AUTOMATED PROCESSING SYSTEM FOR TIDAL ANALYSIS OF MEDIUM FREQUENCY RADAR WINDS

By

Sreenivas Vemula

RECOMMENDED: C. S. Sonwalk
_______________
_______________
Denise Thorsen
Advisory Committee Chair

Clark E Mayer
Chair, Department of Electrical and Computer Engineering

APPROVED: Clark E Mayer
Dean, College of Engineering and Mines

Susan M. Henrichs
Dean of the Graduate School

December 13, 2005
Date

AUTOMATED PROCESSING SYSTEM FOR TIDAL ANALYSIS OF MF RADAR WINDS

A

THESIS

Presented to the Faculty

of the University of Alaska Fairbanks

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

By

Sreenivas Vemula, B.E.

Fairbanks, Alaska

December 2005

Abstract

The medium frequency (MF) radar at Platteville, Colorado (40.18° N, 104.7° W) is used to estimate the zonal and meridional wind motions in the middle atmosphere. This radar has been in operation since January 2000. We currently have four years of wind estimates sampled every five minutes. An automated processing system has been developed in IDL to process these estimates and obtain the monthly mean winds and tidal parameters. The automated processing currently processes the wind estimates in time domain analysis using a least square fitting technique. The criteria for determining when the estimated tidal parameters are valid have been studied along with the error analysis of the data and processing. The diurnal and semidiurnal parameters are obtained using this least square fitting method and these tidal parameters are assumed to be valid only when the condition number is less than 10. In the spectral domain, the fast Fourier transform and Lomb-Scargle periodogram methods have been studied. A test signal is generated and its performance using both FFT and Lomb-Scargle methods are discussed for three different cases which are equivalent to our actual data. The results of the wind estimates from 2000-2003 collected using the MF radar have been processed using the automated processing system. This automated processing system can be used to generate the wind parameters on a 24 hour, 7 day a week basis for an elaborate study. Our data are compared with MF radar data from Saskatoon, Canada and Urbana, Illinois. Most of the time our data are similar to the behavior of GSWM-02 model.

Table of Contents

List of Figures

List of Tables

Page

# Acknowledgements

I would like to thank my advisor Dr. Denise Thorsen for giving me an opportunity to work on this project. I express my sincere thanks for her time and patience along with the guidance. It has been a very good experience working under her and I learned a lot from her.

My graduate committee members Dr. Vikas Sonwalkar and Dr. Richard Collins have been helpful in providing valuable suggestions when needed and I am thankful to them. I also acknowledge the financial support of Geophysical Institute.

Finally, I am grateful to my family and friends who have always been my support.

# 1 Introduction

The medium frequency (MF) radar at Platteville, CO (40.18° N, 104.7° W) is used to study middle atmosphere dynamics. Wind motions for heights from 60 km to 100 km are estimated from partial reflections observed by the radar. These raw wind estimates are used to derive mean horizontal winds, planetary waves, tides and gravity waves. The goal of this thesis is to provide the framework for the automatic processing of the raw wind estimates into monthly mean winds and monthly mean tidal amplitudes and phases.

## 1.1 Middle Atmosphere Wind Motions

Atmospheric tides are the dominant wave motions in the middle atmosphere. Even though seasonal variations are usually quite strong at all latitudes, tides are still the most regular and persistent oscillations observed. Theoretical discussion of these tidal motions and their governing equations can be found in *Chapman and Lindzen* [1970] and *Andrews et al.* [1987]. For a comprehensive tutorial on tidal and planetary waves the reader is directed to *Forbes* [1987]. An overview of the key results from these references is given here.

Typically the Navier-Stokes equations for a compressible gas are used to describe the motion of the atmosphere. It is generally assumed: *(i)* that the atmosphere is in local thermodynamic equilibrium, *(ii)* that the atmosphere is a geometrically thin fluid layer on a rotating earth and *(iii)* that the atmosphere is in hydrostatic equilibrium which implies the vertical accelerations are small compared with the acceleration due to gravity ($g$). The earth's ellipticity and surface topography are generally ignored, thereby ignoring the influence of mountains and land-sea distribution. Tidal fields are considered as linear perturbations about some basic state.

Tides are global-scale oscillations in temperature, wind, density, and pressure at periods that are sub-harmonics of a solar day. These tides can be *forced* due to periodic excitation of various sources, or be *free* and determined by the resonant response of the atmosphere. In addition the forced waves can be further classified as *migrating* or *non-migrating*. This classification separates tides that are forced due to sun-synchronous heating or heating by localized excitation sources respectively (e.g. UV radiation, infrared, water vapor, etc.). Atmospheric tides may propagate eastward or westward but the largest component of these tides is the thermally forced *migrating* (sun-synchronous) wave that propagates in the westward direction. This thesis focuses on thermally forced migrating atmospheric tides where the dominant heat source is the absorption of solar radiation.

Figure 1-1 shows an example representation of possible solar heating (SH) as a function of time. The heating by the sun is diurnal in nature. There is no heating prior to sunrise (SR) or after sunset (SS). The heating reaches its peak at solar local noon (SLN). This kind of heating causes tides to occur which are basically diurnal in nature.

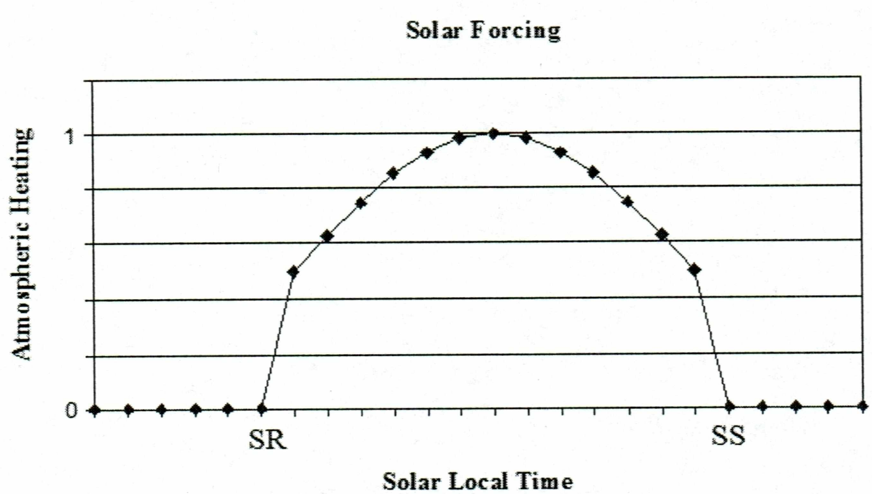



Figure 1-1 Schematic of strength of typical solar forcing.

To show this we can assume an arbitrary functional form for the solar heating and examine its Fourier series coefficients. In this example calculation, we have assumed that the solar heating can be represented by a raised cosine with maximum heating at SLN,

$$SH(t) = 0.5 + 0.5\cos\left(\pi\frac{(t-12)}{\tau}\right) \tag{1.1}$$

where τ is the time between sunrise and sunset and is assumed to be centered at SLN. It is instructive to examine the Fourier series representation of *SH(t)*, which is given by:

$$SH(t) = a_o + \sum_{n=1}^{\infty}\left(a_n\cos\left(\frac{2\pi nt}{T}\right) + b_n\sin\left(\frac{2\pi nt}{T}\right)\right) \tag{1.2}$$

Here T is the period of the forcing and is equal to 24 hours. The coefficients $a_0$ , $a_n$ and $b_n$ are given by:

$$a_o = \frac{1}{T}\int_{SR}^{SS} SH(t)dt \tag{1.3}$$

$$a_n = \frac{2}{T}\int_{SR}^{SS} SH(t)\cos\left(\frac{2\pi nt}{T}\right)dt \tag{1.4}$$

$$b_n = \frac{2}{T}\int_{SR}^{SS} SH(t)\sin\left(\frac{2\pi nt}{T}\right)dt \tag{1.5}$$

Because *SH(t)* is an even function, the $b_n$ coefficients are identically equal to zero. The DC coefficient can be shown to be $a_0 = 0.82\tau/T$, where the constant 0.82 results from our choice for the functional of solar heating. If we had chosen $SH(t) = 1$ from sunrise to sunset, then $a_o = \tau/T$. The $a_n$ is calculated for our given function *SH(t)* as follows:

$$\begin{aligned} a_n &= \frac{2}{T}\int_{SR}^{SS} SH(t)\cos\left(\frac{2\pi nt}{T}\right)dt \\ &= \frac{2}{T}\int_{12-\tau/2}^{12+\tau/2}\left[\cos\left(\frac{2\pi nt}{T}\right) + \cos\left(\frac{\pi(t-12)}{\tau}\right)\cos\left(\frac{2\pi nt}{T}\right)\right]dt \\ &= \frac{(-1)^n}{n\pi}\left[\cos\left(\frac{n\pi\tau}{T}\right)\left[\frac{12n\tau}{144-n^2\tau^2}\right] + \sin\left(\frac{n\pi\tau}{T}\right)\right] \end{aligned} \tag{1.6}$$

When $144-n^2\tau^2 = 0$ then $n\tau = 12$ and $a_n$ becomes

$$a_n = \frac{(-1)^n}{\pi n}\left[1+\frac{\pi}{4}\right] \tag{1.7}$$

Figure 1-2 shows the Fourier coefficients for three values of $\tau$ (4, 12, and 22). These three cases are representative of polar winter, equinox, and polar summer solar forcing at Fairbanks, AK. The representative forcing at a mid-latitude site like Platteville, CO would be bounded by the extremes of the polar forcing examples shown. The x- and y-axis represents the period in hours and the amplitude of the Fourier coefficient, respectively. The 24 hr, 12 hr, 8 hr, and 6 hr components (n=1, 2, 3, and 4) for all the cases are shown. For comparison the DC component, $a_0$, is also shown at 1/freq = 0.

As expected the diurnal forcing, $a_1$, of the Fourier decomposition is dominant. In fact, $a_1$ is larger than even the DC component until $\tau > 16$ hours. The diurnal forcing reaches its maximum value for $\tau = 12$ hours. The semi-diurnal forcing, $a_2$, reaches its maximum value for $\tau = 6$ hours. Even at $\tau = 6$ hours the diurnal component of the solar forcing is larger than the semi-diurnal component.

Given the variability of the strength in each Fourier component for different lengths of daylight, one might expect that the atmosphere would have large diurnal tides at the equinox with smaller amplitude diurnal tides at the solstices. In fact this is observed at Platteville (see Chapter 4). One might also expect that the diurnal tide would dominate throughout the year at all locations, with the semi-diurnal tidal amplitude being the strongest in early and late winter. Our data do not support this.

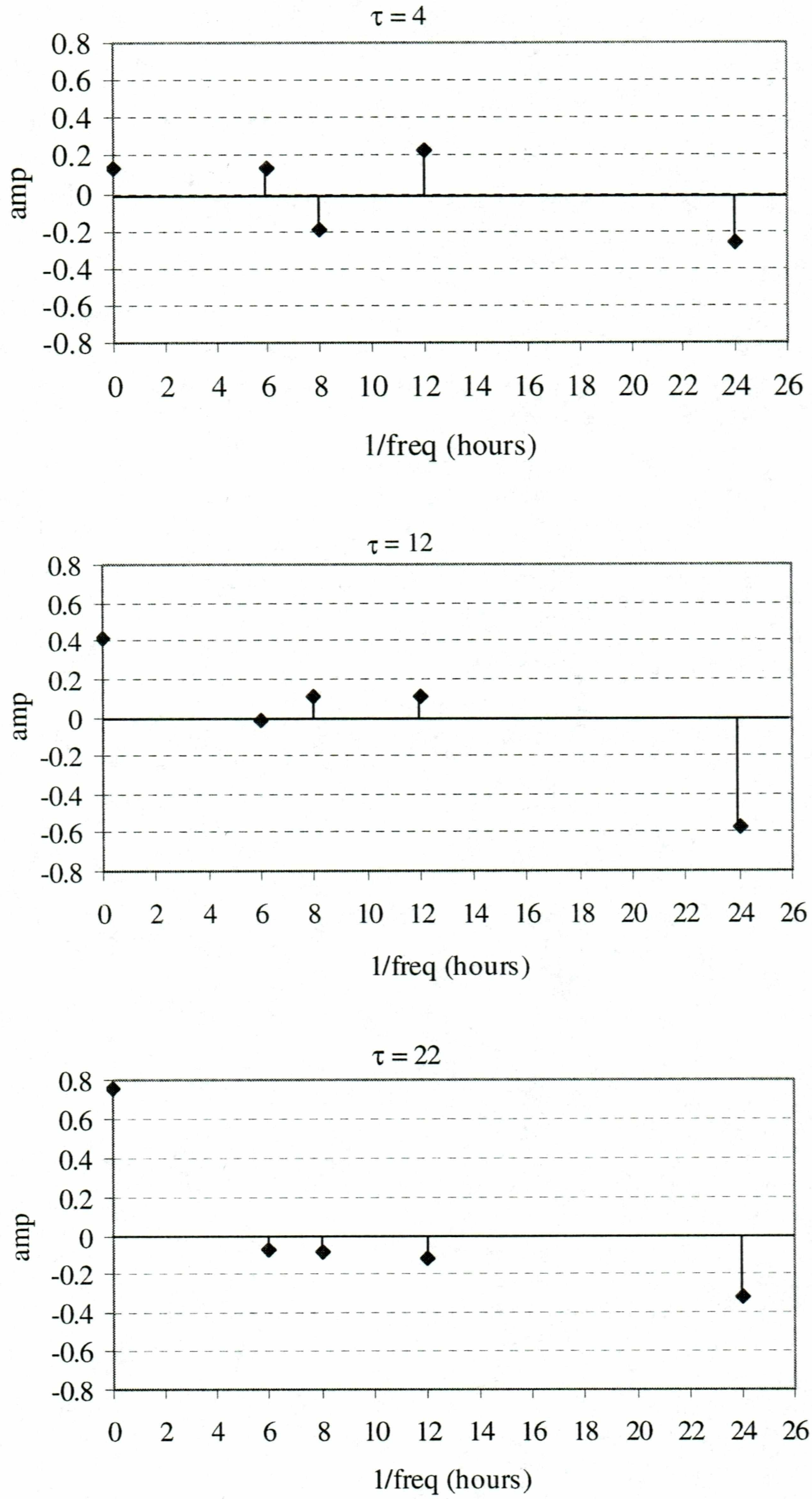



Figure 1-2 Fourier decomposition of solar forcing function for $\tau$ equals 4, 12, 22 hours.

This simplistic analysis does not take into account the efficiency with which each mode is excited in the atmosphere. Specifically, the efficiency of tidal excitation by the absorption of solar radiation is latitudinally dependent. This latitudinal dependence is given by the Eigen functions (Hough functions) [*Forbes,* 1987] which form the solution to Laplace's tidal equations. The Hough functions show that the diurnal tide will be dominant in the equatorial region, whereas the semi-diurnal tide would be dominant in the mid-latitudes and polar regions.

