATLAB实现

### 实验结果



很容易发现，在保持离散点数不变的情况下，随着傅里叶级数的项数的增加，级数逼近的结果越来越接近方波。但是，由于需要拟合的信号中存在阶跃，因此吉布斯现象永远存在。

由于实验中是使用累加求和的方式近似计算积分，因此离散点越多，则级数逼近的信号越接近方波。在离散点数较少时，在非信号的非阶跃处也出现了类似吉布斯现象的情况。

## 傅里叶变换与卷积

### 实验原理

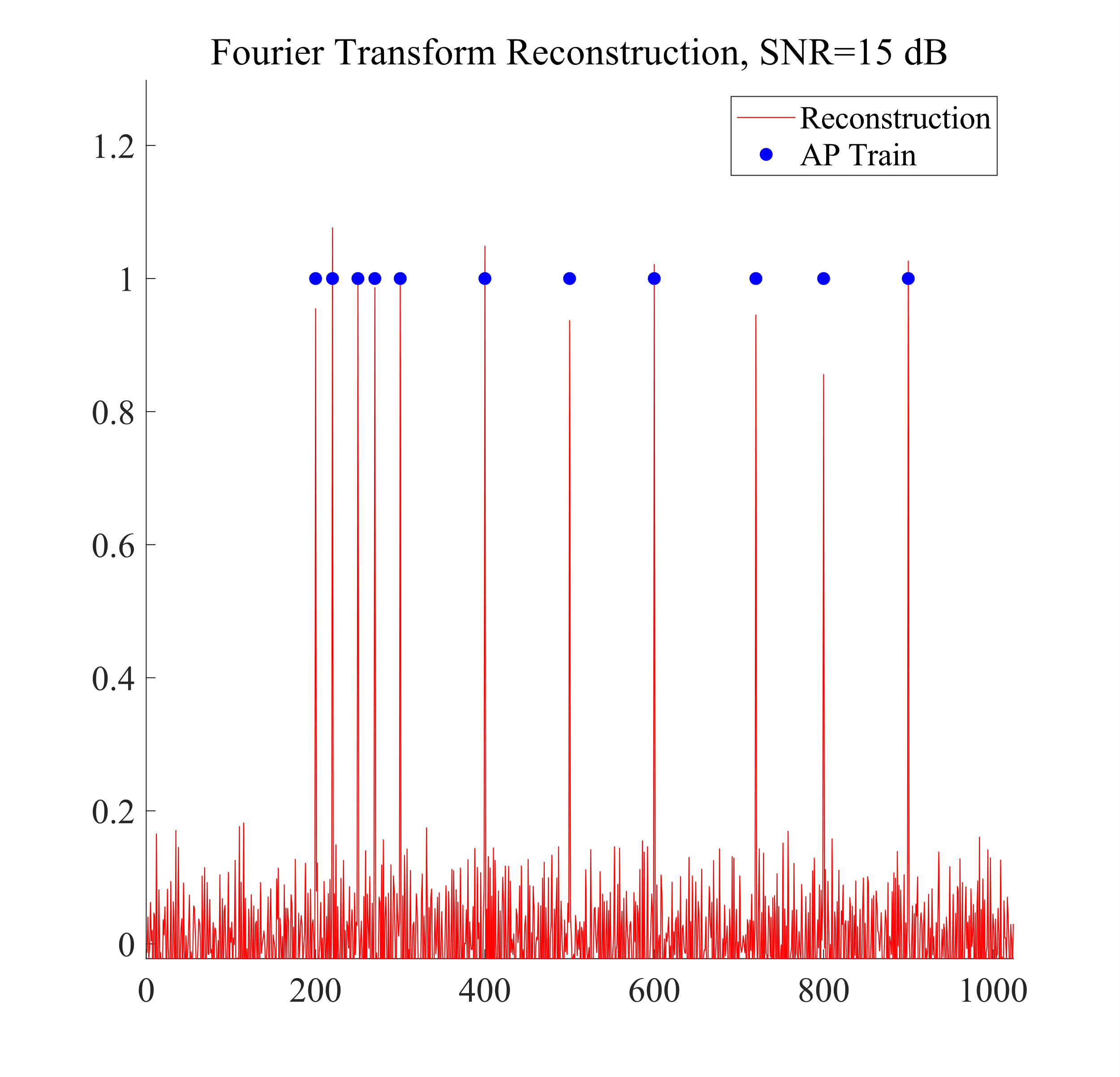
