Convolution

<u>PREPARATION:</u> Convolution is the subject of this problem set. There are two flavors of convolution: **open** and **circular**. Open and circular convolution can be viewed as having absorbing and circular boundary conditions, respectively. **For financial applications, always use open convolution**, never circular. This is because the circular form, being periodic, is non-causal. That is, an echo of events that happen in the future will appear in the past.

The Matlab convolution function `conv` is a directly computed open convolution. In R convolution is circular by default, but you can pass ..., `'open'` as an argument. Unlike Matlab, R uses a Fourier method to evaluate convolution.

Although I have not yet covered in class the connection between convolution and Fourier transforms, a good comparison of the two methods is available here: http://blogs.mathworks.com/steve/2009/11/03/the-conv-function-and-implementation-tradeoffs.

<u>IMPULSE RESPONSES:</u> The following impulse responses will be used in the problems. I recommend that you write a function or functions that can return a time-series for each response given one or more parameters. I recommend that you include an input parameter N_window to control the total length of the impulse response that your function(s) builds.

1. **Delta Function:** (k, N_window) Implements a unit impulse response $\delta[n-k]$ on a domain [0, N_window−1].
2. **Unit Step Function:** (N_window) Implements a unit step response $u[n]$ on a domain [0, N_window−1].
3. **Box Function:** (N_box, N_window) Implements a box response having unit area on a domain [0, N_window−1].
4. **Ema Function:** (Neff, N_window) Implements an ema response $\text{ema}(N_{\text{eff}})$ on [0, N_window−1].

   An ema impulse response is defined as

   $$h_{\text{ema}}[n] \equiv (1-p)p^n u[n] \tag{1}$$

   where the decay rate p is given by $p = N_{\text{eff}}/(1+N_{\text{eff}})$. While in principal the gain is unity, a finite length h_{ema} has a gain of $1 - p^{N_{\text{win}}}$. [1]
5. **Macd Function:** (Neff_pos, Neff_neg, N_window) Implements an macd response $\text{macd}(N_{\text{eff}\,+}, N_{\text{eff}\,-})$ on [0, N_window−1]. The macd response for non-degenerate N_{eff} is defined by

   $$h_{\text{macd}}[n] \equiv h_{\text{ema}(N_{\text{eff}\,+})}[n] - h_{\text{ema}(N_{\text{eff}\,-})}[n] \tag{2}$$

[1] In my work I don't make a gain adjustment, I just ensure that the window is large enough.

6. **Macd M1 Function:** (Neff_pos, Neff_neg, N_window) I don't have a good name for this function, really, but it is very useful in the following for comparison with the macd(). This function is defined by

$$h_{\text{macd}}^{(M1)}[n] \equiv \delta[n - N_{\text{eff}\ +}] - \delta[n - N_{\text{eff}\ -}] \tag{3}$$

<u>DATA:</u> Next you will need some data. Make a mid-quote price series and a trade series using the q-script tools developed in problem-set #1. Use 1-minute bars, regular trading hours, SYMBOL=JPM and EX(CHANGE)=T. The quote and trade files should at least include the columns mid, cnt and px, volume, respectively. Of course I encourage you to explore all the data and parameter variations.

<u>PROBLEMS:</u> In several of these problems I ask you to execute an open convolution the wrong way (!)[2]. The intention is to highlight features that in the future you will have to consider yourself. In all cases submit your code in Matlab or R, and submit plot images or code that will plot each of the steps below. Also, below when I write `conv` I refer to the Matlab version of **open** convolution. If you are writing in R make sure you use the `open` argument.

1. **Impulse Responses:** Make plots of your impulses responses. Use $N_{\text{box}} = N_{\text{eff}} = N_{\text{eff}\ +} = 20$, $N_{\text{eff}\ -} = 40$ and $N_{\text{window}} = 400$. Scale the x-axis the same for all plots (in matlab I use a subplot).

2. **Data:** Plot your quote and trade data.

3. **Truncation:**

   (a) **Incorrect truncation:** With $N_{\text{eff}} = 20$, execute

   $$\text{cand} = \text{conv}\left(h_{\text{ema}}, \text{px}[n]\right)$$

   Overlay the original prices series with your result. What is the length of the price series and cand?