During the past 25 or 30 years there have been remarkable advances made in our understanding of the dynamics, physics and chemistry of the middle atmosphere [*Fritts and Axexander*, 2003, *Forbes*, 1984, *Hagan*, 1999, *Solomon*, 1999]. Although the classical solution for tidal modes, presented above, helps us in understanding the dynamics of the tides, it is not a complete solution. The distribution of ozone, and water-vapor in height, season and location, background winds and mean temperatures, molecular and eddy diffusivity, gravity wave stress and Newtonian cooling effects all have major effects on the behavior of the tides. These tidal structures are very complicated in nature and hence the modeling of these includes better understanding of the atmospheric phenomena. The Global Scale Wave Model (GSWM) attempts to include many of these non-classical effects [*Hagan*, 2003]. In this thesis we will be comparing the tidal analysis of Platteville MF data to the GSWM 2000.

## 1.2 Platteville Medium Frequency (MF) Radar

The Platteville MF radar located at the Platteville Atmospheric Observatory, Platteville, Colorado (40.18° N, 104.7° W) has been in operation since January of 2000. A diagram of Platteville Atmospheric Observatory is shown in Figure 1-3 along with the location of MF radar. The Platteville MF radar is vertically pointing pulsed radar which operates in a spaced received antenna configuration. The green rectangular box (lower right of figure) shows the location of the transmit array and the three circles in green show the location of the three receive antennas. The transmit array is a 3 x 3 dipole array covering an area of 137.16 m (450 ft) by 213.36 m (700 ft). For maximum transmit efficiency the dipoles should be placed λ/4 or 110.89 ft above ground. This was not feasible due to the unreasonable cost. Instead standard-sized power poles of length 65 ft were set into ground so that the dipole elements were 55 ft above the ground. The receiver array consists of three dipoles placed in an equilateral triangle where the distance between each receive antenna is around 1.5λ or 202.8 m. These receive dipoles are constructed and oriented identically to the transmit array dipoles. At this frequency sky noise is dominant so the signal-to-noise ratio of the receive signal will be constant regardless of the height of the receive antenna above the ground. The nominal height chosen for the receive array dipoles is 20 ft. The Platteville MF radar specifications are tabulated in Table 1-1.

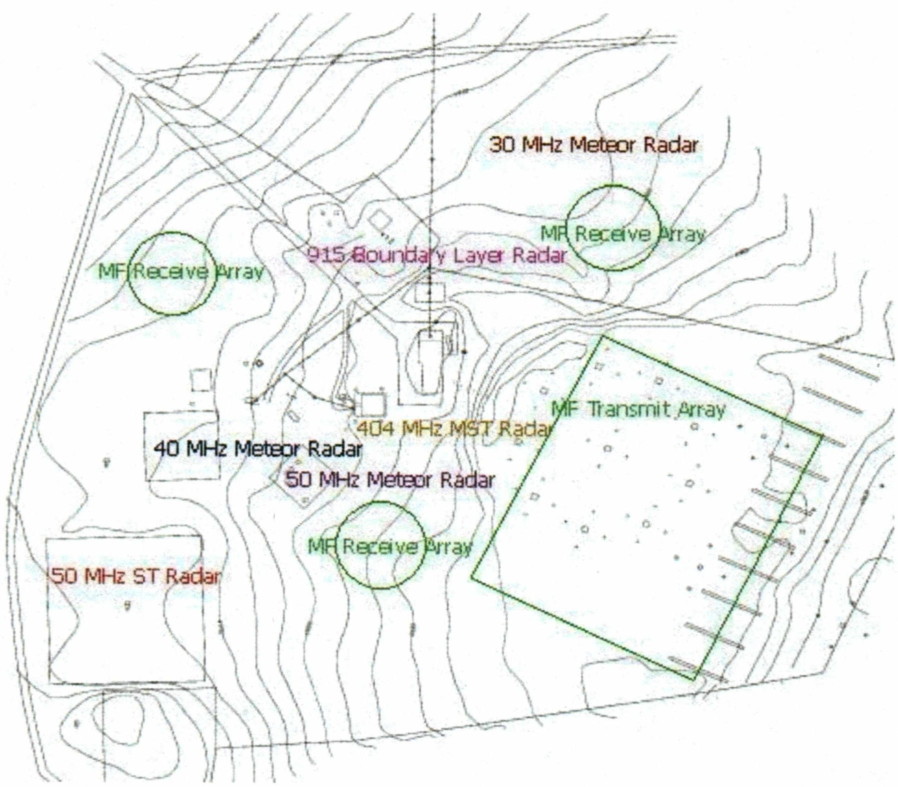



Figure 1-3 Location of MF radar at the Platteville Atmospheric Observatory [*Thorsen,* 2000].

Table 1-1 Radar specifications [*Thorsen,* 2000]

| | |
|---|---|
| RF Frequency: | 2.219 MHz |
| Transmit Peak Power: | 25 kW |
| Mode of Operation: | pulse (20 μs) |
| Antenna Gain: | TX: 15.6 dB, RX: 9.3 dB |
| Antenna Beamwidth: | TX: 32°x34°, RX: 61°x90° |
| Antenna Directions: | Vertical |

The MF radar at Platteville uses the spaced antenna full correlation analysis technique first introduced by *Briggs et al.* [1950] to estimate wind motion. This technique estimates the movement of random patterns observed by multiple non-collinear detectors. For the purpose of measuring the mesospheric winds, these random patterns can be thought of as a diffraction pattern of complex field strengths obtained at the ground from radio waves scattered at mesospheric heights. We assume that the atmospheric scatterers move along with the background wind. In order to measure the two-dimensional motion of the diffraction pattern, a minimum of three non-collinear receive antennas are needed. The analysis technique estimates the drift velocity and other characteristics of this pattern by evaluating the auto- and cross-correlations of the measured amplitude sequences. It is convenient to assume a Gaussian function for the auto- and cross-correlation functions [*Meek*, 1980]. Under this assumption the two-dimensional wind field and the diffraction pattern spatial and temporal scales may be determined by measuring the magnitude of the lag to the maximum of the cross-correlation functions and the lag where the auto-correlation function drops to 0.5.

The MF Radar estimates wind motion every 5 minutes given an adequate signal to noise ratio of the received signal. The data are most reliable during the daytime when the ionization of the atmosphere provides the maximum backscatter of the transmitted signal. The level of ionization in the atmosphere is time and height dependent. Typically above 81 km, the ionization level is large enough that we get wind estimates throughout the day.

Below 81 km the nighttime ionosphere disappears and hence we can obtain wind estimates at these lower heights only during the daytime from sunrise to sunset.

Figure 1-4 shows the percentage of five minute velocity estimates available at each height in January 2001. The maximum number of possible five minute estimates in a month with 31 days is 8928. For heights up to 79 km we typically acquire less than 40% of the maximum possible data. Above 79 km we typically acquire more than 50% of the maximum possible data. At no height do we acquire 100% of the possible 8928 measurements. At 85 km we have the most complete coverage with 72% of the maximum possible data.

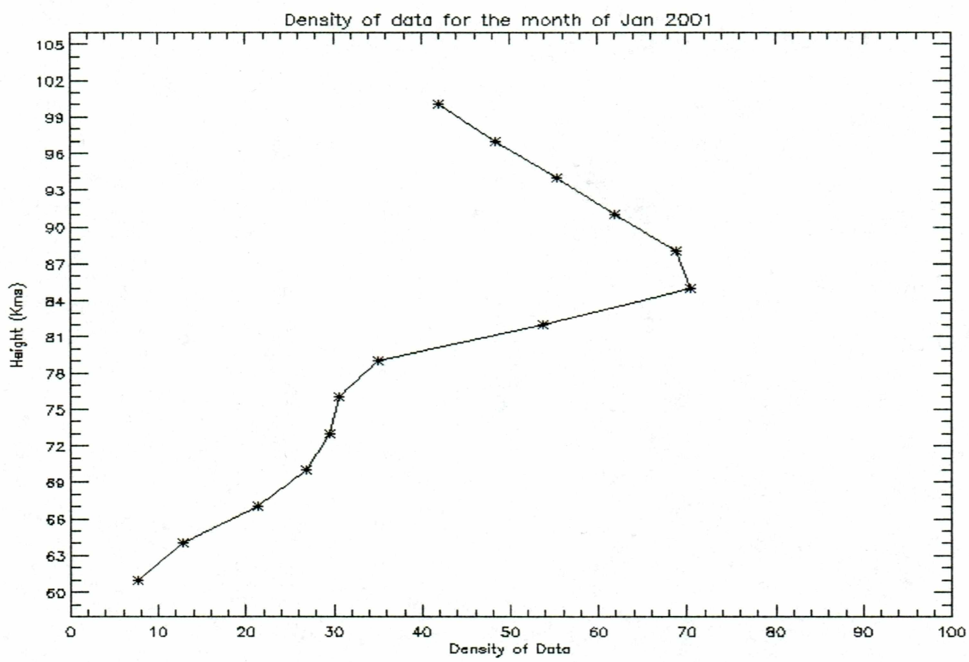



Figure 1-4 Percentage of data available at each height for January, 2001.

## 1.3 Overview of Thesis

In this thesis we investigate issues surrounding the time domain and spectral domain analysis of unevenly sampled MF radar data. The goal of this thesis is to provide a method for the automatic processing of the wind estimates into monthly mean winds and monthly mean tidal amplitudes and phases. Historically the MF radar data are analyzed in the time domain using a least squares fitting technique of the data to an *a priori* assumed model. In Chapter 2 we investigate a time domain analysis technique, harmonic least squares [*Press et al.*, 1992] and acquire criteria to assess the validity of the fitted parameters. In this chapter we apply the harmonic least squares routine to different wave models, and different averaging preprocessing (raw 5-minute, hourly averages, month-day data) to assess the robustness of the processing. The calculation of error using the Monte Carlo method is obtained and compared with the error estimation obtained from the harmonic least squares routine. In Chapter 3 we investigate two spectral methods (FFT and Lomb-Scargle) and their limitations in analyzing unevenly sampled data. We also assess the validity of the *a priori* model assumed in Chapter 2. Based on the results of Chapters 2 and 3, in Chapter 4 we provide a framework for the automated analysis of the MF radar data. We then apply this routine to the four years of Platteville data we have thus far acquired. The results of the Platteville data analysis is compared with the GSWM-02 and archival data from similar MF radars at Saskatoon and Urbana. In Chapter 5 we review the results of the previous chapters and provide suggestions for future work.

# 2 Time Domain Analysis of the MF Radar Data

Historically, medium frequency (MF) radar data are processed in the time domain using a least square fitting method to obtain the tidal parameters [e.g., *Manson et al.*, 1985, *Franke and Thorsen*, 1993]. This method of processing and obtaining tidal parameters assumes an *a priori* model with specific wave periods (eg. 12 hour and/or 24 hour periods), which may or may not be valid. In Chapter 3 we will investigate the validity of the tidal model used in this chapter. The MF data at Platteville are not uniformly sampled. The number of data points is neither constant across each height nor across every day. The data are more available at upper heights than at lower heights and more available during the day than at night. The completeness and distribution of valid data affect how well the model can be fit to the data. Given the characteristics of our real data we need to determine what criteria to apply to decide whether or not the fit parameters we obtain are reasonable.

In this chapter we start with a description of raw MF velocity measurements. As stated above, these data are non-uniform and their non-uniformity is variable in height and time. We will also discuss two preprocessing methods that produce time averaged velocity data. Time averaged data are useful because they can be more uniformly sampled. Next we will provide a theoretical description of the harmonic least square fitting method [*Press. et al.*, 1992, *Palo*, 2003]. This will be followed with an investigation of the criteria for determining a valid fit. After determining the criteria for a valid fit we will apply these criteria to compare different tidal models and preprocessing methods. Since the errors on the raw velocity data are unknown the meaning of the error estimates derived from the fitting routine are also unknown. To clarify the meaning of the error estimates we compare those estimates with estimates obtained using a Monte Carlo method. Finally given the results of this chapter we will present a method for automated processing of the MF velocity data.

## 2.1 Platteville MF Radar Data

The MF radar data collected at Platteville, Colorado are processed online using the full correlation analysis technique [*Briggs et al.*, 1950] to obtain horizontal velocity estimates every five minutes. We currently have archived the raw radar data from this radar over 2000-2003, a four year period. Whether or not the online processing produces a valid velocity estimate depends on the state of the radar and the state of atmosphere. Obviously we do not get velocity estimates when the radar is not operational. This may be because of power failures, instrument failure, etc. Additionally, excessive environmental noise (e.g. lightning strikes) can cause a non-valid estimate of the wind. These are both randomly occurring events that affect all heights uniformly. The amount of backscatter power received by the radar can also affect whether or not a valid estimate can be obtained. The backscatter power is dependent in part on the electron density of the ionosphere. In the height range observed by the MF radar the ionospheric electron density has a diurnal variation. This leads to a diurnal variation in the number of valid estimates obtained over the course of a day.

Figure 2-1 and Figure 2-2 show the average number of raw velocity estimates in an hour bin for a winter (Dec., Jan., Feb., 2000-2001) and summer (Jun., Jul., Aug., 2001) seasons, respectively. The maximum number of possible five minute estimates in an hour is twelve. In each hour bin, at each height we average the total number of five minute estimates across each day contained in those three months. We can observe that at higher altitudes above 78 or 81 km range, the data are available for the entire 24-hour duration, whereas at lower heights data are only available during the daytime. However, even at the upper heights we consistently obtain a larger percentage of the possible data during the day with fewer valid estimates at night. This is attributed to the diurnal variation in the electron density as discussed above. We can clearly see the transition

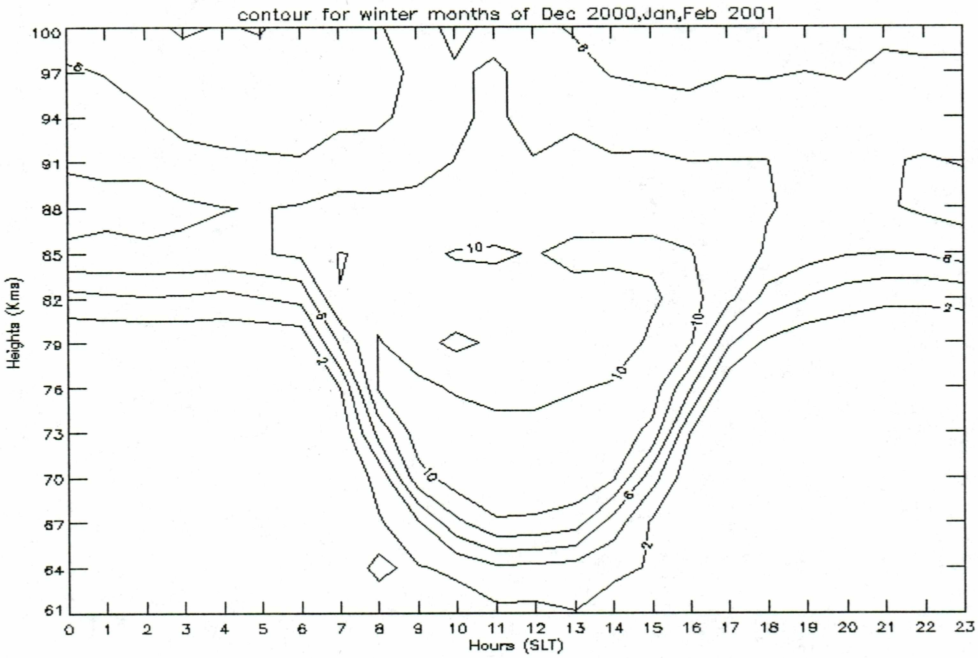



Figure 2-1 Contours of average number of data points available per hour for winter season (Dec. 2000, Jan., Feb. 2001).