定义为循环卷积，、和为补零到相同长度后的信号。则循环卷积有以下性质：

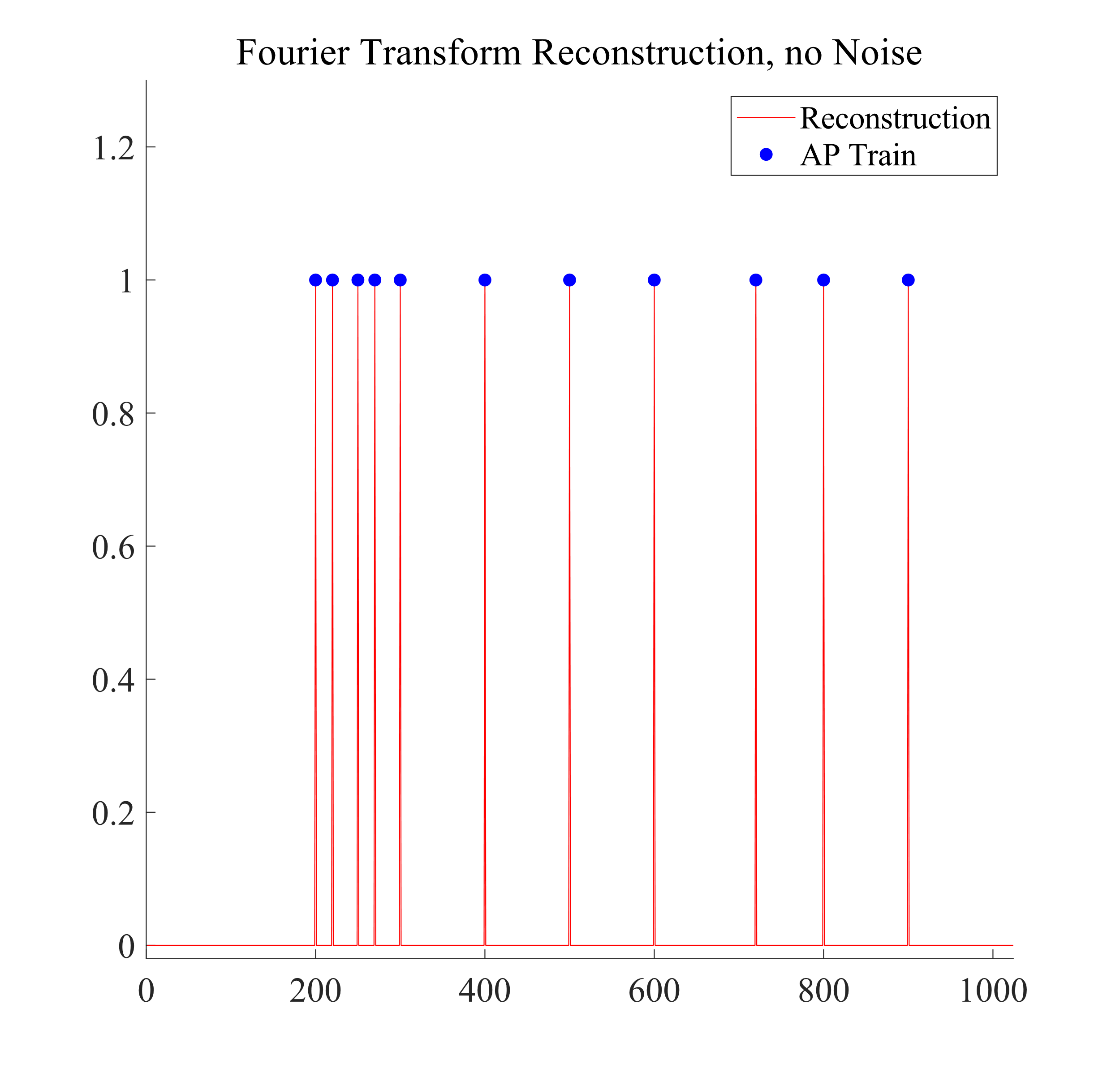


该性质也可用于计算反卷积。

### MATLAB实现

### 实验结果





对于无噪声的信号，通过傅里叶变换和循环卷积可以重建出误差很小的信号。

但是，对于带有噪声的钙信号，由于叠加的噪声并不是周期性信号，傅里叶变换无法对其进行有效的降噪。由于叠加的是白噪声，因此直接对频域信号进行计算时，甚至有可能增大高频部分的噪声。

