VSRT Memo # 056
10 March 2009

Diurnal ozone variations, smoothing of MOSAIC data, and signal-to-noise ratios

Vincent L. Fish

Abstract

This memo provides a simple script for showing the diurnal variation of mesospheric ozone using GNU Octave. Graphical examples of data smoothing are provided in order to demonstrate the tradeoffs involved with averaging data. A brief summary of the signal-to-noise ratio is included, along with its implications for data collection.

1 Introduction

Mesospheric ozone concentration is expected to be much lower during daylight hours than at night due to rapid photodissociation by the Sun (see VSRT Memo #040 for more details). It is desirable to combine data from multiple units and over many days of observing due to the modest sensitivity of a single MOSAIC unit. Since sunrise and sunset times vary seasonally, data from multiple days should be combined using "local equinox time," defined in VSRT Memo #048.

A simple analysis script to produce spectra of the MOSAIC data using GNU Octave has been presented in VSRT Memo #054. This memo has two goals: to provide a sample GNU Octave script to bin data by local equinox time and produce a plot of the diurnal variation, and to demonstrate the effects of smoothing the data. It is intended that this script will not be used as is for the purpose of data reduction, but rather as a starting point for pedagogical exploration and/or more sophisticated analysis.

2 Notes on the plots produced by the script

The GNU Octave script appears in Section 5. It will produce plots similar to those in this section, although some of the code is omitted for brevity.

The spectrum of all MOSAIC data averaged together is shown in Figure 1. The peak of the emission is in channel 32, although channel 33 is nearly as strong. Rather than simply using channel 32, the script averages data from channels 32 and 33 for increased sensitivity.

It is often convenient to think in terms of the signal-to-noise ratio, SNR $= S/\sigma$, where S is the signal strength and σ is a measure of the noise. It can also be estimated in a spectrum by the spread of the channel-to-channel fluctuations about the mean (see Section 3.1). Combining data from channels 32 and 33 increases the SNR of the combined data point by nearly $\sqrt{2}$ (see Section 3.2).

Figure 2 shows the daily variation of ozone and the effects of applying smoothing. The black curve shows the data at full resolution (0.1 hours). The green and red curves show the

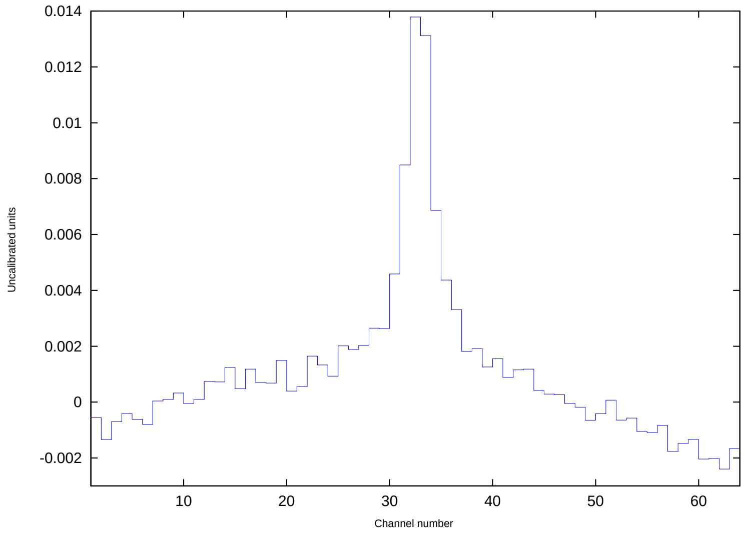

Figure 1: Spectrum of MOSAIC data.

effects of applying 3- and 5-channel boxcar averaging, respectively (see Section 3.3). The boxcar averaging smooths out the high-frequency variations in the data, which are presumed to be just noise. However, it does so at the expense of resolution. For instance, if there really were 12-minute variations in the spectrum, you would have difficulty seeing them in the smoothed spectra. Another example is shown as the cyan (bluish) line in Figure 2, which shows the results of applying 15-channel smoothing to the data. The cyan line fails to capture the steepness of the falloff in the spectrum near sunrise. It is important not to smooth data beyond the resolution that is required to identify the features you are looking for.