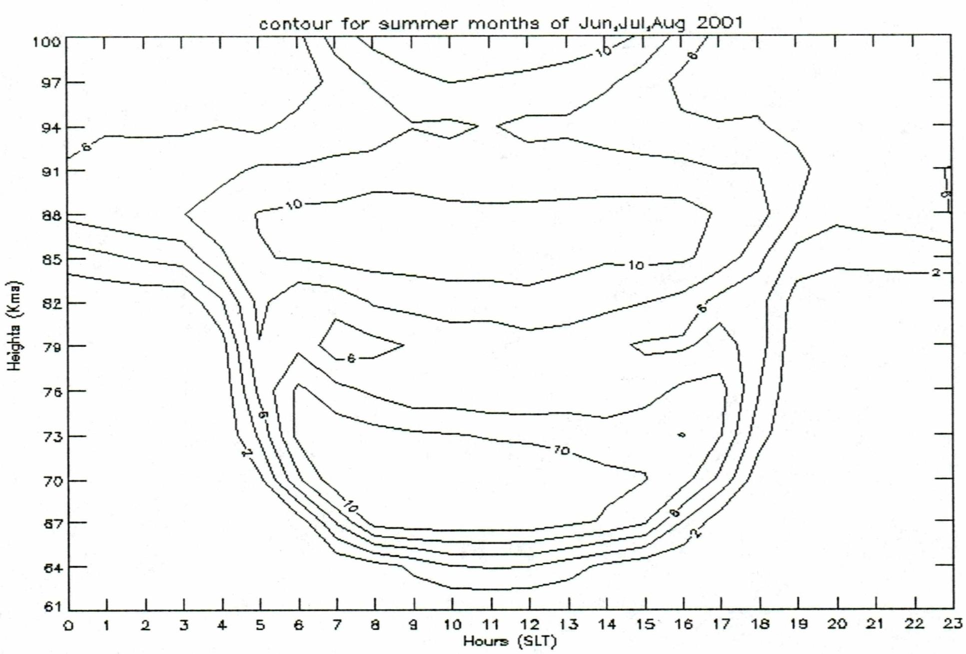
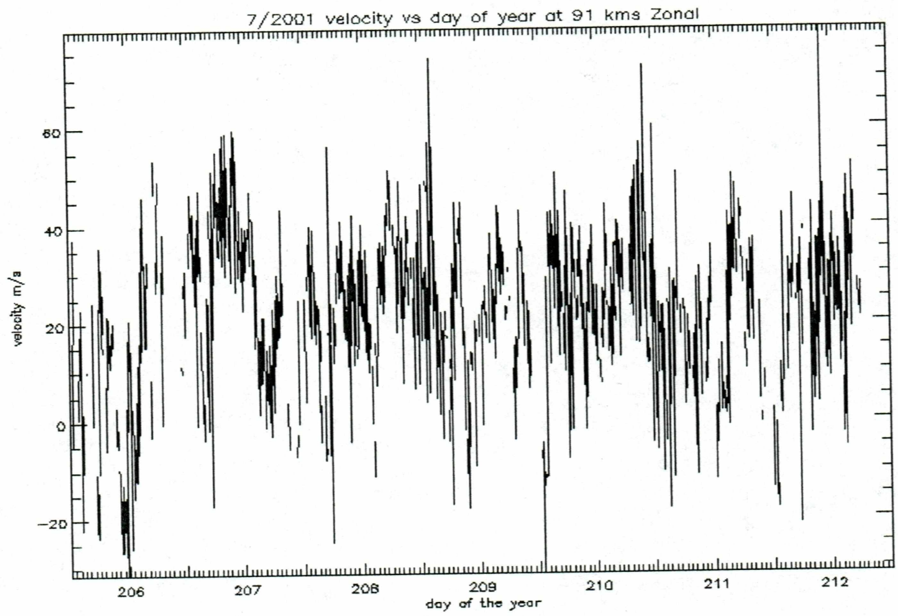



Figure 2-2 Contours of average number of data points available per hour for summer season (Jun., Jul., Aug. 2001).

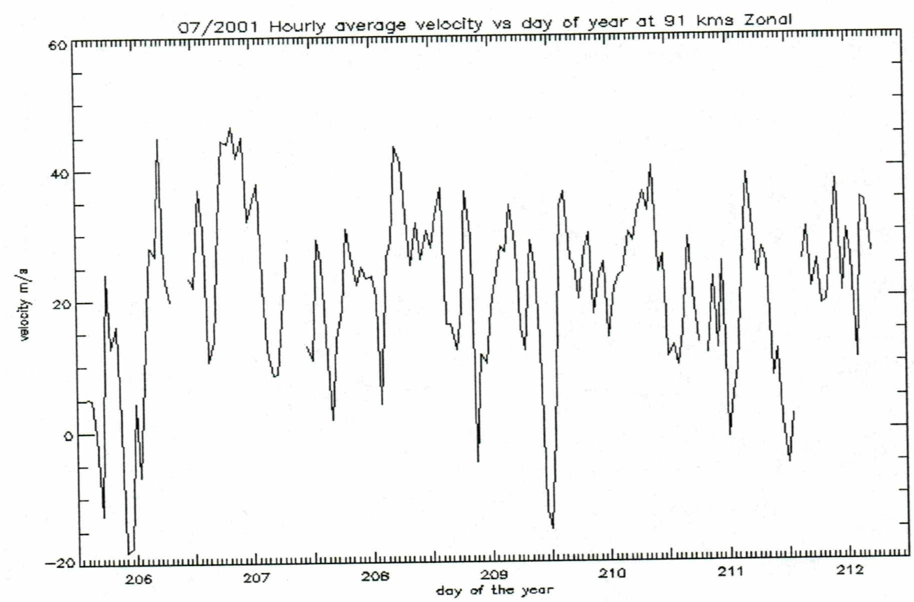
from the lower heights to the upper heights. We can also see that there is more hourly coverage at lower altitudes during the summer months than the winter months. This can be attributed to more daylight during summer and hence more ionization of the atmosphere.

Figure 2-3(a) shows the raw five minute velocity estimates in the zonal (eastward) direction at 91 km for the last week of July 2001. The y-axis shows velocity in m/s and the x-axis shows the day of the year where solar local noon is indicated by the day number. The velocity estimates range from -30 m/s to 120 m/s with an average motion of 20 m/s eastward. Although not easily seen here there are numerous missing data points in this sequence. The missing data points, or gaps in the continuity of the sampled data, make the raw five minute velocity estimates appear unevenly sampled. Figure 2-3(b) shows the hourly average of the raw data contained in Figure 2-3(a). Each data point represents the average of the five minute estimates within an hour bin where at least four estimates are available. The effect of time averaging is to make the velocity time series more evenly sampled at the cost of loosing the higher frequency components of the signal. Given our interest in tidal periods removing the higher frequency components is not detrimental to our analysis. Although the hourly average data sequence is more evenly sampled we can still see four times where data are missing during just this one week period. There are several harmonic oscillations present in this data. Specifically one can clearly see a semi-diurnal oscillation.

Figure 2-4(a) shows the five minute raw zonal wind estimates for the last week of July 2001 at a height of 70 km. The x and y-axes are similar to those mentioned in the previous paragraph. The range of the velocity estimates is from -80 m/s to 15 m/s. The average motion is of -40 m/s in the westward direction. We can clearly see that there are long periods of time at night where valid data do not exist. The data are present only during the daytime. This is an effect of the low ionization during nighttime. Similarly Figure 2-4(b) shows the hourly average of the data at 70 km for the same period as shown

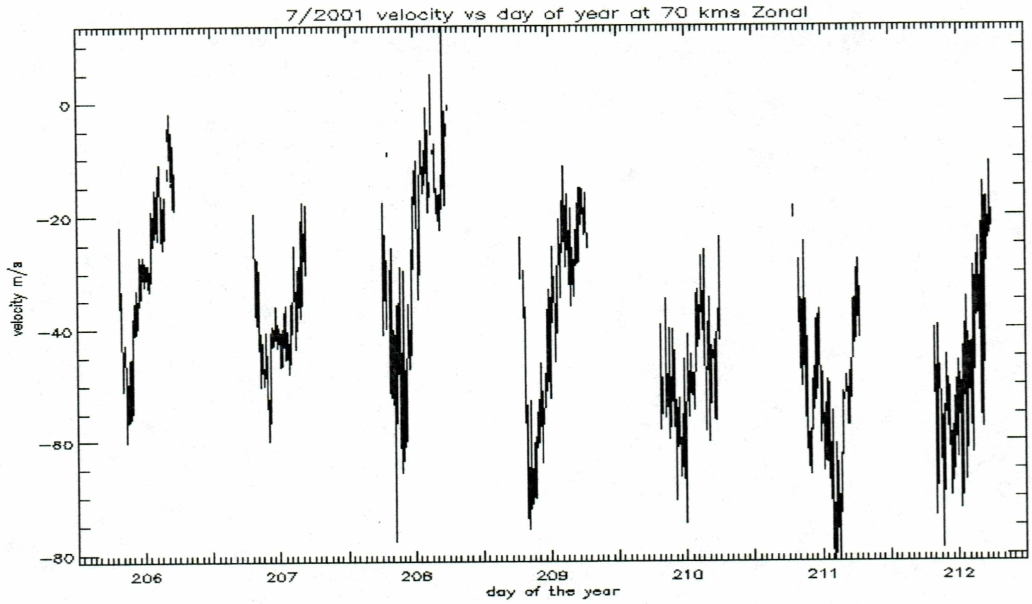



(a)

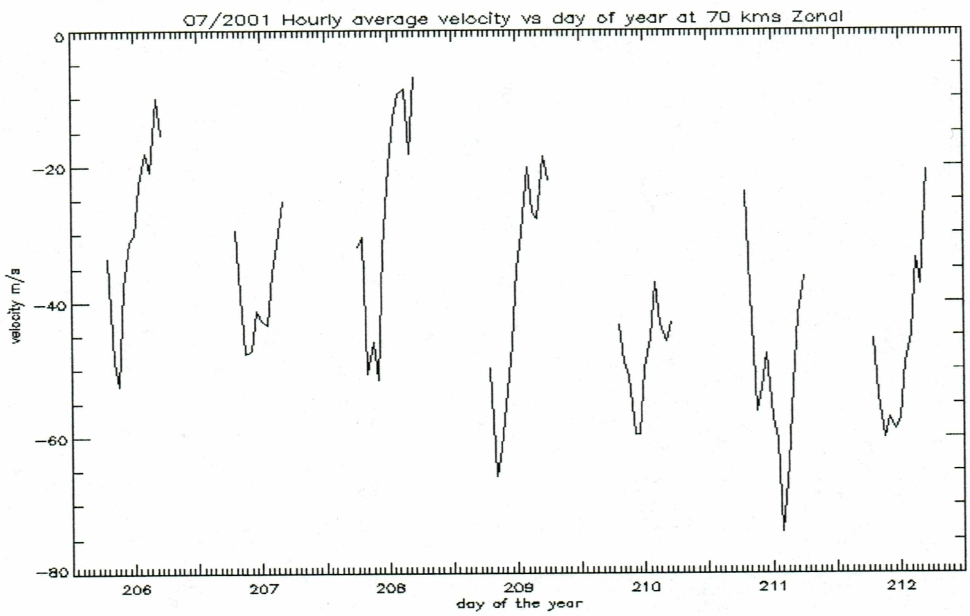



(b)

Figure 2-3 Zonal (eastward) wind estimates for the last week of July 2001 at 91 km (a) five minute velocity estimates (b) hourly average velocities. The location of day number indicates time of SLN.




(a)




(b)

Figure 2-4 Zonal (eastward) wind estimates for the last week of July 2001 at 70 km (a) five minute velocity estimates (b) hourly average velocities. The location of day number indicates time of SLN.

in Figure 2-4(a). The major times of missing data in the hourly averages are the same as those in the raw data. The nighttime gaps in the data limit our ability to determine the diurnal and semidiurnal tidal fits.

Since we are interested in obtaining tidal fits for wave signatures that are coherent over an entire month it is useful to create a composite month-day average. The composite month-day is created for each height by averaging all the data within an hour bin across the entire month. Again, hour bins with less than 4 data points are rejected. The month-day average zonal winds for the months of January 2001 and July 2001 are shown in Figure 2-5 and Figure 2-6, respectively. The x-axis is the duration of a day and the y-axis is the heights for which we have estimates. Each division is 50 m/s, and the bottommost line for the July 2001 plot corresponds to a height of 64 km. The remaining lines represent data from increasing heights as indicated on the y-axis. The plot also gives us the number of days we have valid estimates and there are at least four valid estimates out of a maximum of twelve estimates possible in an hour.

Clearly seen is the near semi-diurnal oscillation in January above 82 km. The trough is at 5 hours at 84 km and the peak at around 13 hours. We can also see the phase progression with height indicating an upward propagating wave. Above 85 km in both January and July the data are available for the entire 24 hours. Using this averaging scheme the averaged data are uniformly sampled. Below 85 km we only obtain daytime estimates as before. In the summer month the data are available for more than 15 hours, even at lower heights from 64 km up to 82 km, whereas in the winter month the data are available for only 11 hours at the lower heights from 61 km to 76 km. The availability of data for more than 15 hours even at lower heights during summer months can be attributed to the fact that there is a greater ionization effect in the atmosphere during these months.