## 一维离散傅里叶变换及其逆变换（DFT & IDFT）

### 实验原理

长度为的数字信号序列的离散傅里叶变换可以定义为：



和可以用列向量的形式表示：



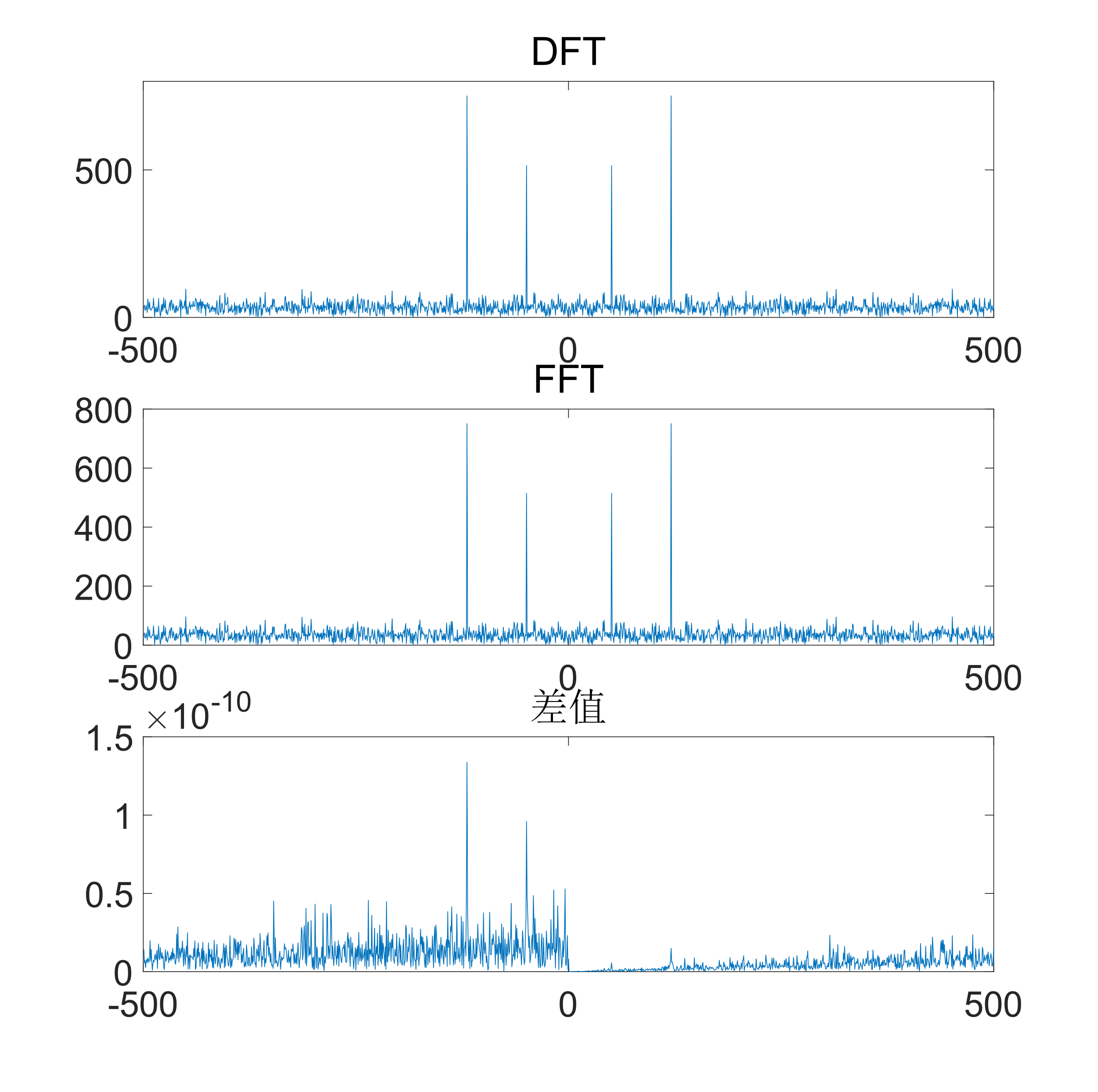