Smoothing data causes loss of information. The panels in Figure 3 show data before and after 3-channel boxcar averaging has been applied. The panels also show the spectra before and after decimation by a factor of two (i.e., dropping alternate channels). Decimation of the unsmoothed data loses half the information. Smoothing data also loses information, but note that the channel-to-channel variations are lower. Never decimate data without smoothing it first, or you're just throwing away half of your signal! It is also apparent in these plots that smoothing data causes the data in each channel to be correlated with those on either side. The smoothed, decimated spectrum looks much more like the smoothed, undecimated spectrum than the unsmoothed, decimated spectrum looks like the unsmoothed, undecimated spectrum. In other words, there is less information lost by decimation after smoothing because the smoothing has already caused information loss.

An alternate graphical demonstration of boxcar smoothing is provided in Figure 4. The smoothed photographs contain less information than the original.

Other smoothing functions besides boxcar smoothing are possible. For instance, Hanning smoothing (Section 3.4) is often used in radio astronomy. Figure 5 compares the results of 3-channel boxcar and Hanning smoothing. Hanning smoothing retains slightly more information than does boxcar smoothing, but the differences are minor.

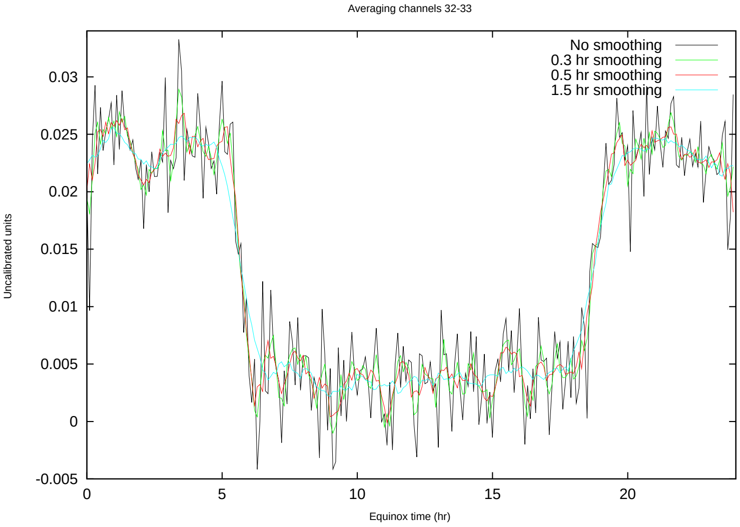

Figure 2: Diurnal variation of ozone (black). The same curve is shown with 3-channel (green), 5-channel (red), and 15-channel (cyan) boxcar smoothing. The cyan curve does not adequately sample the sharp falloff of ozone near 6^h.

3 On signals, noise, their ratio, and smoothing functions

3.1 How can you read the noise off a spectrum without errorbars?

When errors follow a normal distribution with standard deviation σ, approximately 68% of the points fall within $\pm 1\sigma$, 95% within $\pm 2\sigma$, 99% within $\pm 2.5\sigma$, and 99.7% within $\pm 3\sigma$. (For a perfect normal distribution, the fraction of points that fall within $\pm m\sigma$ is given by $\mathrm{erf}(m/\sqrt{2})$.) Looking at this another way, you would expect to need to see 100 data points to find one that has a random error of 2.5σ and about 370 data points to find one that has a random error of 3σ. This provides a handy rule of thumb: the range of the maximum outlier points of a spectrum above and below the expected value measures between 4σ and 5σ. For example, mesospheric ozone concentration is expected to be small and fairly constant throughout the daylight hours. In Figure 6, the black curve has 101 points from 7^h to 17^h local equinox time, inclusive. The difference between the maximum and minimum values turns out to be 4.3σ.