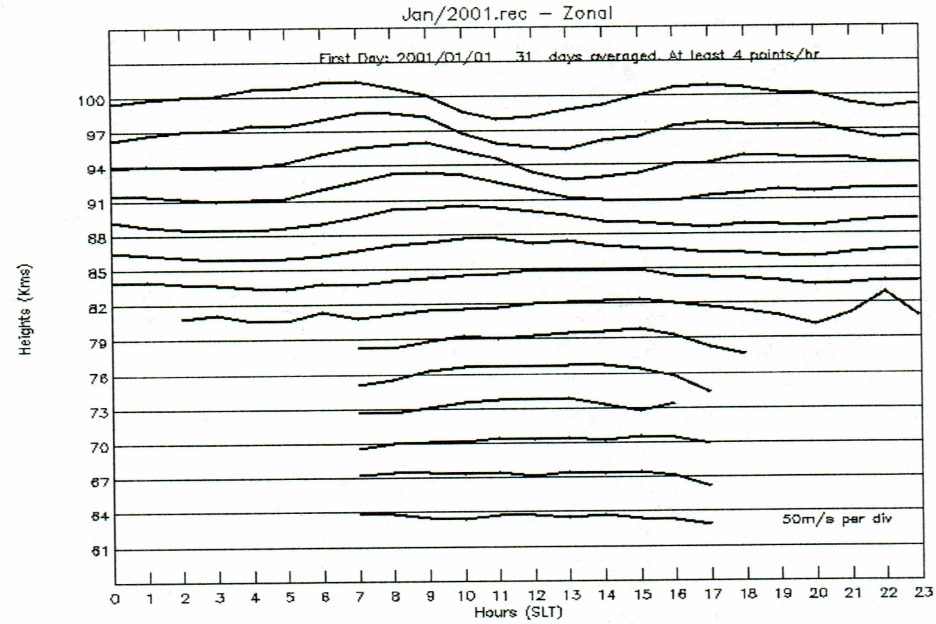



Figure 2-5 Zonal wind profile of January 2001.

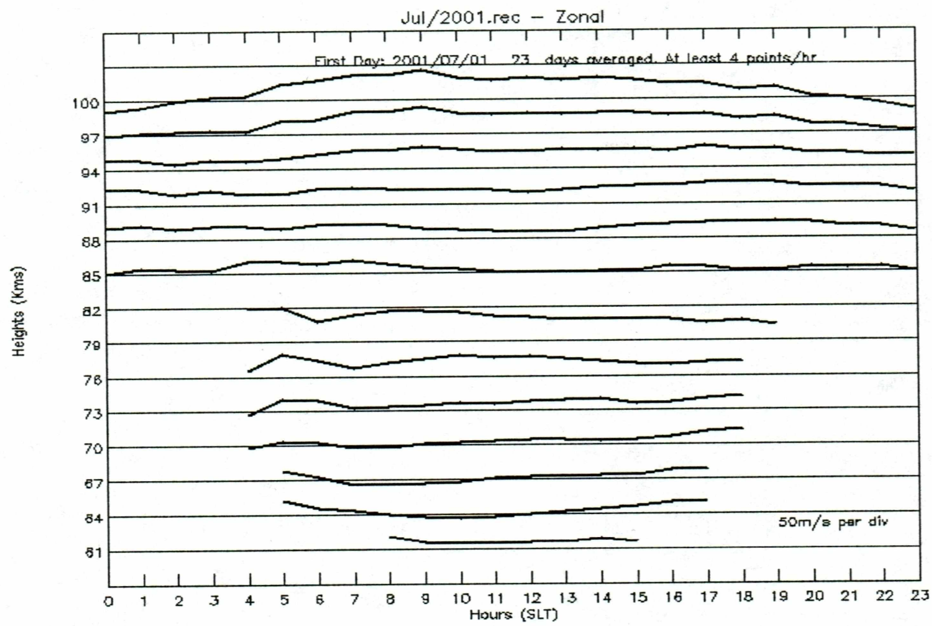



Figure 2-6 Zonal wind profile of July 2001.

## 2.2 Harmonic Least Square Routine

The harmonic least squares fitting routine that we are using has been provided to us by *Palo* [2003]. Here we explain how the routine works. Equation (2.1) shows our model where *y(t)* represents our data, and $T_k$ represents the possible periods (12 hr = semidiurnal, 24 hr = diurnal) that we are assuming are the primary modes existing in our data.

$$y(t) = \sum_{k=0}^{M} a_k \cos\left(\frac{2\pi\, t}{T_k}\right) + b_k \sin\left(\frac{2\pi\, t}{T_k}\right). \tag{2.1}$$

The unknown parameters $a_k$ and $b_k$ are to be determined.

The maximum likelihood estimate of the model parameters is obtained by minimizing the quantity

$$\chi 2 = \sum_{i=1}^{N} \left[ \frac{y_i - \sum_{k=0}^{M} a_k \cos\left(\frac{2\pi\, t_i}{T_k}\right) + b_k \sin\left(\frac{2\pi\, t_i}{T_k}\right)}{\sigma_i} \right]^2 \tag{2.2}$$

which is also known as the chi-square or merit function. In this equation $y_i$ is our sampled datum taken at $t_i$, and $\sigma_i$ is the measurement error (standard deviation) of the *i*th data point. If the measurement errors are not known, they may all be set to a constant value. We do not know the errors associated with the raw velocity estimates and therefore set $\sigma = 1$.

The best fit parameters $a_k$ and $b_k$, are obtained by minimizing $\chi^2$. The minimization is organized as follows. First we need to build the basis subspace matrix, *H*. *H* is an *N* x *2M* matrix where *N* is the number of data points in *y* and *M* is the number of primary modes that we are fitting to. For example if we are trying to fit to both the semidiurnal (12 hour) and diurnal (24 hour) tides, the first row of *H* would look like:

$$H = \begin{bmatrix} \cos\left(\frac{2\pi\, t_0}{12}\right) & \sin\left(\frac{2\pi\, t_0}{12}\right) & \cos\left(\frac{2\pi\, t_0}{24}\right) & \sin\left(\frac{2\pi\, t_0}{24}\right) \end{bmatrix} \tag{2.3}$$

where $t_0$ is the time of the first data point and $M$=3 because the mean component is always assumed. Each subsequent row in $H$ is identical except that $t_0$ is incremented until the times of all data points are included in the matrix.

The solution matrix is obtained through the following equation:

$$x = \left(H^T H\right)^{-1} H^T y \tag{2.4}$$

where $x$ is a matrix containing the estimated parameters $a_k$ and $b_k$ associated with the magnitudes of the quadrature sine and cosine components of each period. The total amplitude of the wave is given by the square root of the sum of the squares of each component part. The phase is given by the arctan of the ratio of the sine to cosine components. The variances of our estimated parameters are given by the diagonal elements of the covariance matrix:

$$\mathrm{cov} = \left(H^T H\right)^{-1}. \tag{2.5}$$

The condition number of $H^T H$ gives us a measure of how singular the matrix is. When the condition number is large the matrix is ill-conditioned and may produce erroneous fits.

## 2.3 Criteria for Applying the Least Square Fitting

The least squares method assumes a model and fits the data to that model. We have initially used the month-day data because the data are mostly complete in the sense they are uniformly sampled and that the span of the data available during a 24-hour period is obvious (see Section 2.1). The number of data points available is varied with respect to each height, meaning that at lower heights there is less than 24-hour coverage. There are a maximum of 24 data points possible at each height using the month-day data; we can fit a 12-hr wave only if we have at least six data points [*Crary and Forbes,* 1983].

Similarly we can fit a 24-hr period wave only if we have at least 12 data points. If there are not enough data points available, the extracted tidal parameters will contain errors [*David and Forbes*, 1983]. From these we can say that we can fit both 12-hr and 24-hr periods at upper heights and only a 12-hr period at lower heights.

We know that we need to have a specific minimum number of hours of data to fit and obtain the tidal parameters. The problem of using the five minute raw and hourly average data directly is that we do not immediately know the hourly coverage of data across a day. This makes it difficult to use the number of hours covered during a day as a criterion for determining which tidal model to use (24 hour and 12 hour or only 12 hour); The condition number obtained through the harmonic least squares fitting routine can tell us if the fit is suspect. At this point we need to know at what specific value of the condition number we have confidence that the fit is valid. To answer this question we have applied a least squares routine to the composite month-day data so as to compare the condition number with the hourly coverage.

In Figure 2-7 we compare the condition number versus the span of hours with valid data in the month-day averages. We have generated monthly fits for a two-wave model (12 and 24 hour period) and a one-wave model (only 12 hour) for every month where data were available from 1999 to 2002. Each height is fitted separately. Remember that the lower heights (below 84 km) typically do not have 24 hour coverage even in the month-day average. In the upper panel the condition number is plotted as a function of number of hours with valid data. The 'x' mark is used to plot the two-wave model condition number and the '+' symbol is used for the one-wave model condition number. We have focused the plot for condition numbers less than 20 in order to clearly see the transition between low condition number and high condition number. There are 34 cases with condition numbers greater than 20, with a maximum condition number of $1.25 \times 10^{18}$ for the 2-wave model and $6.211 \times 10^{18}$ for the 1-wave model. In the lower panel of Figure 2-7 the number of attempted fits is plotted as a function of the number of hours with valid

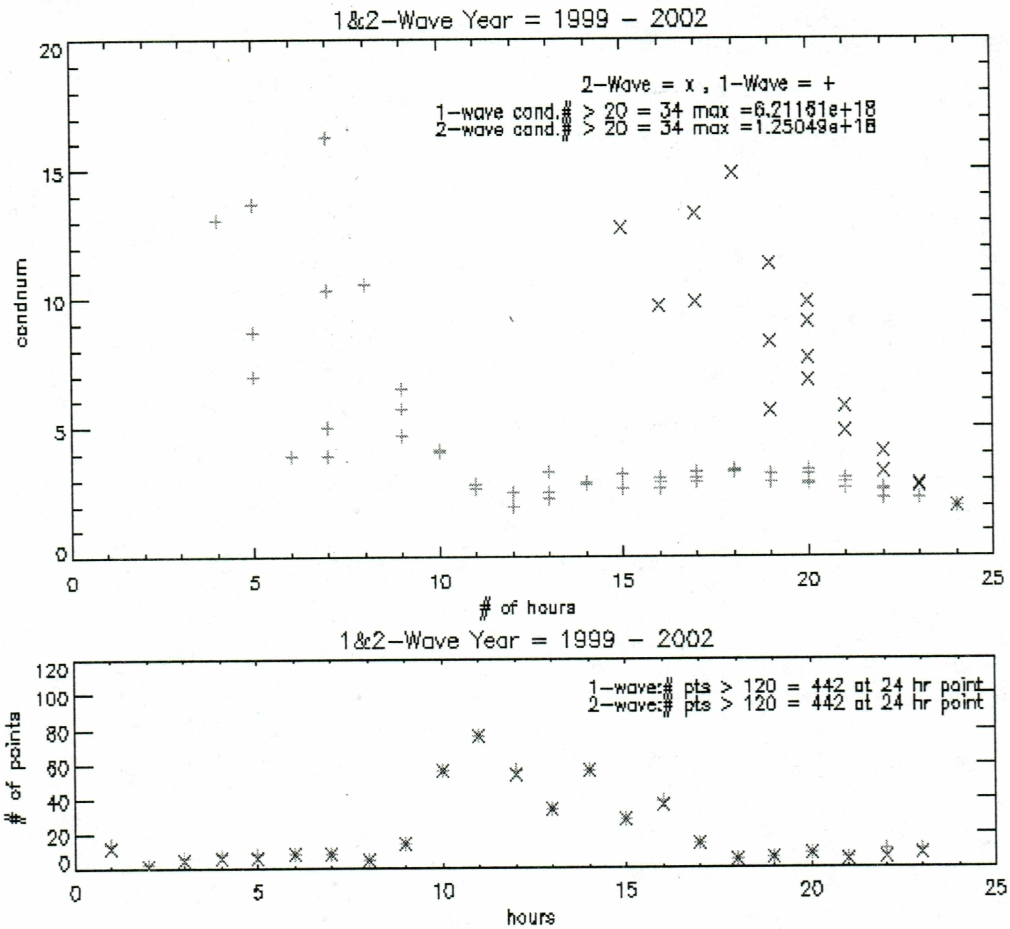



Figure 2-7 Hours versus condition number for the entire data.

data. For most of the time the data are available for all 24 hours. Note, the number of heights for which 24-hour data are available is 442. At lower heights data are available only during the daytime. Therefore, there are a small number of heights for which the hourly coverage is between 9 and 16 hours.

Figure 2-7 shows that for the two wave model (24 and 12-hour period) the condition number is small when the number of hours is more than 12 or 13. As the number of hours decreases the condition number increases. At around 10 hours the condition number is as high as 8000. For the one wave model (12-hour only) the condition number is small when the number of hours is greater than 6 or 7 hours. Again as the number of hours decreases the condition number increases. Below 5 hours of data the condition number is high, and is more than 10000 where there are only 3 or less than 3 hours of data (not shown).

From these observations we conclude that for a one wave model, fitting only a 12 hour sinusoid, the minimum number of hours data must be available can be around 6 or 8 hours and for the two-wave model, fitting both the 12 and 24 hour sinusoids, it must be around 16 or 17 hours. Observing Figure 2-7 we have decided to take the minimum number of hours as 15 hours for the two-wave and 8 hours for the one-wave model. This criterion cannot be worked out with the raw and hourly average data since we do not know exactly the number of hours of data that are present. So looking into Figure 2-7, we have considered taking the criterion of assuming a valid fit only if the condition number is less than 10 while obtaining fit parameters for the 24-hour and 12-hour tides.

## 2.4 Model Comparison of Least Square Fitting Method

In the previous section we determined the rejection criteria for the condition number generated by the harmonic least squares fitting routine. In this section we use that condition number rejection criteria to assess the difference between fitting two harmonics (12 hour and 24 hour) simultaneously, the two-wave model, and fitting each harmonic separately, the one-wave model. The desire is to have a robust estimate over a large range of heights. Even before we begin we are reminded that we cannot fit a 24 hour harmonic to the lower heights because we do not have the hourly coverage throughout the day. For these comparisons we use data from a winter month (January, 2001) and a summer month (July, 2001) that have been averaged into a composite month-day.