由式2-4定义的离散傅里叶变换及其逆变换也可以用矩阵表示为：

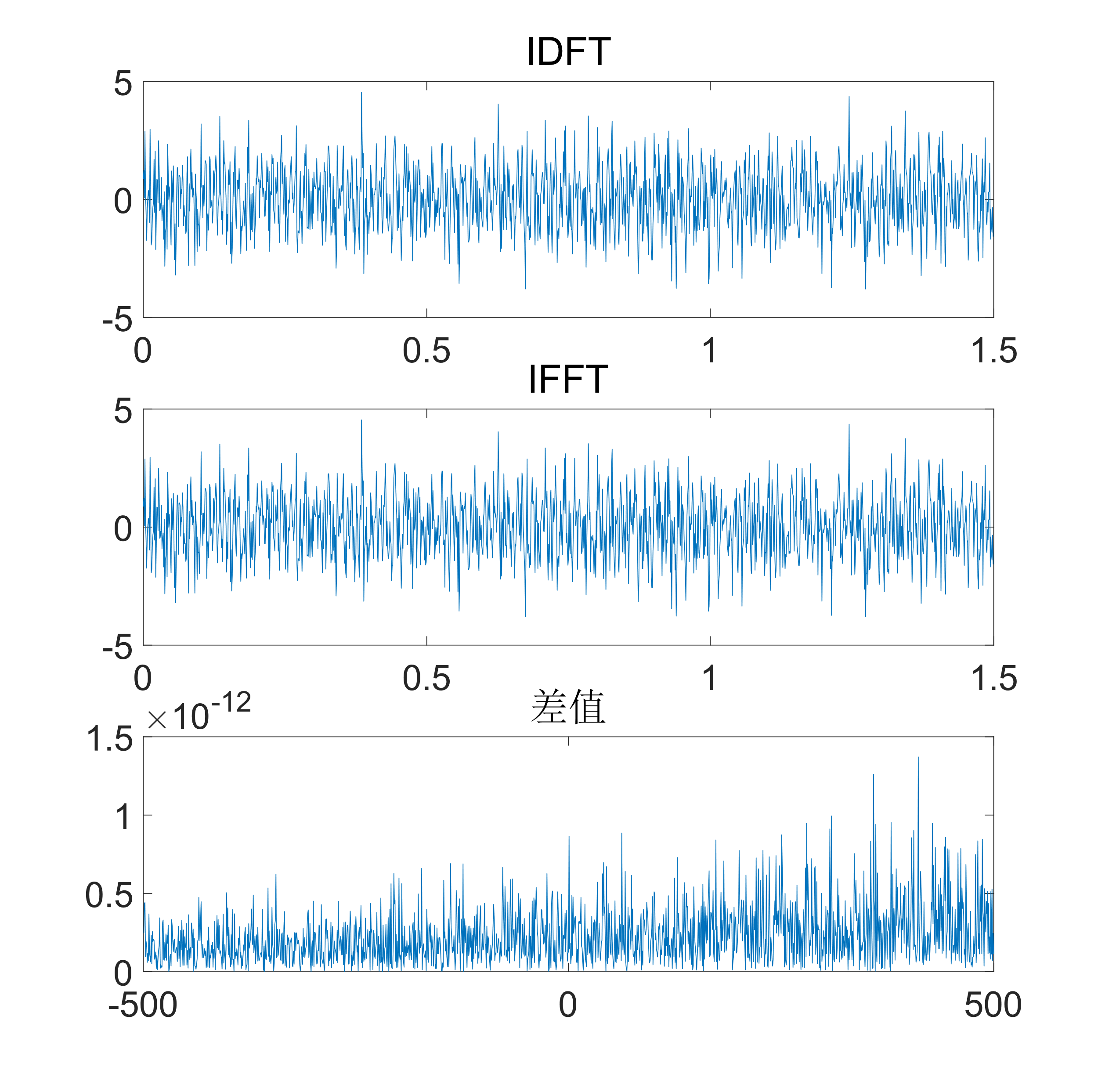




### MATLAB实现

### 实验结果





## 二维离散傅里叶变换及其逆变换（DFT2 & IDFT2）

### 实验原理