3.2 Why does combining 2 data points of equal SNR raise the SNR by $\sqrt{2}$?

This can be thought of as adding two data points $S_1 + \epsilon_1$ and $S_2 + \epsilon_2$, where the ϵ's represent instantiations of noise drawn from a Gaussian random distribution centered at zero and with standard deviation σ. Since the signal is assumed not to be changing, $S_1 + S_2 = 2S$. If the two data points are independent measurements, their noises will not be correlated and so will statistically add in quadrature (i.e., Pythagorean addition). The noise on the combined data point is $\sqrt{\epsilon_1^2 + \epsilon_2^2}$, whose expectation value is $\sqrt{2}\sigma$. Thus the combined SNR is $2S/\sqrt{2}\sigma$, or $\sqrt{2}$ times the SNR of a single data point.

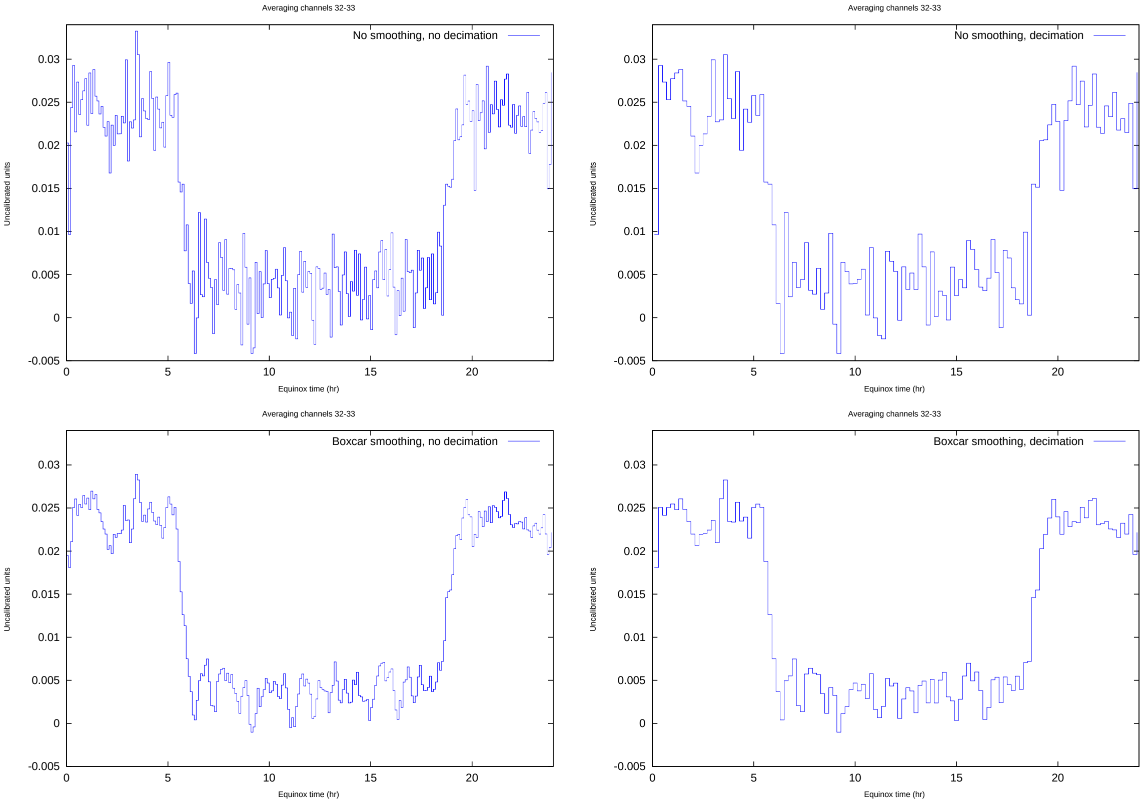

Figure 3: Spectrum before (*top*) and after (*bottom*) application of 3-channel boxcar smoothing. Spectra are shown before (*left*) and after (*right*) decimation of every alternate channel. The smoothed spectrum contains less information than the unsmoothed spectrum. Half of the information is lost when the unsmoothed spectrum is decimated.