Figure 2-8 shows the estimated monthly mean wind for each height for January and July, 2001. The square (black) boxes are derived from the harmonic least squares fitting routine using the two-wave model. Notice that this model only gives estimates

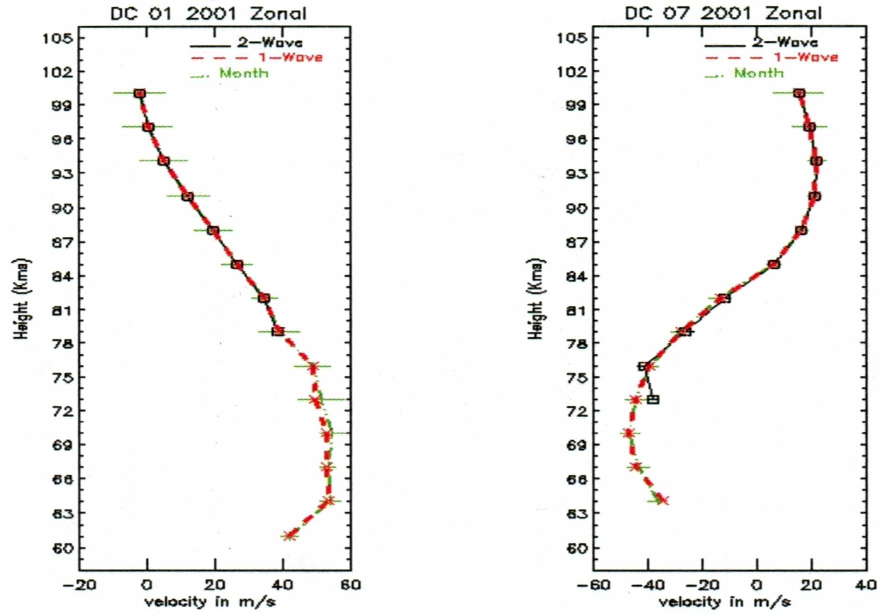
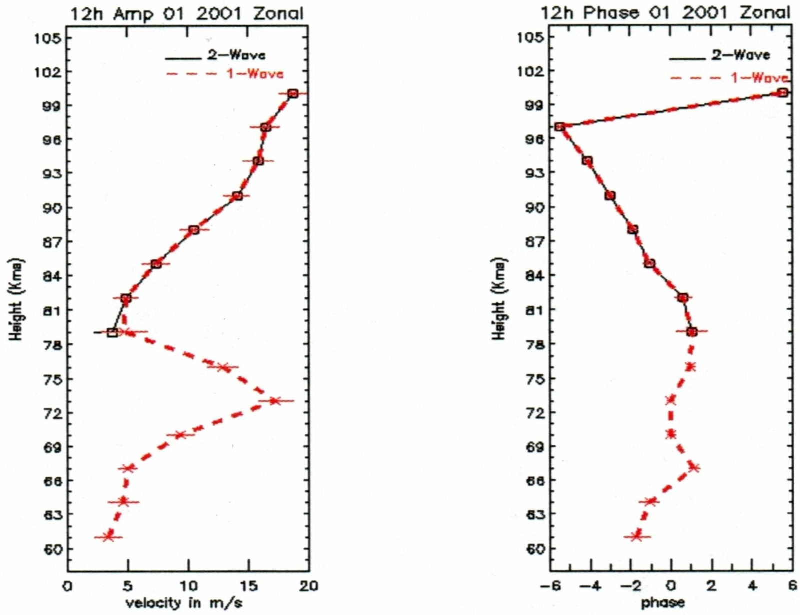



Figure 2-8 Mean velocity plotted as a function of altitude measured at Platteville, CO in January and July 2001.

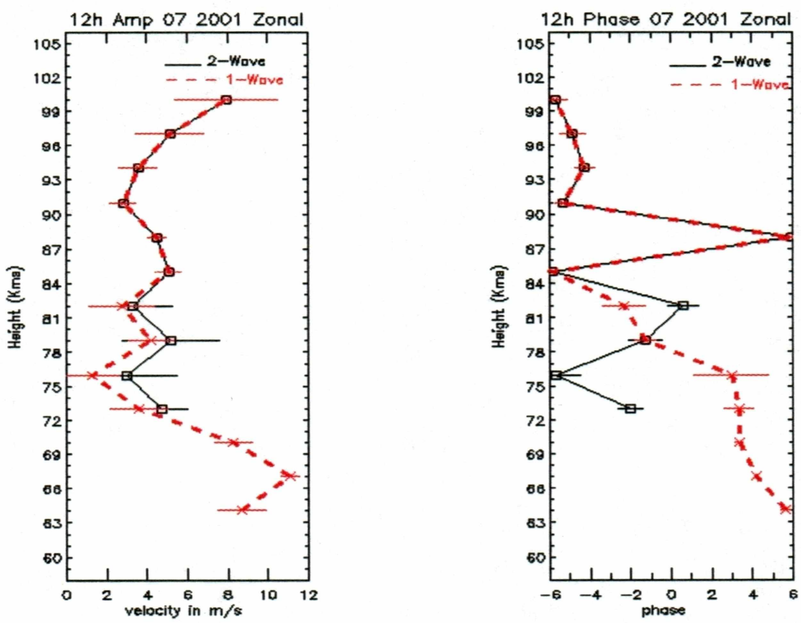
down to 79 km in January and 73 km in July. The (red) crosses are derived from the one-wave (12 hour only) model. Fitting for only the 12 hour harmonic generates valid fits down to 61 km in January and 64 km in July. Finally we also compared the mean wind fits derived from the harmonic least squares routine to a straight average of the data across the month. The straight average (green) is visible by the horizontal lines reflecting the variance of the data. The variance of the straight average is larger than that derived by the harmonic least squares routine because it includes the variability of the tides. Note that all fits are easily within 1 m/s of each other, with the exception of the two-wave model at 73 km in July, 2001. In January 2001, the one-wave and straight average show a maximum eastward velocity of 55 m/s at 73 km which reduces to around 0 m/s by 97 km. In July those same models show a westward jet (-45 m/s) maximizing at 70 km with a reversal at around 85 km and an eastward jet (20 m/s) maximizing at 94 km.

Figure 2-9 shows the amplitude and phase of the 12-hour semidiurnal tide estimated using the two-wave model and the one-wave (12 hour only) model for January (a) and July (b), 2001. As seen before the one-wave model allows for estimates across a larger range of heights. In January both models show a propagating semidiurnal tide with amplitude that grows with increasing altitude (5 m/s at 79 km to 20 m/s at 100 km). The time of maximum of this tide comes earlier in the day with increasing height from about 1 pm (1 hour) at 79 km to 6 am (-6 hour) at 100 km. Below 79 km the two-wave model does not produce a valid fit. However the one-wave model produces valid fits to 61 km in January. The one-wave model shows a 12 hour wave whose time of maximum does not change much with height, remaining around noon or becoming slightly delayed in height until 79 km. The amplitude of this wave maximizes at around 17 m/s at 73 km, decreasing in amplitude until 79 km, after which the amplitude starts to grow.

In July the estimates from the one-wave model fit and the two-wave model fit agree well above 85 km. Both show a wave whose time of maximum ranges between 6 am (-6 hours) and 8 am (-4 hours) and a wave amplitude that ranges from 3 to 10 m/s. Below 85 km the two different models diverge. There are four heights below 85 km where the two-wave model still presumes to give valid estimates. For these heights the amplitude of the two-wave fit and the one-wave fit differ a maximum of 2 m/s. It is understandable that the fitting routine would have difficulty given the small amplitude at these heights (around 3 m/s). The fitting routine shows its difficulty by the large error bars at these heights. We note that even though the amplitudes derived from the two models differ at these heights their difference is well within the error bars of the estimate. The estimated time of maximum for the four overlapping heights below 85 km differs between the two models by a maximum of 5 hours (nearly in quadrature) at 73 km. Again one notes that if the amplitude of the wave is small its phase must necessarily be suspect.




(a)



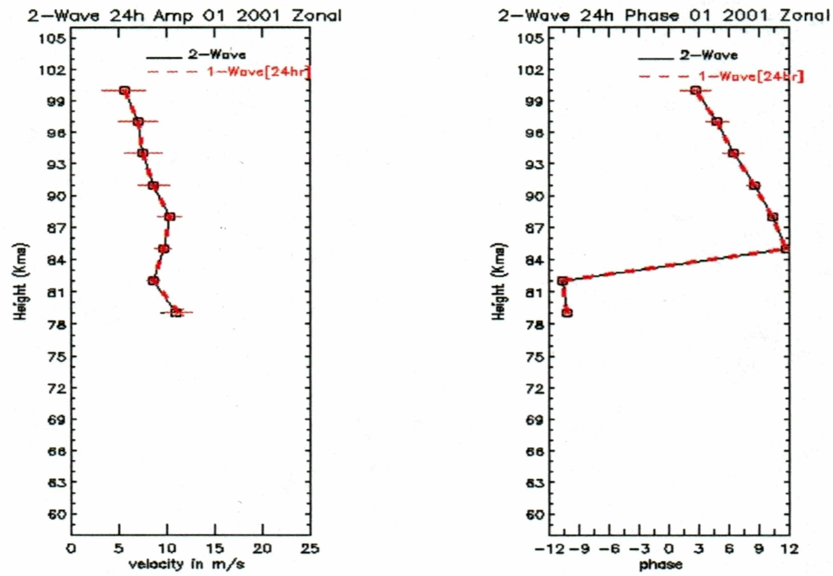

(b)

Figure 2-9 12-hour amplitudes and phases of two-wave and one-wave (12-hour only) models (a) January 2001 (b) July 2001.

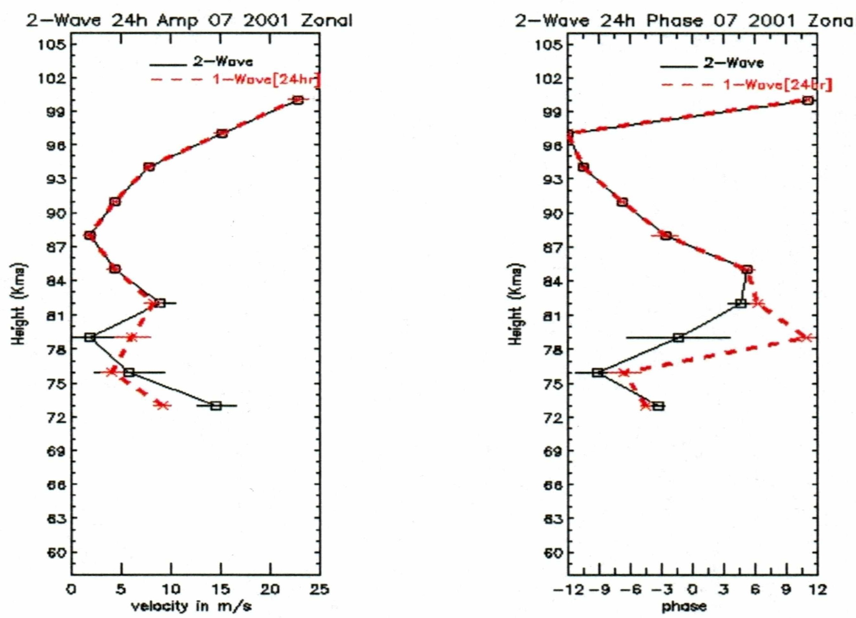
Figure 2-10 shows the amplitude and phase of the 24-hour diurnal tide estimated using the two-wave model and the one-wave (24 hour only) model for January (a) and July (b), 2001. Since both the two-wave model and one-wave model need to fit a 24 hour wave there is no advantage of the one-wave model in providing valid fits at additional heights. In January both models show a propagating diurnal tide with amplitude that decreases with increasing altitude (12 m/s at 79 km to 6 m/s at 100 km). The time of maximum of this tide comes earlier in the day with increasing height, ranging from a little after midnight (-11 hour) at 79 km to 3 pm (3 hour) at 100 km.

In July the estimates from the one-wave model fit and the two-wave model fit agree well above 85 km. Both show a propagating wave whose amplitude first decreases with altitude (2 m/s at 88 km) and then increases again (23 m/s at 100 km). The time of maximum of this tide comes earlier in the day with increasing height, ranging from a little after 6 pm (6 hour) at 85 km to just before midnight (-12 hour) at 100 km. Below 85 km the two different models diverge although the general trend persists. There are four heights below 85 km where the two models still presume to give valid estimates. For these heights the amplitude of the two-wave fit and the one-wave fit differ a maximum of 5 m/s. It is understandable that the fitting routine would have difficulty given the reduced hourly coverage at these heights. The fitting routine shows its difficulty by the large error bars at these heights. The estimated time of maximum for the four overlapping heights below 85 km differs between the two models by a maximum of 11 hours (nearly in quadrature) at 79 km.

As can be seen from the above comparison there is little difference between the one-wave model and two-wave model in the upper heights when we have 24 hour coverage in the data. The differences occur when we do not have 24 hour coverage in the data and become more severe as the data become more limited. There is a distinct advantage (larger number of heights with valid fits) from using the one-wave (12 hour only) model. We conclude that the best method would be to fit each wave separately.




(a)




(b)

Figure 2-10 24-hour amplitudes and phases of two-wave and one-wave (24-hour only) models (a) January 2001 (b) July 2001.

## 2.5 Processing Method Comparison for Different Data Formats

In the previous sections we have pre-processed the MF radar data into a month-day average and then used the harmonic least squares routine to fit that data to the monthly mean wind and monthly average diurnal and semidiurnal tides. In this section we would like to examine whether or not there is any advantage to pre-processing the data to a month-day average or if we could apply the harmonic least squares routine directly to the 5-min raw data and/or the hourly average data. The month-day average provided us with a concrete rationale to choose a particular condition number, greater than 10, as a rejection criterion in determining if the fit was valid. This rejection criterion was chosen by examining the average condition number when we had an adequate number of hours of data available for the fit. Now that a rejection criterion has been determined we can apply that criterion universally even if we do not know the coverage of data across a day.

Figure 2-11 shows the comparison of the monthly mean velocity derived using the three different averaging methods (no averaging, hourly averaging, month-day averaging) for January and July 2001. The monthly mean wind was derived simultaneously with the semi-diurnal tide by using the one-wave (12 hour only) model in the harmonic least squares routine. For most heights there is no discernable difference between the different averaging methods. In January 2001, between 70 km and 82 km, the monthly mean velocity derived using the month-day average tends to be less than those derived using the 5-min raw or the hourly average, with the largest difference of 5 m/s at 79 km. In July 2001, the largest difference in estimate from the various averaging methods occurs at 100 km with a difference of almost 7 m/s. The mean wind velocity tends to be decreasing as we go up in height for January and tends to be increasing during July. Using the 5-min raw data appears to provide a slight advantage in July by generating a valid estimate at one additional height.

