2022 春-计算方法-第六次上机作业说明文档

1 实验目的

通过快速傅里叶变换与快速傅里叶逆变换实现对给定函数的 Fourier 分析以及重建。

2 实验要求

通过快速傅里叶变换与快速傅里叶逆变换实现对给定函数的 Fourier 分析，函数 f 以及划分数 n 如下：

1. $f_1(t) = 0.7sin(2\pi \times 2t) + sin(2\pi \times 5t)$, $n = 2^4, 2^7$.
2. $f_2(t) = 0.7sin(2\pi \times 2t) + sin(2\pi \times 5t) + 0.3 \times random(t)$，$random(t)$ 为 $[0, 1)$ 区间内的随机数，$n = 2^7$。

其中 $t \in [0, 1)$，将 $[0, 1)$ 区间均匀划分为 n 份，$f_{1,k} = f_1(k/n)$, $k = 0, 1, \ldots, n-1$，$f_{2,k}$ 为 $f_{1,k}$ 加上一个随机扰动项，$f_{2,k} = f_{1,k} + 0.3r_k$，$r_k \in [0, 1)$ 为随机数。
程序实现完毕后，应撰写实验报告。实验报告中应包含如下内容：

1. 标题、学号、姓名。
2. 实验结果。
   - 对于每个 f 与 n，程序应输出所得到的快速傅里叶变换后的向量 $\mathbf{g}$，将 $\mathbf{g}$ 的每个分量的实部与虚部分开输出，并将输出结果截图加入到报告中。
   - 对于每个 f，将取不同的 n 获得的 $\mathbf{g}$ 的每个分量模长 $|g_i|$ 绘制成图像（为了使图像清晰，**如果将不同的 n 绘制到一张图上请绘制折线图**），图像横轴为频率（获取方式见附录），纵轴为 $|g_k|$，具体绘制方式参考 P32 图 9.4 以及本文件的附录。
   - 对于 f_1，绘制 f_1 离散化后的图像（$f_{1,k}$），以及快速傅里叶变换与快速傅里叶逆变换后的最终结果图像。不同的 n 需绘制在不同的图像上，绘制方式为折线图。
   - 对于 f_2，绘制 f_2 离散化后的图像（$f_{2,k}$），快速傅里叶变换与快速傅里叶逆变换后的最终结果图像，以及**快速傅里叶变换后取频率域前 25% 的系数进行快速傅里叶逆变换所得最终结果的图像**（参考书上例子以及附录）。绘制方式为折线图。
3. 结果分析。试分析采样数目 n 对结果的影响以及对重建（傅里叶逆变换）后的结果造成的影响。对于 f_2，分析去掉高频系数重建对结果造成的影响。如果算法无法运行，试分析原因。
4. 请简要地以文字方式说明实验结果和分析，不要只有图。

3 提交要求

3.1 提交方式

请提交源代码和实验报告。新建目录，并以“HW6-学号-姓名”方式命名，该目录下应包含如下内容：

- src\ （文件夹，存放你的源代码）
- report.pdf （你的实验报告）

将该文件夹以压缩包方式(压缩包名为“HW6-学号-姓名.zip”),发送到课程邮箱 computation_2022@163.com，**邮件标题**以同样方式命名。
请严格按照命名方式要求提交，不要交错邮箱，否则可能漏记成绩。