Likewise, adding together n independent data points of equal SNR raises the SNR of the sum by $\sqrt{n}$. This has fundamental implications for the amount of data needed to achieve a desired sensitivity. To reduce the error bars on a data point by a factor of 4 requires increasing the observing time by a factor of 16. Alternatively the number of observing systems can be increased; 16 MOSAIC units observing for 1 unit of time will produce the same SNR as 1 MOSAIC unit observing for 16 times as long. In many scientific experiments, this $\sqrt{n}$ growth of the SNR determines the sensitivity that can be obtained, as it may be unrealistic to build 4 times as many instruments or observe 4 times as long to get the next factor of 2 in SNR.

3.3 What is boxcar smoothing?

Boxcar smoothing is a fancy term for doing a rolling average of the data. In other words, 3-channel boxcar smoothing replaces every data point d_i with $(d_{i-1} + d_i + d_{i+1})/3$. This is mathematically equivalent to discrete convolution with a kernel whose value is 1 between -1.5 and 1.5 and zero elsewhere. If you plot that up (Figure 7), it looks like a boxcar.

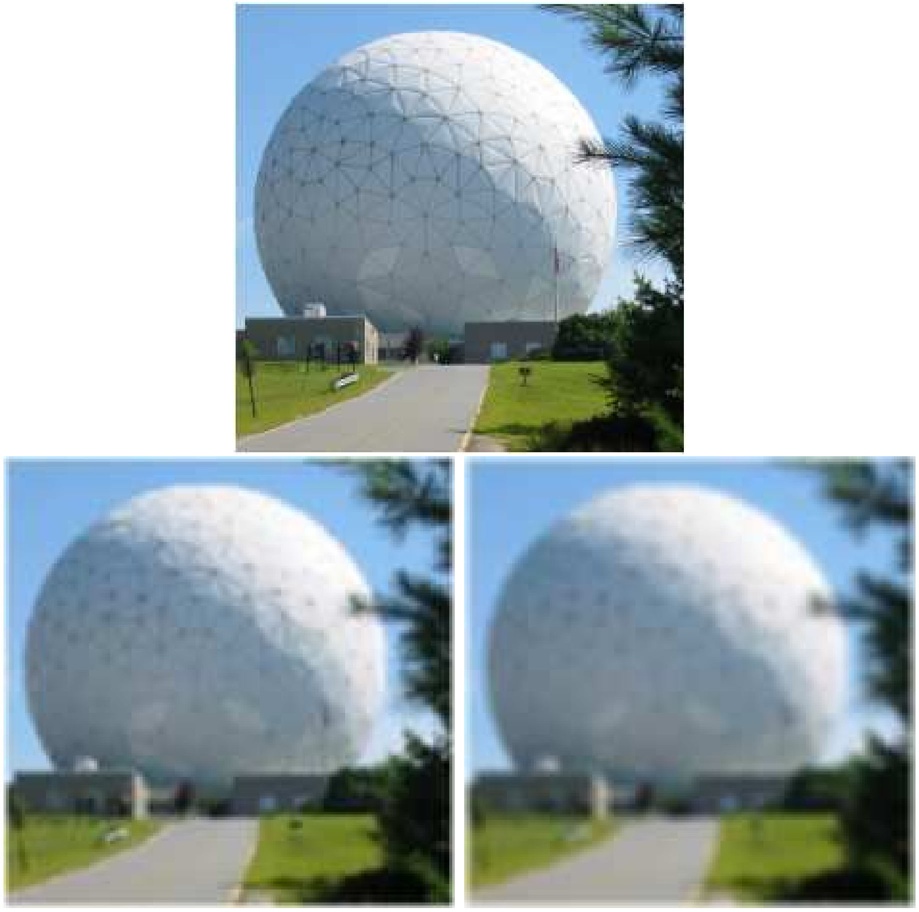

Figure 4: *Top*: An image of MIT Haystack Observatory. *Bottom left*: The image after 3-pixel boxcar smoothing in each direction. *Bottom right*: The same but with 5-pixel boxcar smoothing.