类似一维离散傅里叶变换，对于的二维离散信号，傅里叶变换和逆变换的定义为：



和一维傅里叶变换类似，傅里叶变换及其逆变换也可以用矩阵乘法表示：













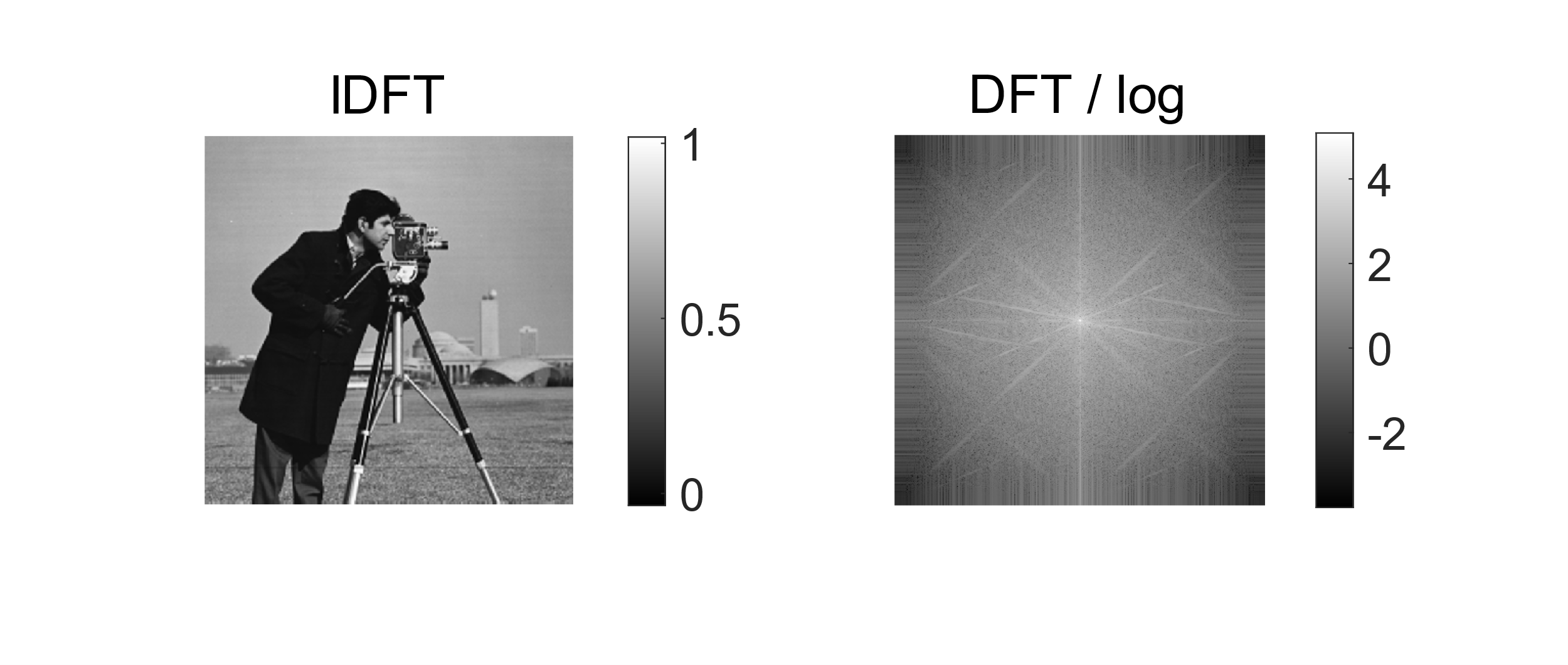