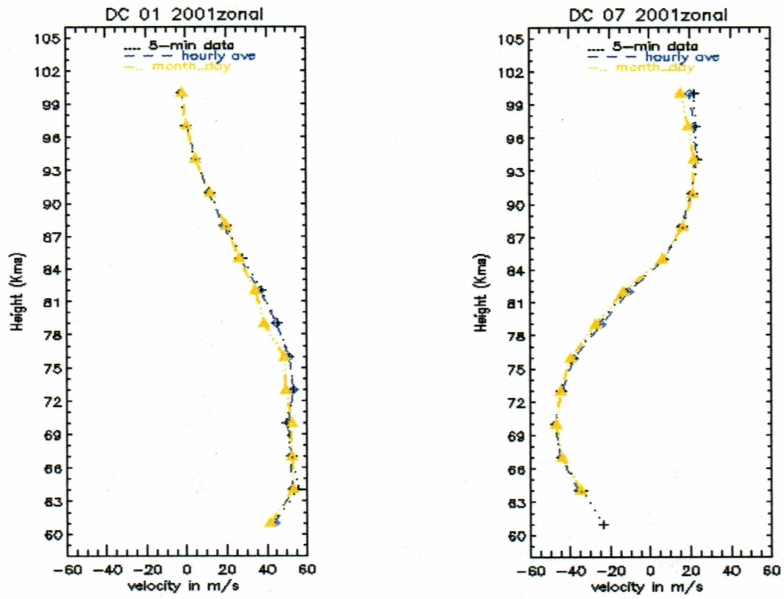
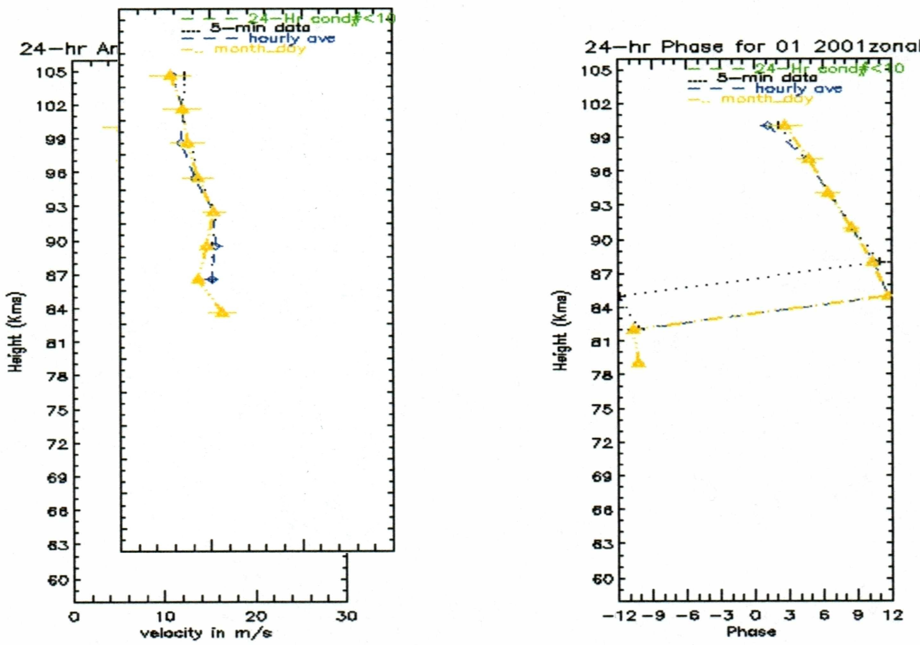
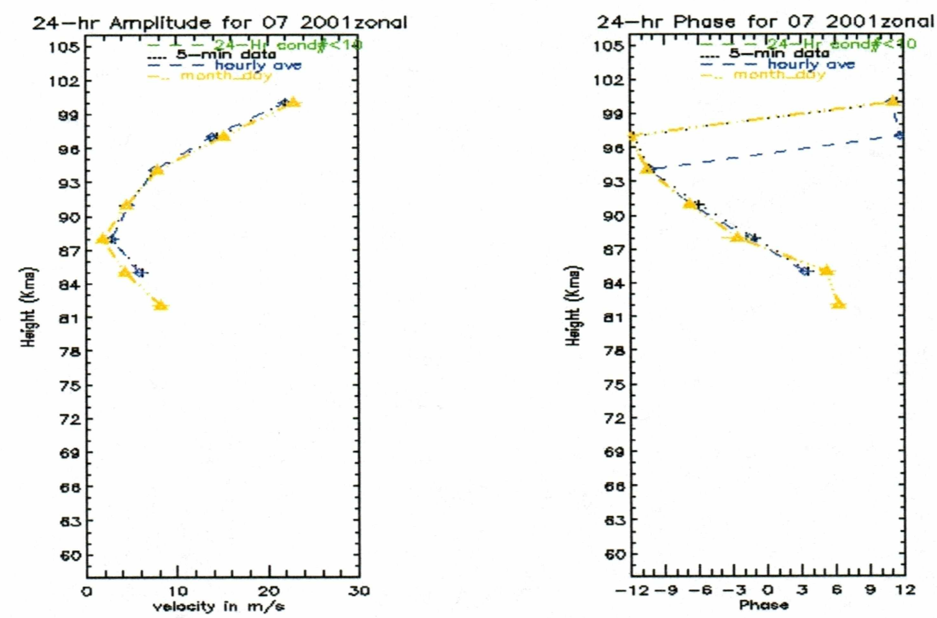



Figure 2-11 Comparison of mean velocity derived using three different averaging methods for January and July 2001.

Figure 2-12 shows the comparison of the diurnal tidal amplitude and phase derived using the three different averaging methods for January and July 2001. As previously seen, both months show a propagating diurnal tide, i.e. the time of maximum is progressively earlier with increasing height. In January the amplitude decreases with increasing height while in July the tidal amplitude increases with increasing height. Again the differences between estimates derived using the three different averaging methods appear to be minor. There is at most a 2 m/s difference in tidal amplitude and a little greater than 1 hour difference in the time of maximum derived using the three averaging methods. For the diurnal tide, using the month-day averaging method provides a valid estimate at one additional height.




(a)

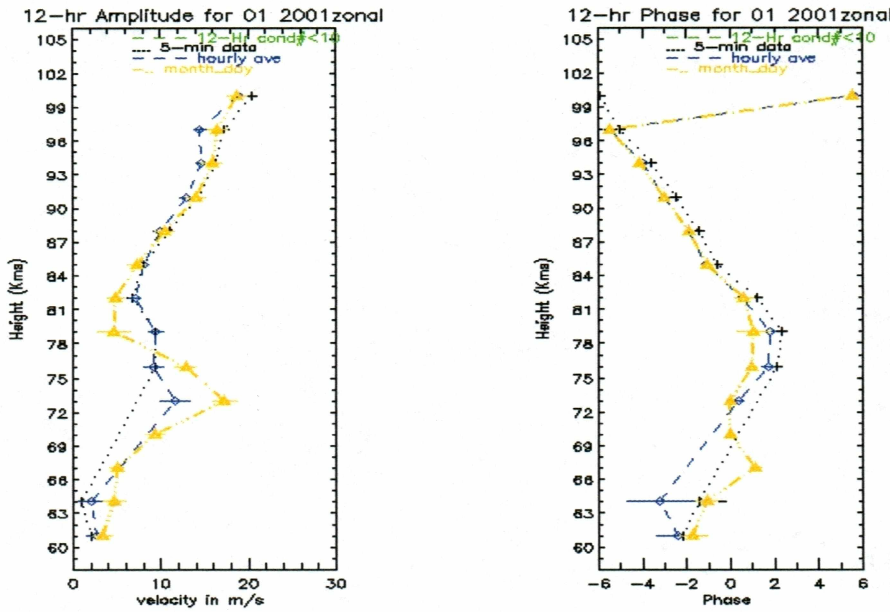



(b)

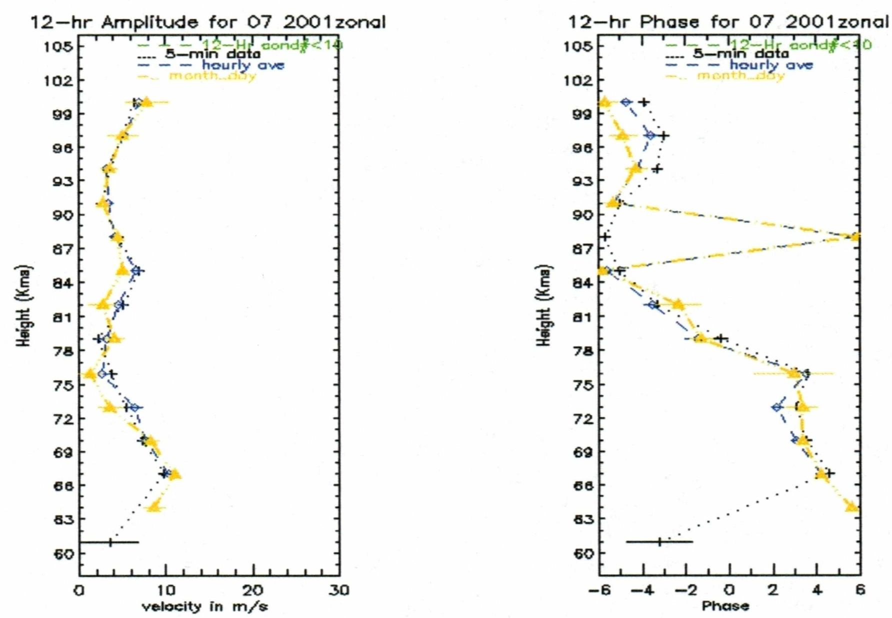
Figure 2-12 Comparison of the 24-hour amplitude and phase derived using three different averaging methods for (a) January, 2001 (b) July, 2001.

Figure 2-13 shows the comparison of the semi-diurnal tidal amplitude and phase derived using the three different averaging methods for January and July 2001. As seen previously January shows a propagating tide above 79 km while July shows a propagating tide below 88 km. In January the tidal amplitude increases with increasing height while in July the tidal amplitude is fairly constant over height. The differences between estimates derived using the three different averaging methods are slightly more evident for the semi-diurnal tide than they were for the diurnal tide. In January the tidal amplitudes derived using the three averaging methods differ by a maximum of 6 m/s at 73 km, with estimates derived using the 5-min raw data and hourly average data agreeing better than either does with estimates derived using the month-day average data. The largest difference in time of maximum estimates for the semi-diurnal tide in January is slightly more than 2 hours at 64 km. In July the tidal amplitude estimates differ at most by 3 m/s (73 km) and the time of maximum estimates differ at most by 2 hours (100 km).

The above comparisons do not show overwhelming evidence that any particular pre-processing averaging scheme is more advantageous than another. Given that, one might decide, for the sake of simplicity, just to use the 5-min raw data directly and avoid all pre-processing. However, there is a hidden advantage in using the month-day averaging pre-processing. This advantage has to do with the number of multiply operations it takes to formulate an estimate for a particular height (see Equation 2.4). A very crude estimate of the number of multiplies it takes to perform the required matrix multiplications is $(2M)^3 \times N^2$, where $M$ is the number of basis functions ($M$=2 for the one-wave model) and $N$ is the number of data points in the fit. For a particular height with 24 hour coverage this would give $64 \times (24)^2 = 36\ 864$ multiplications for a month-day average height, $64 \times (24 \times 31)^2 = 35\ 426\ 304$ multiplications for an hourly averaged height, and $64 \times (12 \times 24 \times 31)^2 = 5\ 101\ 387\ 776$ multiplications for a 5-min raw data height. The largest portion of time needed to perform a fit at a single height is dominated by the time required for matrix multiplications, since multiplications take up significantly




(a)




(b)

Figure 2-13 Comparison of the 12-hour amplitude and phase derived using three different averaging methods for (a) January, 2001 (b) July, 2001.

more computing time than additions. This means that even though we have to perform more up front additions in creating the month-day average, this method will still run faster since there are far fewer multiplications. One drawback in using the month-day average is that we are limited in creating only monthly average estimates of the mean winds and tides. At times we may want fits on a finer time scale, say weekly average estimates. For these estimates we will need to fall back to our hourly averaging pre-processing. Therefore in our code we will maintain both capabilities.

### 2.6 Error Analysis

We do not know the measurement error in the five minute raw data. The errors, obtained using the harmonic least squares routine, are therefore not clearly defined. To verify that the standard deviations calculated from the harmonic least squares routine are valid error estimates for the fitted parameters we compare those error estimates with estimates derived using a Monte Carlo bootstrap method [*Bradley*, 1982]. In order to determine appropriate error estimates of our fitted parameters one might try to make several measurements of the atmosphere while the atmosphere is held in a particular state. The differences in the measurements would then be solely due to measurement error. The underlying "true" state of the atmosphere could then be determined along with the measurement errors. Unfortunately we cannot hold the atmosphere in a particular state and therefore can obtain only one set of measurements for which we do not know our measurement errors.

The bootstrap method provides a method of creating synthetic data sets from our original data set that can then be used to estimate our measurement errors. The method proceeds as follows. For each height that we want to make a fit, call that height's data sequence $X_0$. This data sequence has $N$ data points. Create a large number, $S$, of synthetic data sequences, $X_1$, $X_2$, …by randomly selecting $N$ data points from the original sequence $X_0$. Because the $N$ data points are randomly sampled the synthetic data sequences will have a certain number of duplicated original data points. Each of these $S$ data sequences are passed through our processing routines in order to create $S$ estimates of each of the monthly mean wind, semi-diurnal tidal amplitude and phase, and diurnal tidal amplitude and phase. We then calculate the mean and standard deviation of each estimated parameter and compare that standard deviation with the original standard deviation obtained from the harmonic least squares routine. For our comparisons we have chosen to create $S = 100$ synthetic data sequences from our original data sequence.

Figure 2-14 shows the standard deviation of the monthly mean wind for January 2001 obtained by the harmonic least squares fit of the original data and by the Monte Carlo bootstrap method. Note that both methods give an error estimate of the monthly mean wind for January of around 1 to 3 m/s with the largest errors occurring at the lowest heights. This is to be expected as there are fewer data at the lower heights. There is less than 1 m/s difference in the standard deviation estimate between the two estimation methods.

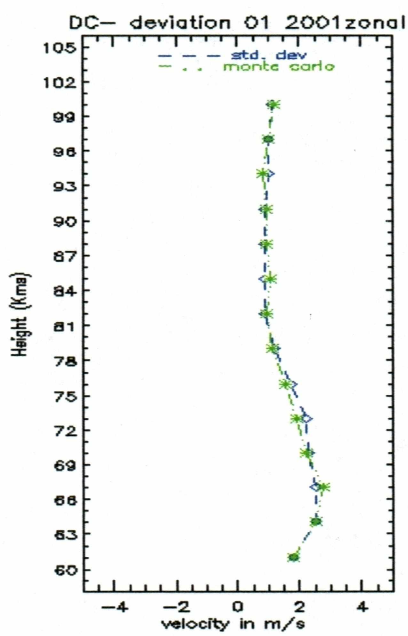



Figure 2-14 The DC deviations from the LS fit and the Monte Carlo deviation.

Figure 2-15 shows the standard deviation of the estimated diurnal amplitude and phase for January 2001 obtained by the harmonic least squares fit of the original data and by the Monte Carlo bootstrap method. Both methods give an error estimate for the diurnal amplitude of less than 2 m/s with the largest error occurring at 100 km. There is less than 1 m/s difference in the standard deviation estimate of the amplitude between the two estimation methods. Both methods give an error estimate for the diurnal time of maximum of less than 1 hour with the largest error also occurring at 100 km. The exception to this is at 82 km, where the Monte Carlo method gives a larger standard deviation for the time of maximum than that obtained in the harmonic least squares routine. The Monte Carlo estimate of the standard deviation for the time of maximum at this height is 2 hours.