3.4 What is Hanning smoothing?

Hanning smoothing uses a narrower convolution kernel than boxcar smoothing. Compare 3-channel Hanning smoothing, which replaces d_i with

$$\frac{1}{4}d_{i-1} + \frac{1}{2}d_i + \frac{1}{4}d_{i+1},$$

with 3-channel Boxcar smoothing, which replaces d_i with

$$\frac{1}{3}d_{i-1} + \frac{1}{3}d_i + \frac{1}{3}d_{i+1}.$$

In effect, each channel after Hanning smoothing is more sensitive to values close to it than to others far away. In boxcar smoothing, each channel after smoothing is equally sensitive to all channels with which it is being averaged.

Arbitrary convolution kernels can be chosen for smoothing. The coefficients in front of the d terms should sum to 1, else the spectrum will be rescaled. In general, the coefficients in the

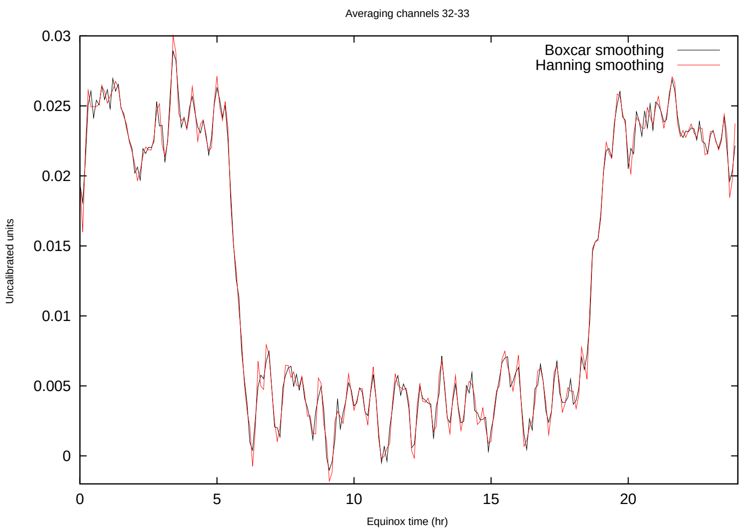

Figure 5: Comparison of boxcar (black) and Hanning (red) smoothing.

smoothing function should not increase with increasing distance from the center channel (d_i). The identity smoothing operation (i.e., the convolution kernel which provides no change to the spectrum at all) is $1d_i$, or expressed in the format as the above two equations,

$$0d_{i-1} + 1d_i + 0d_{i+1}.$$

The Hanning convolution kernel is “closer” to this function than the boxcar convolution kernel (see Figure 7), which explains why the black curve in Figure 5 is a bit smoother than the red curve.

For a very technical explanation of why Hanning smoothing is often used in radio astronomy, see `http://www.vla.nrao.edu/astro/guides/sline/current/node11.html`, which describes the motivation for and effects of Hanning smoothing on data from the Very Large Array. Warning: some knowledge of Fourier transforms is required!

4 Other tips and tricks

It is good practice to display spectra by squaring off each channel (as in Figure 3) rather than by simply connecting the data at the center of each channel (as in Figure 2). The channel width of spectra are more easily read in Figure 3, produced by the `stairs` command, than by connecting the center of each channel with the `plot` command. Unfortunately, the GNU Octave `stairs` command does not directly support the specification of parameters such as the color of the line used for the plot. (This is one way in which GNU Octave is not fully compatible with MATLAB.)