### MATLAB实现

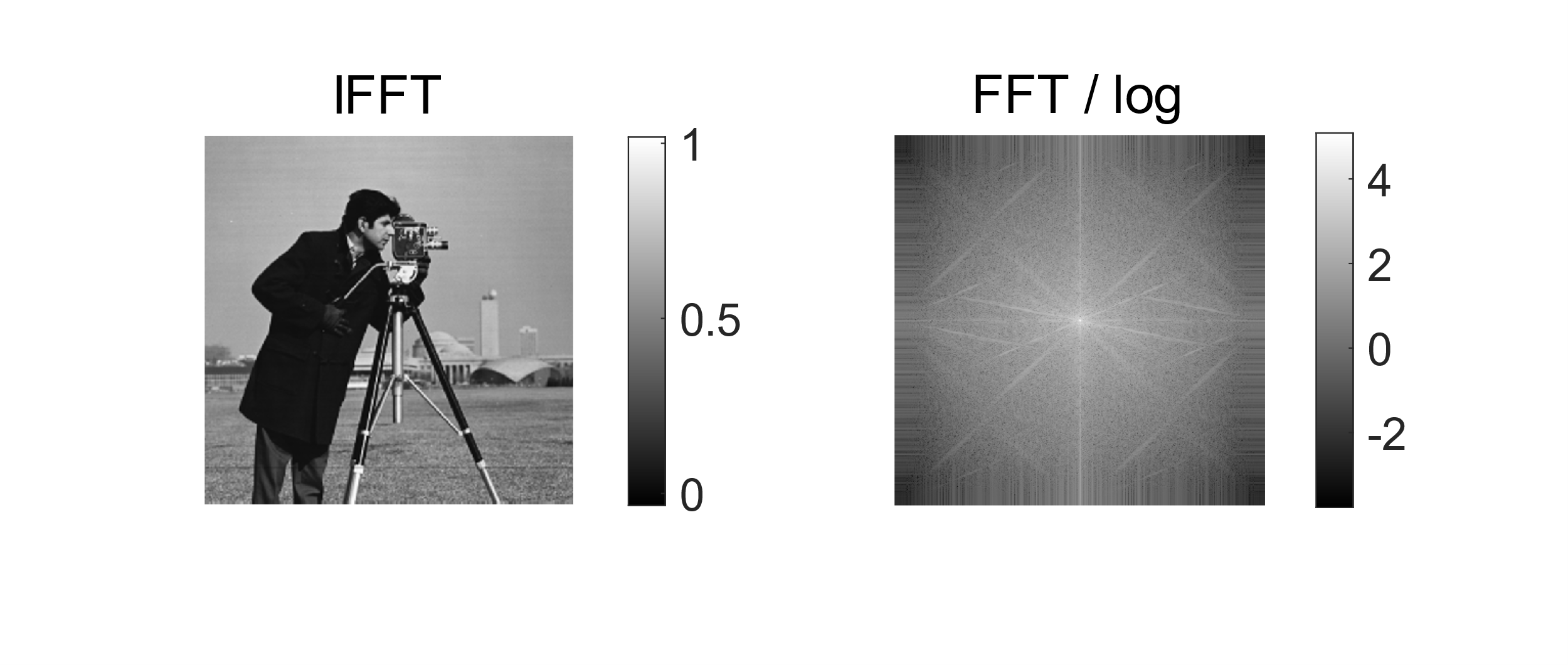
### 实验结果

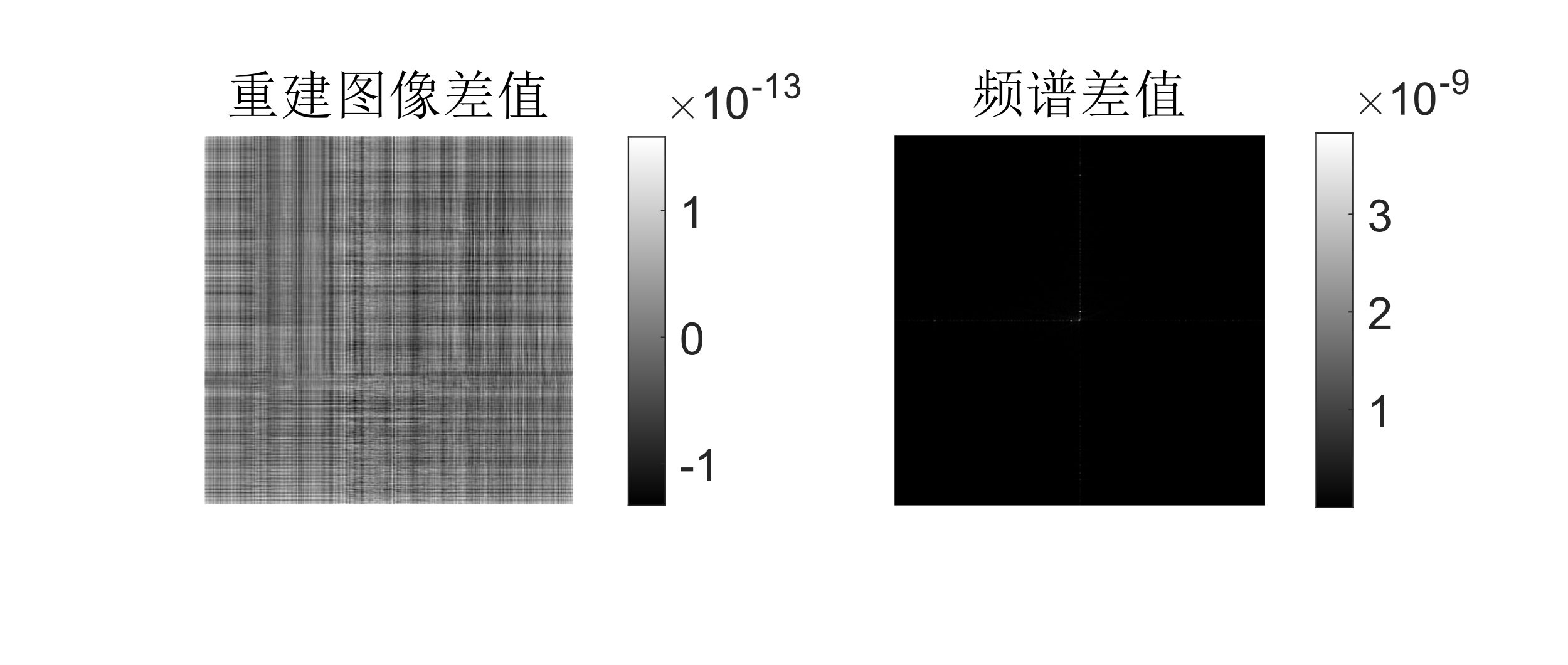
使用MATLAB中的图片cameraman.tif，并将其尺寸放大到，动态范围线性缩放到0-1进行实验。

使用MATLAB中的函数immse衡量图像之间的误差（均方误差），对于尺寸为的两张图片，其均方误差为：









通过库函数（fft2和ifft2）计算得到的结果与通过自定义函数得到的结果之间的均方误差为，可以认为是浮点数运算过程中引入的误差。