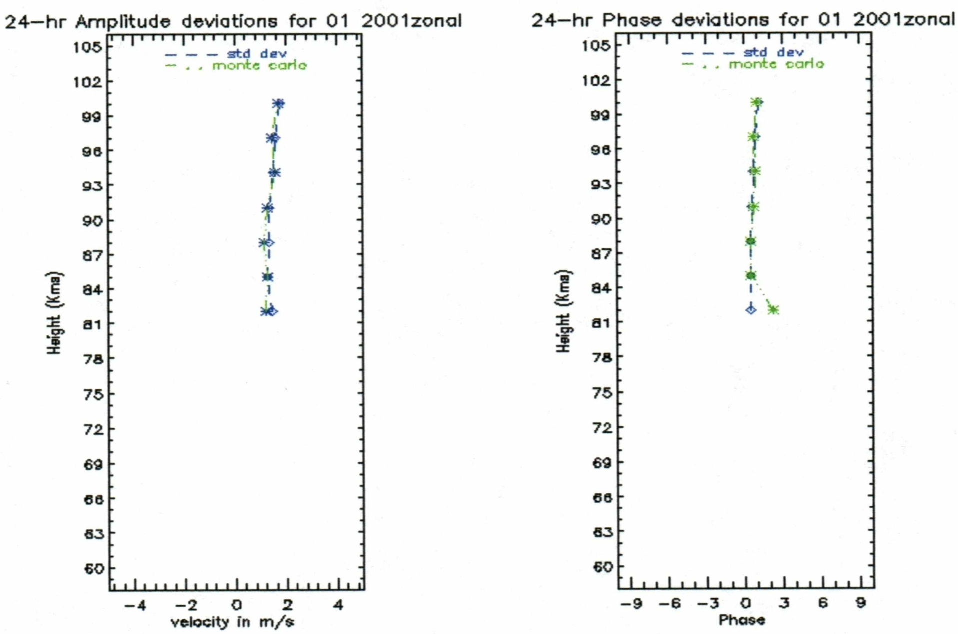



Figure 2-15 The 24-hr amplitude and phase deviations from the LS fit and the Monte Carlo deviation.

Figure 2-16 shows the standard deviation of the estimated semi-diurnal amplitude and phase for January 2001 obtained by the harmonic least squares fit of the original data and by the Monte Carlo bootstrap method. Both methods give an error estimate for the semi-diurnal amplitude of less than 3 m/s with the largest error occurring at 73 km. There is less than 1 m/s difference in the standard deviation estimate of the amplitude between the two estimation methods. Both methods give an error estimate for the semi-diurnal time of maximum of less than 1 hour across most heights. The exception to this is at the lowest two heights which have errors in the time of maximum of less than 3 hours. There is less than 1 hour difference in the standard deviation estimate of the time of maximum between the two estimation methods.

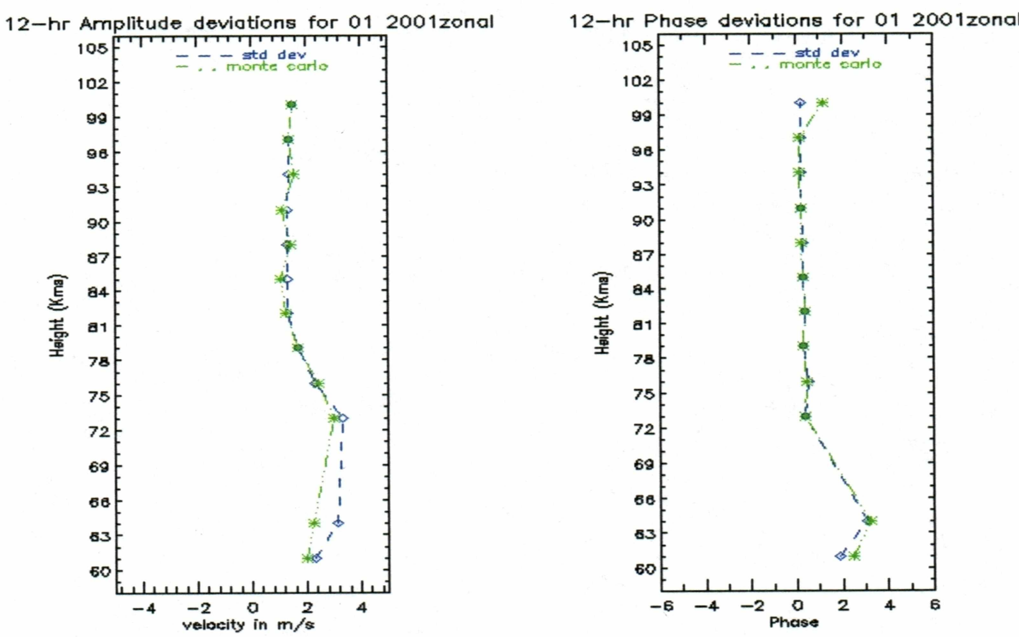



Figure 2-16 The 12-hr amplitude and phase deviations from the LS fit and the Monte Carlo deviation.

## 2.7 Processing Procedure

In this chapter we have examined the processing of our data in the time domain using a harmonic least squares analysis routine. We noted that our raw data are not uniformly sampled on a 5-minute interval due to missing data points. We also noted that the missing data points were randomly distributed across the day at the upper heights but were systematically grouped at night at the lower heights. We could force the data sequence to be more uniformly sampled by averaging and we proposed two averaging methods: hourly average and composite month-day average. These averaging methods alleviated most of the random gaps in data at the upper heights but could do nothing about the systematic nighttime gap in data at the lower heights. Because the lower heights did not have full 24 hour coverage of data we knew that we would have difficulty estimating the diurnal tides at these heights.

We needed an easily applied rejection criterion for determining when a fit was valid. We investigated using the condition number derived from the harmonic least squares fit routine and determined that a condition number less than 10 gave appropriate fits. We determined that it was inappropriate to try and fit a 24 hour wave to a data sequence that had less than 15 hours coverage across the day. Since several of our lower heights have less than 15 hours coverage, a sensible approach would be to fit each period separately, so that estimates of the semi-diurnal tidal parameters would be determined even if the diurnal tidal parameters could not be.

We looked for any advantage that particular methods of pre-processing averaging might have in obtaining valid estimates and determined that there were no advantages for any method. Although all methods provided essentially the same fits, it was noted that the computational time required to perform a fit using the month-day average is considerable less than the computational time required to perform a fit using the 5-min raw data. However, as there are times when one might want to calculate the fits on a shorter time scale than one month, we have left available the option to process the data using the hourly averaged data.

Finally, we clarified our understanding of the standard deviation obtained from the harmonic least squares routine by comparing that standard deviation with an equivalent parameter determined using a Monte Carlo bootstrap method. We note that these two standard deviations are nearly identical. We suspect that the reason for this is that most of the variability in the measured data are due to the geophysical variability rather than measurement errors. Therefore, we conclude that it is appropriate to use the standard deviation obtained from the harmonic least squares routine as our error estimate for the fitted parameters.

# 3 Frequency Domain Analysis of MF Radar Data

In this chapter we will examine two frequency domain analysis techniques; Discrete Fourier Transform (DFT), and Lomb-Scargle, and investigate how well each handles the unevenly sampled nature of our data. As mentioned in Chapter 2, our data set has missing data points that are randomly distributed across the day at the upper heights but are systematically grouped at night at the lower heights. The DFT method assumes uniformly sampled data [*Press et al.*, 1992]. In using this method we must force our data sequence to be uniformly sampled by filling in our missing data points. How to fill in the missing data points is a difficult question in itself and not one that we will address in this thesis. The Lomb-Scargle method was specifically developed to deal with unevenly sampled data.

In Chapter 2 we used a time domain analysis technique that assumes an *a priori* model with specific wave periods (12 hour and 24 hour). Using the spectral techniques in this chapter we do not assume an *a priori* model, but can observe the periods that turn out to be significant. It is hoped then that we can validate the *a priori* choice made in Chapter 2 by examining our data in the frequency domain. One difficulty in using frequency domain techniques is that they are more computationally intensive than the harmonic least squares analysis performed in Chapter 2.

In this chapter we will apply both the DFT and Lomb-Scargle methods to identical synthetic data sequences that have been preconditioned with both random and systematic missing data points. The synthetic data sequences will have the following form:

$$x(t_n) = A\sin\left(\frac{2\pi\, t_n}{T}\right) \tag{3.1}$$

where the sampling $t_n$ is at 1 hour intervals, the amplitude $A$ is 5 and the period $T$ is either 12 hours or 24 hours. The preconditioning will result in irregular sampling of the synthetic data similar to the actual data both in the upper heights (random distribution) and lower heights (systematic distribution). We will examine how each method handles missing data points from both a theoretical and experimental point of view. Here we are primarily interested in retrieving the sinusoidal amplitude $A$.

## 3.1 Discrete Fourier Transform

The basic definition of the DFT is commonly given as:

$$X(f_k) = \sum_{n=0}^{N-1} x(t_n) e^{-j(2\pi/N)nk} \tag{3.2}$$

where, $x(t_n)$ represents our preconditioned synthetic data sequence, and $X(f_k)$ represents the Fourier transform of the original time sequence sampled at $f_k = k/N$ frequencies. Note that $X(f_k)$ as defined in Equation 3.2 is a function of the number of points in the data sequence, $N$. To remove this functionality we normalize the computed DFT by dividing Equation 3.2 by $N$.

As previously discussed, the DFT requires that the data sequence be uniformly sampled. Since we are trying to examine how the DFT deals with missing data points we have preconditioned our synthetic data sequence by setting the missing data points to zero. This artificially assures that we still have a uniformly sampled data sequence to provide to the DFT routine. Unfortunately, this also means that $X(f_k)$ will now be a function of the number valid data points in the preconditioned data sequence. For example, take the DFT of Equation 3.1 for which only $p$ out of $N$ data points are valid and the $N$-$p$ points are set to zero. Using Equation 3.2, the DFT of equation 3.1 becomes:

$$\begin{aligned} X(f_k) &= \frac{1}{N}\sum_{n=0}^{N-1} A_p \sin\left(\frac{2\pi n}{12}\right) e^{\frac{-j2\pi nk}{N}} \\ &= \frac{1}{j2N}\sum_{n=0}^{N-1} A_p \left[ e^{j2\pi n\left(\frac{1}{12}-\frac{k}{N}\right)} - e^{-j2\pi n\left(\frac{1}{12}+\frac{k}{N}\right)} \right] \end{aligned} \tag{3.3}$$

where $A_p = A$ at $p$ out of $N$ points and $A_p = 0$ at $N\text{-}p$ out of $N$ points. By taking the summation in Equation 3.3 we arrive at the following result:

$$X(f_k)=\begin{cases} \pm\dfrac{pA}{j2N} & \text{for } \dfrac{1}{12}=\pm\dfrac{k}{N} \\ \varepsilon_p & \text{for } \dfrac{1}{12}\neq\pm\dfrac{k}{N} \end{cases} \tag{3.4}$$

In this result we have assumed that the $N\text{-}p$ missing data points are randomly distributed. As expected, $X(f_k)$ satisfies $X(f_k) = X(-f_k)^*$ since the original time domain data sequence is real. The residual term, $\varepsilon_p$, represents the inability of the DFT formulation to fully cancel off frequency terms without the full $N$ data points. Note that $\varepsilon_p$ becomes larger as $p$ becomes smaller and is identically equal to zero when $p=N$ (i.e. no data missing).

Figure 3-1 shows the DFT of our preconditioned synthetic data. The left column represents the synthetic data with a single 12 hour sinusoid and the right column a single 24 hour sinusoid. We examine three cases where $p=N$ (top), $p=0.8\,N$ (middle) and $p= 0.3\,N$ (bottom). Since we are interested in our ability to retrieve the amplitude $A$, we have chosen to present only the absolute value of the positive frequencies. This choice necessarily means that the magnitude of any peaks observed will be at most $A/2$. The solid (black) line graphically displays Equation 3.4 for each of our six cases. For each of the six cases the desired harmonic (12 hour on the left and 24 hour on the right) is faithfully captured. As expected the magnitude of the peak becomes smaller in direct proportion to the number of valid data points ($p$). When $p=N$, the magnitude of the peak is 2.5 or $A/2$. When $p=0.3\,N$, the magnitude of the peak is 0.375 or 30% of 2.5. One can also see that the magnitude of the off frequency samples becomes larger as $p$ becomes smaller. The dashed (green) line shows the re-normalization of Equation 3.4 by $N/p$. Notice that now each of the six cases shows a magnitude of the peak of 2.5 or $A/2$. As one would expect the off frequency samples are now also larger by a corresponding amount. It is interesting to see that even when only 30% of the original synthetic data sequence is presented to the DFT the desired harmonic is faithfully reproduced so long as the 70% missing data points are randomly distributed.

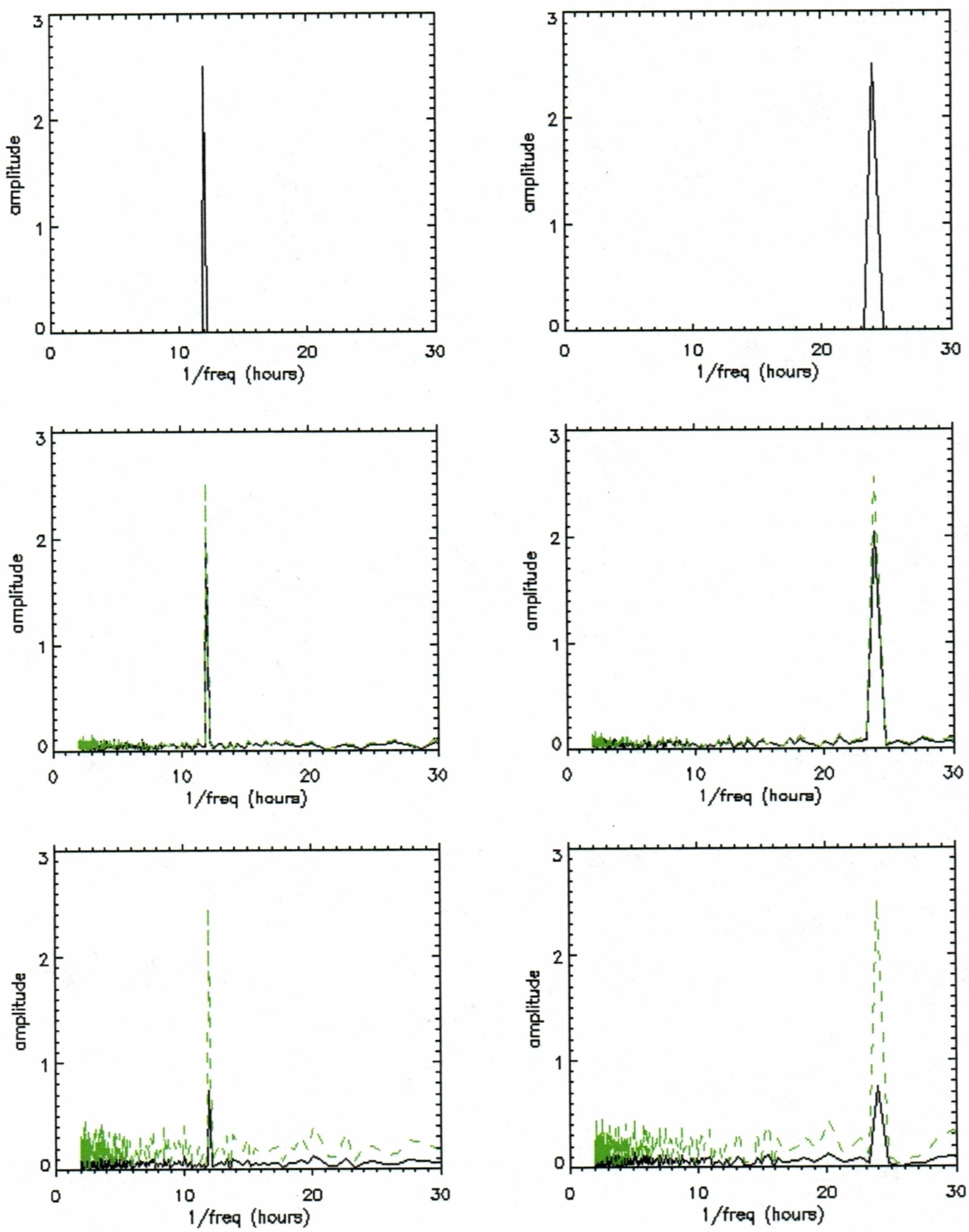



Figure 3-1 DFT for a 12 hour sinusoid (left column) and a 24 hour sinusoid (right column) with 100% (top), 80% (middle) and 30% (bottom) of data randomly selected.

