Due Date : August 7 2018

Problem set up

Suppose we are given the first $N+1$ Fourier coefficients of a piecewise function $f(x) \in L^2[0,2\pi]$:

$$\hat{f}_k = \frac{1}{2\pi}\int_0^{2\pi} f(x)\exp(-ikx)\,dx \quad , \tag{1}$$

$k = -\frac{N}{2}, \cdots, \frac{N}{2}$. We want to approximate the true function $f(x)$ on a set of equally spaced grid points given by

$$x_j = \frac{2\pi}{N}j \quad , \quad j \in [0, N-1] \quad .$$

The Fourier partial sum approximation of f is given by

$$\mathcal{P}_N f(x) = \sum_{k=-N/2}^{N/2} \hat{f}_k \exp(ikx) \quad . \tag{2}$$

When $f(x)$ is piecewise smooth, the approximation (2) will result in the *Gibbs phenomenon.* The goal of this homework set is to observe the effects of the Gibbs phenomenon and to try to reduce its effects using (i) filtering and (ii) ℓ_1 regularization.

To do the ℓ_1 regularization, you will also need the formula for the discrete Fourier coefficients of $f(x)$:

$$\tilde{f}_k = \frac{1}{Nc_k}\sum_{j=0}^{N-1} f(x_j)\exp(-ikx_j) \tag{3}$$

where $c_k = 2$ if $k = \pm\frac{N}{2}$ and $c_k = 1$ otherwise. This leads to the approximation of f:

$$\mathcal{I}_N f(x) = \sum_{k=-N/2}^{N/2} \tilde{f}_k \exp(ikx) \quad . \tag{4}$$

Observe we can write (3) as a matrix vector multiply:

$$A\mathbf{f} = \tilde{\mathbf{f}},$$

where $\mathbf{f}$ is an N vector $\{f(x_j)\}_{j=0}^{N-1}$. The length of $\hat{\mathbf{f}}$ here is $N+1$ (for the problem below you can "cut off" the last coefficient $\tilde{f}_{N/2}$ if you desire a square matrix.)

Note that in some instances $\mathcal{I}_N f(x) = \mathcal{P}_N f(x)$ and it is always true that $\mathcal{I}_N f(x_j) = f(x_j)$, which is called the interpolation property.

To test your MATLAB code, you should consider the function $f(x) = \cos 2x$, which is a periodic smooth function. Calculate the Fourier coefficients (do *not* use a quadrature formula!) using (1), and be sure that (2) produces good results. Here good means perfect (up to machine error) so long as N is large enough. You an try something like $N = 32$ so that $N/2 = 16$. You should not turn this part in. Make sure you understand why this result should be perfect. You should also check to make sure your formula for $\tilde{f}_k$ works. To do this, calculate $|\hat{f}_k - \tilde{f}_k|$ and plot this error (or plot k vs $\log|\hat{f}_k - \tilde{f}_k|$) and make sure that you get what you expect. The error should be close to machine precision so the log should look like -16 (or at least -8 if you use single precision). For your information, there is a log plot in the matlab code I sent out earlier.

Once you are sure your code is working properly, you can answer the following questions. You are free to modify $f(x)$, the choices of N and the filter. Play around with the code and see what comes up. You do not need to do the items marked **BONUS** but you may be interested to see what you get.

1. Define $f(x) = \begin{cases} -x & 0 \leq x < \pi \\ 2\pi - x & \pi \leq x \leq 2\pi \end{cases}$

   Calculate the Fourier coefficients $\hat{f}_k$, $k = -\frac{N}{2}, \cdots, \frac{N}{2}$ and the Fourier partial sum $\mathcal{P}_N f$ for $N = 16, 32, 64$. Plot two graphs: One graph with the exact solution and $\mathcal{P}_{32} f$, and the other with x_j vs. $\log |f(x_j) - \mathcal{P}_N f(x_j)|$, for $N = 16, 32, 64$. Comment about your results.

2. Now consider mitigating the Gibbs ringing effect using the raised cosine filter: $\sigma(\eta) = \frac{1}{2}(1 + \cos(\pi\eta))$. Here $\eta \in [0, 1]$ and we can write the discrete form $\sigma_k = \frac{1}{2}(1 + \cos(\pi 2k/N))$, since k goes from $-N/2, \cdots, N/2$. The *filtered* Fourier partial sum is given by

   $$\mathcal{P}_N^{\sigma} f(x) = \sum_{n=-N/2}^{N/2} \sigma_k \hat{f}_k \exp(ikx) \quad .$$

   Repeat the process as in the non-filtering case and make the corresponding graphs. Comment on your results.

   BONUS: Try using $\sigma(k) = \tau(k)$, the edge detection filter desribed in class. Can you recover the discontinuity locations of f?

3. Now consider using the inverse method approach. In this case we will first assume that $\tilde{\mathbf{f}} \approx \hat{\mathbf{f}}$ which are vectors indexed by k calculated in (3) and (1) respectively. This is equivalent to saying that a finite sum closely approximates an integral. Hence we get the following optimization problem:

   $$min_f ||Lf||_1$$

   subject to

   $$||A\mathbf{f} - \hat{\mathbf{f}}||_2 \leq \epsilon$$

   where A is given above. You should again plot these same two graphs as before. (Hint: you can use the code I sent you today by replacing P in the code with A here, as well as the input data. You will also have to adjust the parameters. If you want the ℓ_1 operator to be TV (or first order finite difference) choose $m_{spa} = 1$.)

   BONUS: It is up to you to try (i) other functions (with multiple discontinuities, more variation, etc); (ii) different ℓ_1 operators (choose $m_{spa} = 2$ or $m_{spa} = 3$ for more variation); (iii) different row selector matrices (so in this case we assume we don't have all of the data).