Loading MOSAIC data into GNU Octave can take up a lot of memory. To see the memory usage of local user variables, type `whos` at the Octave prompt. Performance may be improved by deleting large variables after they are no longer necessary. For instance, the variable `data`

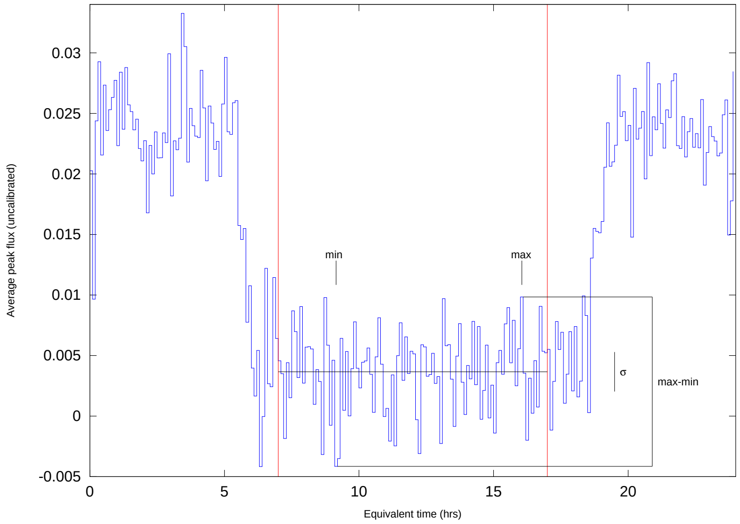

Figure 6: Example of reading the noise from a plot. The red lines show the range between 7^h and 17^h local equinox time, where the ozone concentration is expected to be approximately constant with time. The horizontal black line shows the average value in this time range. The two short vertical black lines indicate the minimum (left) and maximum (right) outlier channels. The difference between the maximum and minimum values is 4.3 times the noise.

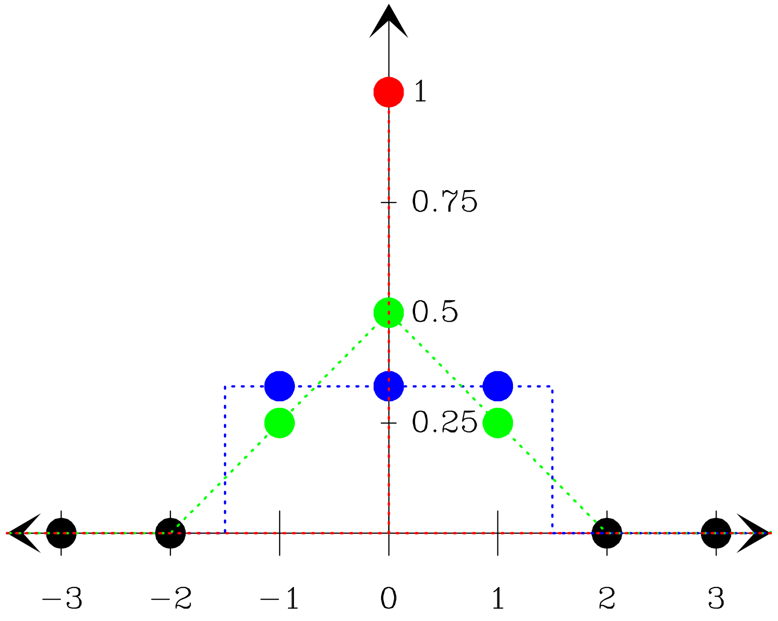

Figure 7: Convolution kernels for smoothing operations. *Red*: No smoothing. *Green*: Hanning smoothing (3 channels). *Blue*: Boxcar smoothing (3 channels). Points in black are in common to all three convolution kernels.

in the script in Section 5 is no longer used after the variables `num_rec`, `ltm`, and `spectrum` have been defined. The command `clear data` will free up the memory associated with the variable `data`.