   (b) **Correct truncation:** Repeat the above but throw out the elements in cand that extend beyond the original series. Specifically:

   $$y[n] = \text{cand}(1 : \text{length}(\text{px}))$$

   Explain what this achieves.

   Going forward I always expect you to truncate your convolution results, so I won't write this step out explicitly in the following. In my own work I always return a convolution to cand, as in candidate.

[2] Doing the incorrect work correctly is the right answer.

4. **Initialization:**

   (a) **Incorrect initialization:** With $N_{\text{eff}} = 20$, execute

   $$y[n] = \text{conv}\left(h_{\text{ema}}, \text{px}[n]\right)$$

   Overlay the original price series and explain your result.

   (b) **Correct initialization:** Now execute

   $$y[n] = \text{conv}\left(h_{\text{ema}}, \text{px}[n] - \text{px}[0]\right) + \text{px}[0]$$

   Again, overlay the original price series. Explain the change from the previous version and describe the result now.

   (c) **(In)Correct initialization #2:** With $N_{\text{eff}\,+} = 20$ and $N_{\text{eff}\,-} = 40$, execute the following

   $$\begin{aligned} y_1[n] &= \text{conv}\left(h_{\text{macd}}, \text{px}[n]\right) \\ y_2[n] &= \text{conv}\left(h_{\text{macd}}, \text{px}[n] - \text{px}[0]\right) + \text{px}[0] \\ y_3[n] &= \text{conv}\left(h_{\text{macd}}, \text{px}[n] - \text{px}[0]\right) \end{aligned}$$

   The third expression is the correct one. Provide an explanation comparing the results.

5. **Gain:** Denote gain by g.

   (a) **Ema gain:** For the gains $g \in (0.2, 1.0, 5)$ execute the following:

   $$y[n] = \text{conv}\left(g\ h_{\text{ema}}, \text{px}[n] - \text{px}[0]\right) + g\ \text{px}[0]$$

   In each case overlay the original price series and explain your result. Which gain is correct? Why have I multiplied the offset px[0] by the gain in the last term?

   (b) **Macd gain:** Repeat the above using an macd:

   $$y[n] = \text{conv}\left(g\ h_{\text{macd}}, \text{px}[n] - \text{px}[0]\right) + g\ \text{px}[0]$$

   What is wrong about all of these results? What is the gain of the macd impulse response itself? Using the linearity of the convolution function and the definition of $h_{\text{macd}}[n]$ in terms of $h_{\text{ema}}[n]$ functions, write an expression that shows that $y_3[n]$ in ($4c$) above is the correct way to initialize and compute the convolution with an macd.

6. **Ema of a Price Series:** The purpose of an ema is to regularize an input series. This means that detail is blurred, since each input point is averaged with its one-sided neighbors[3]. The wavelet literature refers to a

[3]The detail is not lost when there exists a functional inverse of the impulse response. In fact $h_{\text{ema}}[n]$ is invertible.

regularized series as a "smooth" and the original as a "rough". The degree of smoothness is governed by the length of the ema window, which itself is parameterized by its first moment N_{eff}.

Calculate and plot (in overlay) the ema impulse responses using the first-moment sequence $N_{\text{eff}} \in (2, 5, 10, 20, 50, 100)$. Then, convolve the price series with each ema,

$$y[n] = \text{conv}(h_{\text{ema}(N_{\text{eff}})}[n], \text{px}[n] - \text{px}[0]) + \text{px}[0]$$

and plot these results and overlay the original series. Explain your observations.

7. **Ema and Delay:** Regularization has a price. That price is delay. The higher the level of regularization the more the output is delayed. We can artificially impart equivalent delay to distinguish the delay and local-average effects.

   Repeat the work from 6 but substitute an ideal delay $h[n] = \delta[n - N_{\text{eff}}]$ for $h_{\text{ema}(N_{\text{eff}})}$. Then, overlay the ideal-delay results with the associated ema results for each N_{eff}. Explain your observations.

8. **Macd-M1 of a Log-Price Series:** A spot compounding return is written as

   $$r_{t,t+h} = \log(\text{px}_{t+h}) - \log(\text{px}_t)$$

   Rewriting for discrete time, where I substitute n for t and η for h we have

   $$r_{n,n+\eta} = \log(\text{px}_{n+\eta}) - \log(\text{px}_n)$$

   The interval η is just the gap between a pair of positive and negative impulse functions.