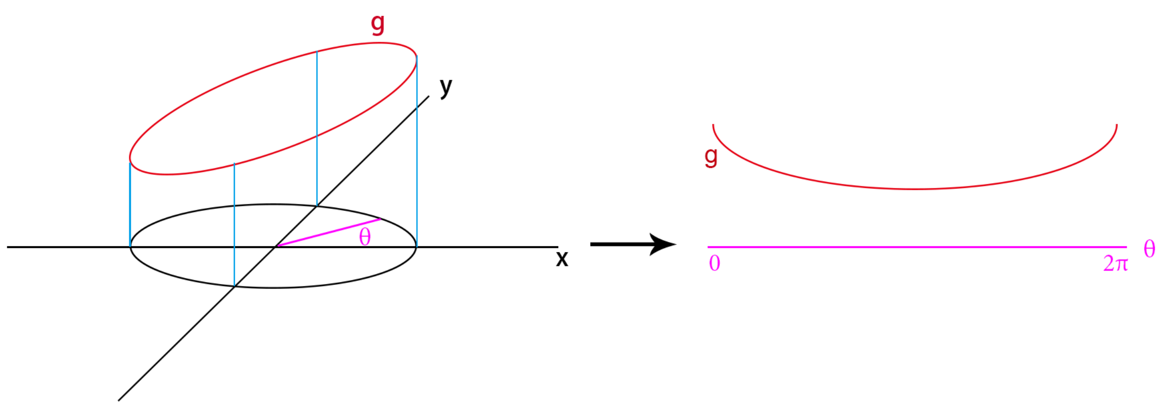
3.2 截止时间

在 5 月 23 日 23:59 分前提交，截止日期一周以后不再接受。**若有特殊情况请向助教说明**。

4 附录

4.1 角频域与频率域

离散傅里叶变换所得到的向量 **g** 为多项式 $p(z) = \frac{1}{n}\sum_{k=0}^{n-1} f_k z^k$ 在 n 个点 $1, \omega_n^1, \ldots, \omega_n^{n-1}$ 处取值构成的，w_n^i 为均匀分布在复数域单位圆上的 n 个点，角度值 $2k\pi/n$ 即为角频率，如图：



采样频率 $f_s = 1/\Delta t = n$，角频率 $2k\pi/n$ 对应的频率值为 $\frac{k}{n} * f_s = k$.

4.2 快速傅里叶逆变换 (IFFT)

Algorithm 1 IFFT

$n \leftarrow length[f]$
if $n == 1$ **then**
  return f
end if
$\omega_n \leftarrow e^{i2\pi/n}$
$\omega \leftarrow 1$
$\mathbf{f}^0 \leftarrow (f_0, f_2, \ldots, f_{n-2})$
$\mathbf{f}^1 \leftarrow (f_1, f_3, \ldots, f_{n-1})$
$\mathbf{g}^0 \leftarrow$IFFT$(\mathbf{f}^0)$
$\mathbf{g}^1 \leftarrow$IFFT$(\mathbf{f}^1)$
for $k \leftarrow 0$ to $n/2-1$ **do**
  $g_k \leftarrow \mathbf{g}_k^0 + \omega \mathbf{g}_k^1$
  $g_{k+n/2} \leftarrow \mathbf{g}_k^0 - \omega \mathbf{g}_k^1$
  $\omega \leftarrow \omega\omega_n$
end for
return g

4.3 取低频信号

根据上述角频率与频率的关系，频率域取前 25% 系数等价于角频率域取前 25% 系数，示例：

```cpp
vector<complex<Double>> g = FFT(f);
vector<complex<Double>> g1;
for (int i = 0; i < n; i++) {
    if (i < n * 0.25)
        g1.push_back(g[i]);
    else
        g1.push_back(complex<Double>(0.0, 0.0));
}
```

4.4 所需图像

当多个图像在一张图上时请选择折线图。

1.f_1 取不同 n 获得的 **g** 图像，横轴频率纵轴 $|g_i|$.

2.f_1 原向量与 $IFFT(FFT(f_1))$ 图像，横轴为 $[0,1)$，纵轴为实部的值（因为是复向量）。对两个 n 画两张图。

3.f_2 原向量与 $IFFT(FFT(f_2))$ 图像，以及快速傅里叶变换后取频率域前 25% 的系数进行快速傅里叶逆变换所得结果的图像，横轴为 $[0,1)$，纵轴为实部的值。只需绘制 $n=2^7$，可以把三条折线画到一张图。

4.5 代码提示

本次作业涉及复数运算的代码。C++ 提供了对复数运算的支持，一个示例如下：

```cpp
#include <iostream>
#include <complex>
#include <vector>
using namespace std;
#define PI 3.1415926535
typedef double Double;

vector<complex<Double>> FFT(const vector<complex<Double>>& f) {
    int n = f.size();

    //complex<Double> w(real, imag)
    complex<Double> wn = exp(-2 * PI / n * complex<Double>(0.0, 1.0));
    complex<Double> w(1, 0);

    //TODO......
}
```