The command `hold` will cause subsequent plots to be overlaid atop the existing plot instead of clearing the plot first. The command `hold off` will turn off this feature.

Boxcar smoothing of images can be easily accomplished using the popular ImageMagick tools. To apply n-channel boxcar smoothing (n odd) matrix, use the command `convert input-file -convolve 1,1,1,...,1 output-file` at the Unix prompt, where n^2 ones appear in the sequence.

GNU Octave can compute some statistics easily. For instance, finding the average value and noise in the daylight portion of the plot in Section 3.1 (and checking the rule of thumb mentioned therein) can be accomplished in a few lines:

```
minval = min(ltm_spectrum(71:171))
maxval = max(ltm_spectrum(71:171))
average = mean(ltm_spectrum(71:171))
rms = std(ltm_spectrum(71:171))
(maxval-minval)/rms
```

There may be instructional value in coding the loops for computing the average and the noise without resorting to these functions as well.

5 Script

If this script is saved as a file called `diurnal.m`, it can be invoked at the GNU Octave prompt by typing `diurnal`. Lines beginning with # are comments.

```
# DIURNAL -- Does diurnal averaging by ltm
# -------

# Load data
# See VSRT Memo #054 for details of this block
fid=fopen('chsout.txt','r');
# The following prints as four separate lines but should appear on one
[data,count] = fscanf(fid,"%*s %f %*s %f %*s %f %*s %f %*s %f %*s %f %*s %f
%*s %f %*s %f %*s %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f
%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f
%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f",[73,Inf]);
fclose(fid);
n = count/73;
num_rec = data(5,:);
ltm = data(9,:);
spectrum = data([10:73],:);
unitcolumn = ones(n,1);
total_records = num_rec*unitcolumn;
weighted_spectrum = spectrum*num_rec'/total_records;
```

```
# Define an array of local equinox times 0.0, 0.1, 0.2, ..., 23.8, 23.9
ltmarray = linspace(0,23.9,240);

# Compute the spectrum as a function of equinox time
ltm_num_rec = zeros(1,240);
ltm_spectrum = zeros(1,240);
for i = 1:n
  # Convert the equinox time (ltm) to an index in the 240-element arrays
  # Remember that array indices in Octave start at 1, not zero
  index = 10*ltm(1,i)+1;
  if (index == 241)
    index = 1;
  endif
  ltm_num_rec(index) += num_rec(i);
# Use just channel 32...
#  ltm_spectrum(index) += num_rec(i)*spectrum(32,i);
# ...or use averages of channels 32 and 33 instead
  ltm_spectrum(index) += num_rec(i)*(spectrum(32,i)+spectrum(33,i))/2;
endfor
for j = 1:240
  ltm_spectrum(j) /= ltm_num_rec(j);
endfor

# Compute spectrum after 3-channel averaging
ltm_boxcar3_spectrum = zeros(1,240);
for j = 2:239
  # This and subsequent similar lines are wrapped for clarity
  ltm_boxcar3_spectrum(j) = (ltm_spectrum(j-1)+ltm_spectrum(j)
    +ltm_spectrum(j+1))/3;
endfor
# Define the edge channels manually to deal with wrap at 24.0 hours ltm
ltm_boxcar3_spectrum(240) = (ltm_spectrum(239)+ltm_spectrum(240)