   For this problem apply the macd-M1 function to a log-price series according to

   $$y[n] = \text{conv}(h_{\text{macd}-M1}(N_{\text{eff}\,+}, N_{\text{eff}\,-})[n], \log(\text{px}[n]) - \log(\text{px}[0]))$$

   where $N_{\text{eff}\,+} \in (2, 5, 10, 20, 50)$ and $N_{\text{eff}\,-} = 2N_{\text{eff}\,+}$. Clearly $N_{\text{gap}} = N_{\text{eff}\,+} - N_{\text{eff}\,-}$.

9. **Macd of a Log-Price Series:** Repeat the preceding problem where $h_{\text{macd}-M1}$ is replaced with h_{macd}. For each $N_{\text{eff}\,+}$ overlay this result with the corresponding one from the preceding problem. Explain your results.

10. **Smoothing a Trade-Bin Series:** For this problem use the binned trade volume as the input.

    (a) **Ema:** Apply an ema to this series with $N_{\text{eff}} = 21$.

(b) **Compound Smoothing:** Apply to unit step function to the binned trade series:

$$x[n] = \text{conv}(u[n], \text{tr} - \text{tr}[0]) + \text{len}(\text{tr}) \times \text{tr}[0]$$

Follow this by applying a macd to the result, as in:

$$y[n] = \text{conv}(\frac{h_{\text{macd}}[n]}{\Delta N}, x[n] - x[0]) + \text{tr}[0]$$

with $N_{\text{eff}\,+} = 7$ and $N_{\text{eff}\,-} = 14$ and $\Delta N \equiv N_{\text{eff}\,-} - N_{\text{eff}\,+}$. Overlay this result with the one from the ema. Comment on your observations.

Explain clearly the gain of each impulse response, and why offsets that I added back after each convolution are correct.

(c) **Compound Smoothing Impulse Response:** Derive a new impulse response h_{cmp} by convolving the unit step with the macd:

$$h_{\text{cmp}} \equiv \text{conv}(u[n], h_{\text{macd}}[n]/\Delta N) \tag{4}$$

where the length of $u[n]$ and h_{cmp} is set to length of the trade-volume series. Make a plot of the unit step, macd and compound impulse responses, all on the same scale. What is the gain of h_{cmp}?

Using the compound impulse response, recompute the smoothed series again:

$$y[n] = \text{conv}(h_{cmp}[n], \text{tr}[n] - \text{tr}[0]) + \text{tr}[0]$$

Compare this result to the one above.

Explain why ΔN needs to be considered when constructing (4).

11. **Circular Convolution:** This problem makes a simple comparison of open and circular convolution. For this problem, download the cconv function from the Matlab Central page http://www.mathworks.com/matlabcentral/fileexchange/13030−circular−convolution. Place this file in a path that Matlab can read.

Circular convolution is itself interesting and direct computation (rather than the Fourier-based approach) has applications in signal processing.

However, circular convolution is **non-causal**. Artifacts of future events can appear in the past. This is disastrous for financial calibrations. As current or future professionals you need to be acutely aware of how the `conv` function that you use is implemented to ensure that it is open, and therefore causal.

(a) **Impulse Responses:** Make an ema impulse response h_{ema} with $N_{\text{eff}} = 32$ and $N_{\text{window}} = 256$. Then, make a delta-function response $h_\delta = \delta[n - k]$ with $k = 192$ and N_{window} the same. Plot these two impulse responses.

(b) **Causal Convolution:** Using open convolution (`conv` in Matlab) compute and plot the convolution of the ema and delta function:

$$y[n] = \mathrm{conv}(h_{\mathrm{ema}}, h_\delta)$$

(c) **Non-causal Convolution:** Using the circular convolution function cconv, compute and plot the response

$$y_{\mathrm{circ}}[n] = \mathrm{cconv}(h_{\mathrm{ema}}, h_\delta)$$

Overlay this plot with that of open convolution.

(d) **Comparison:** Identify in your plot the non-causality region of circular convolution (in this example). Explain the origin of this artifact.

If the impulse were a significant market event and you calibrated a regularizing model using circular convolution, what error would you mak?e