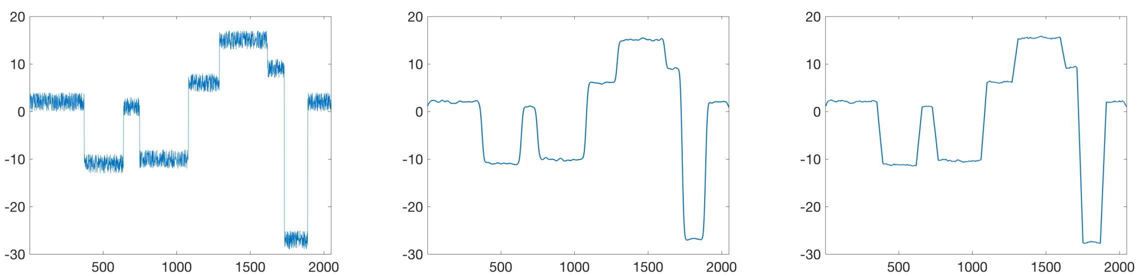
Math 369: Lab 8

Prof. Schaeffer, TA: Linan Zhang

Background: For Lab 8, you will write a program that takes a noisy 1D signal and outputs a "smoother" version by applying a filter to the input data. The input data (left) and sample outputs (middle and right, generated using two different filters) are plotted below. In Lab 7, we used the MATLAB built-in functions to create a 2D Gaussian filter and apply it to an image. Here we directly define a 1D filter and obtain a smoother signal via an integral equation. Filtering is widely applied in the fields of: image/signal processing, computer vision, numerical methods for spatio-temporal systems (partial differential equations), etc. Also, filtering and smoothing are fundamental to the topic of deep learning.



Problem: Let $f(t)$ be the noisy input signal and let $u(t)$ be a smooth approximation. To find $u(t)$, you will write an algorithm that approximates the following equation:

$$u(t) := \int_1^T f(s)G(t-s)\,\mathrm{d}s \tag{1}$$

where G in the above equation is known as a filter. The Lab 8 assignment consists of two parts:

(a) Write an algorithm (`MyGaussianFilter.m`) that solves Equation (1) using the composite trapezoidal rule with the Gaussian filter:

$$G(x) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right),$$

where $\sigma > 0$ is a parameter you can tune.

(b) Write an algorithm (`MyAverageFilter.m`) that solves Equation (1) using the composite midpoint rule with the average filter:

$$G(x) := \begin{cases} 1/(2r), & x \in [-r, r] \\ 0, & \text{otherwise}, \end{cases}$$

where $r > 0$ is a parameter you can tune.

For each part, you should: (1) tune the parameter and obtain a reasonable approximation u, (2) turn in your code and a plot for u, (3) use vectorization and have no more than one loop.