    +ltm_spectrum(1))/3;
ltm_boxcar3_spectrum(1) = (ltm_spectrum(240)+ltm_spectrum(1)
    +ltm_spectrum(2))/3;

# Compute spectrum after 5-channel averaging
ltm_boxcar5_spectrum = zeros(1,240);
for j = 3:238
  # This and subsequent similar lines are wrapped for clarity
  ltm_boxcar5_spectrum(j) = (ltm_spectrum(j-2)+ltm_spectrum(j-1)
    +ltm_spectrum(j)+ltm_spectrum(j+1)+ltm_spectrum(j+2))/5;
endfor
ltm_boxcar5_spectrum(239) = (ltm_spectrum(237)+ltm_spectrum(238)
    +ltm_spectrum(239)+ltm_spectrum(240)+ltm_spectrum(1))/5;
ltm_boxcar5_spectrum(240) = (ltm_spectrum(238)+ltm_spectrum(239)
    +ltm_spectrum(240)+ltm_spectrum(1)+ltm_spectrum(2))/5;
ltm_boxcar5_spectrum(1) = (ltm_spectrum(239)+ltm_spectrum(240)
    +ltm_spectrum(1)+ltm_spectrum(2)+ltm_spectrum(3))/5;
ltm_boxcar5_spectrum(2) = (ltm_spectrum(240)+ltm_spectrum(1)
    +ltm_spectrum(2)+ltm_spectrum(3)+ltm_spectrum(4))/5;

# Do decimation
ltmarray_decimated = zeros(1,120);
ltm_boxcar3_spectrum_decimated = zeros(1,120);
for j = 1:120
  ltmarray_decimated(j) = ltmarray(2*j);
  ltm_boxcar3_spectrum_decimated(j) = ltm_boxcar3_spectrum(2*j);
endfor

# Demonstrate Hanning smoothing
ltm_hanning_spectrum = zeros(1,240);
for j = 2:239
  # This and subsequent similar lines are wrapped for clarity
  ltm_hanning_spectrum(j) = 0.25*ltm_spectrum(j-1)+0.5*ltm_spectrum(j)
    +0.25*ltm_spectrum(j+1);
endfor
ltm_hanning_spectrum(240) = 0.25*ltm_spectrum(239)+0.5*ltm_spectrum(240)
    +0.25*ltm_spectrum(1);
ltm_hanning_spectrum(1) = 0.25*ltm_spectrum(240)+0.5*ltm_spectrum(1)
    +0.25*ltm_spectrum(2);


# Create plots

# Show spectrum
# Use stairs rather than plot to show channels clearly
stairs(weighted_spectrum);
axis([1 64]);
xlabel(’Channel number’);
ylabel(’Uncalibrated units’);
legend(’off’);
print -depsc spectrum.eps

# Show effects of smoothing
plot(ltmarray,ltm_spectrum,’k’);
hold;
plot(ltmarray,ltm_boxcar3_spectrum,’g’);
plot(ltmarray,ltm_boxcar5_spectrum,’r’);
```

```
xlabel(’Equinox time (hr)’);
ylabel(’Uncalibrated units’);
title(’Averaging channels 32-33’);
legend(’No smoothing’,’0.3 hr smoothing’,’0.5 hr smoothing’);
print -depsc smoothing_example.eps
# Turn off the overlay feature
hold off;

# Show effects of decimation
stairs(ltmarray,ltm_boxcar3_spectrum);
legend(’No decimation’);
xlabel(’Equinox time (hr)’);
ylabel(’Uncalibrated units’);
title(’Averaging channels 32-33’);
print -depsc no_decimation_example.eps
stairs(ltmarray_decimated,ltm_boxcar3_spectrum_decimated);
legend(’Decimation’);
xlabel(’Equinox time (hr)’);
ylabel(’Uncalibrated units’);
title(’Averaging channels 32-33’);
print -depsc decimation_example.eps

# Show Hanning vs. boxcar smoothing
plot(ltmarray,ltm_boxcar3_spectrum,’k’);
hold;
plot(ltmarray,ltm_hanning_spectrum,’r’);
xlabel(’Equinox time (hr)’);
ylabel(’Uncalibrated units’);
title(’Averaging channels 32-33’);
legend(’Boxcar smoothing’,’Hanning smoothing’);
print -depsc hanning_example.eps
hold off;
```