In the previous example we only considered missing data points that were randomly distributed. This is a good model for heights above 85 km, but not for lower heights. At lower heights our data are present only during daytime and not present at night. This situation is equivalent to windowing our data with a pulse train that has a period of 24 hours and a pulse width equal to the duration of daylight hours. Equation 3.3 now becomes

$$X(f_k) = \frac{1}{N}\sum_{n=0}^{N-1} A\sin\left(\frac{2\pi n}{12}\right)\sum_{m=0}^{N/T-1}\Pi\left(\frac{n-12-mT}{L}\right)e^{-j\frac{2\pi nk}{N}} \tag{3.5}$$

where Π is a rectangular pulse centered at $n=12+mT$ with width $L$. Equation 3.5 shows the DFT of the multiplication of two time series, which is just the circular convolution in the frequency domain of the DFT of each individual time series. The DFT of the sinusoid is given by Equation 3.4 with $p=N$. The DFT of the pulse train is given by

$$X_{\Pi}(f_k) = \begin{cases} \frac{1}{T}e^{-j\pi k(L-1/N)}\dfrac{\sin\left(\frac{\pi kL}{N}\right)}{\sin\left(\frac{\pi k}{N}\right)} & k=\frac{mN}{T}, m=0,1,2,\mathrm{N/T}-1 \\ 0 & \text{else} \end{cases} \tag{3.6}$$

Equation 3.6 shows that the DFT of a pulse train is a series of impulses with a sinc envelop. This means that our original sinusoid when circularly convolved with this DFT will have its energy spread into frequencies other than its own.

Figure 3-2 shows the DFT of our synthetic data widowed with a pulse train with period $T$ = 24 hours and pulse width $L$ = 24 (top), $L$=16 (middle), and $L$=8 (bottom). The left column represents the synthetic data with a single 12 hour sinusoid and the right column a single 24 hour sinusoid. Again we present only the absolute value of the

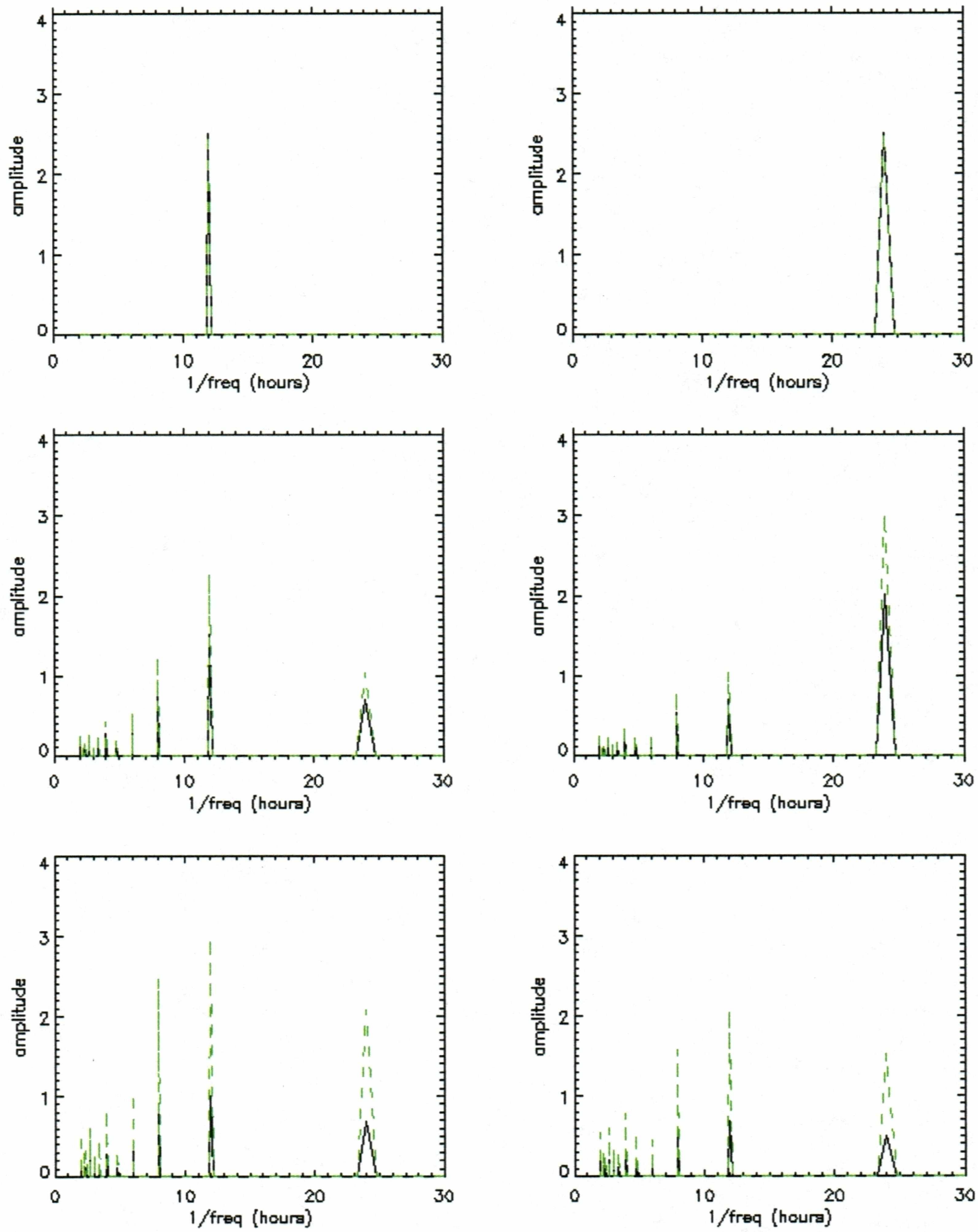



Figure 3-2 DFT for a 12 hour sinusoid (left column) and a 24 hour sinusoid (right column) with 24 hours of data (top), 16 hours of data (middle) and 8 hours of data (bottom) periodically selected data.

positive frequencies. The solid (black) line graphically displays the circular convolution of Equation 3.4 with Equation 3.6 for each of our six cases. For each of the six cases the desired harmonic (12 hour on the left and 24 hour on the right) can be observed. Additionally, undesired harmonics are also observed and can be comparable in amplitude to the desired harmonic. For instance in the bottom right panel the 12 hour peak is greater than the desired 24 hour peak when only 8 hours of data per day are available. As expected, 8 hours of data per day are not sufficient to capture a 24 hour harmonic. Even for the 12 hour sinusoid, 8 hours of data give rise to significant peaks at 8 hours and 24 hours. The dashed (green) line shows the solid (black) line re-normalized by T/L. Note that the peaks of the desired harmonic do not necessarily reflect the amplitude of our original sinusoid. Since the DFT of the pulse train, Equation (3.6), is infinite in extent it will alias into our desired window and either add constructively or destructively dependent on the phase of the interfering frequency.

### 3.2 Lomb-Scargle Periodogram

An alternative to the DFT determination of PSD is Lomb-Scargle periodogram method [*Lomb*, 1976, *Scargle*, 1982]. This method for PSD estimation was developed specifically to deal with the problem of unevenly sampled data. The Lomb method is similar to the DFT except that this method evaluates the fit only at times $t_i$ that are actually measured and not on an evenly sampled grid.

The Lomb normalized periodogram is defined by

$$P_N(\omega) \equiv \frac{1}{2}\left\{ \frac{\left[\sum_j (h_j - \bar{h})\cos\omega(t_j - \tau)\right]^2}{\sum_j \cos^2 \omega(t_j - \tau)} + \frac{\left[\sum_j (h_j - \bar{h})\sin\omega(t_j - \tau)\right]^2}{\sum_j \sin^2 \omega(t_j - \tau)} \right\} \tag{3.7}$$

and $\tau$ is defined by the relation

$$\tan(2\omega\tau) = \frac{\sum_j \sin 2\omega t_j}{\sum_j \cos 2\omega t_j}. \tag{3.8}$$

The constant $\tau$ is a kind of offset that makes $P_N(\omega)$ completely independent of shifting all the $t_i$'s by any constant. It was shown by Lomb that this particular choice of offset has another, deeper effect: it makes the equation of $P_N(\omega)$ identical to the equation that one would obtain if one estimated the harmonic content of a data set, at a given frequency $\omega$, by linear least-squares fitting to the model

$$h(t) = A\cos\omega t + B\sin\omega t \tag{3.9}$$

Effectively the Lomb-Scargle method performs a least squares fit of the data to multiple sinusoids all at once.

Figure 3-3 shows the Lomb-Scargle periodogram of our synthetic data that have been preconditioned, as before, with various percentages of randomly missing data. The left column represents the synthetic data with a single 12 hour sinusoid and the right column a single 24 hour sinusoid. For each of the six cases the desired harmonic (12 hour on the left and 24 hour on the right) is faithfully captured. For the Lomb-Scargle method no additional renormalization is required to account for the different percentages of missing data. As with the DFT, the noise floor increases when a larger percentage of data are missing. Other than not having to track the number of missing data points there appears to be no difference between the DFT and the Lomb-Scargle methods for randomly missing data points.

Figure 3-4 shows the Lomb-Scargle of our synthetic data widowed with a pulse train with period $T$ = 24 hours and pulse width $L$ = 24 (top), $L$=16 (middle), and $L$=8 (bottom). The left column represents the synthetic data with a single 12 hour sinusoid and the right column a single 24 hour sinusoid. For each of the six cases the desired harmonic (12 hour on the left and 24 hour on the right) can be observed. Additionally, as

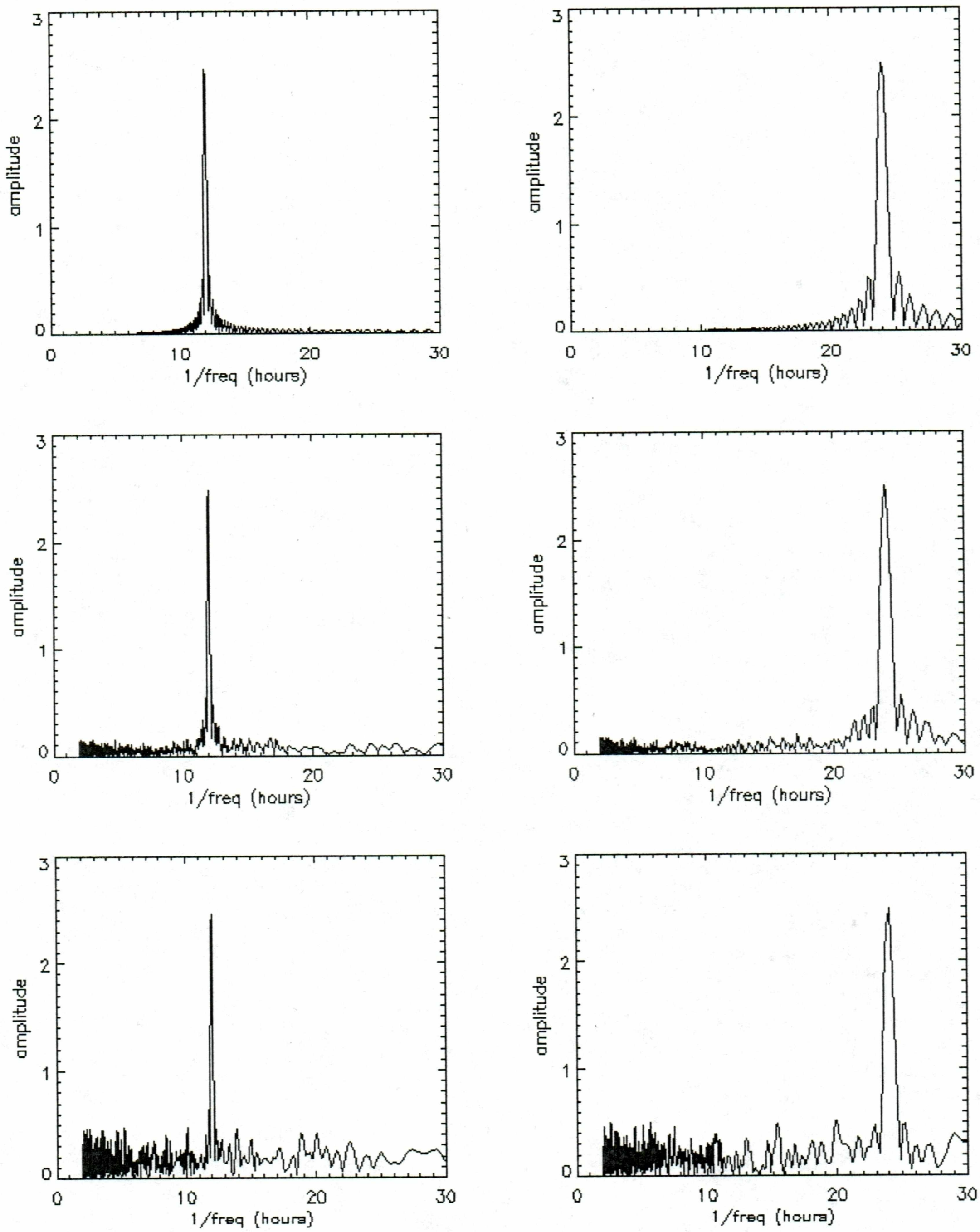



Figure 3-3 Lomb-Scargle periodogram for a 12 hour sinusoid (left column) and a 24 hour sinusoid (right column) with 100% (top), 80% (middle) and 30% (bottom) of data randomly selected.

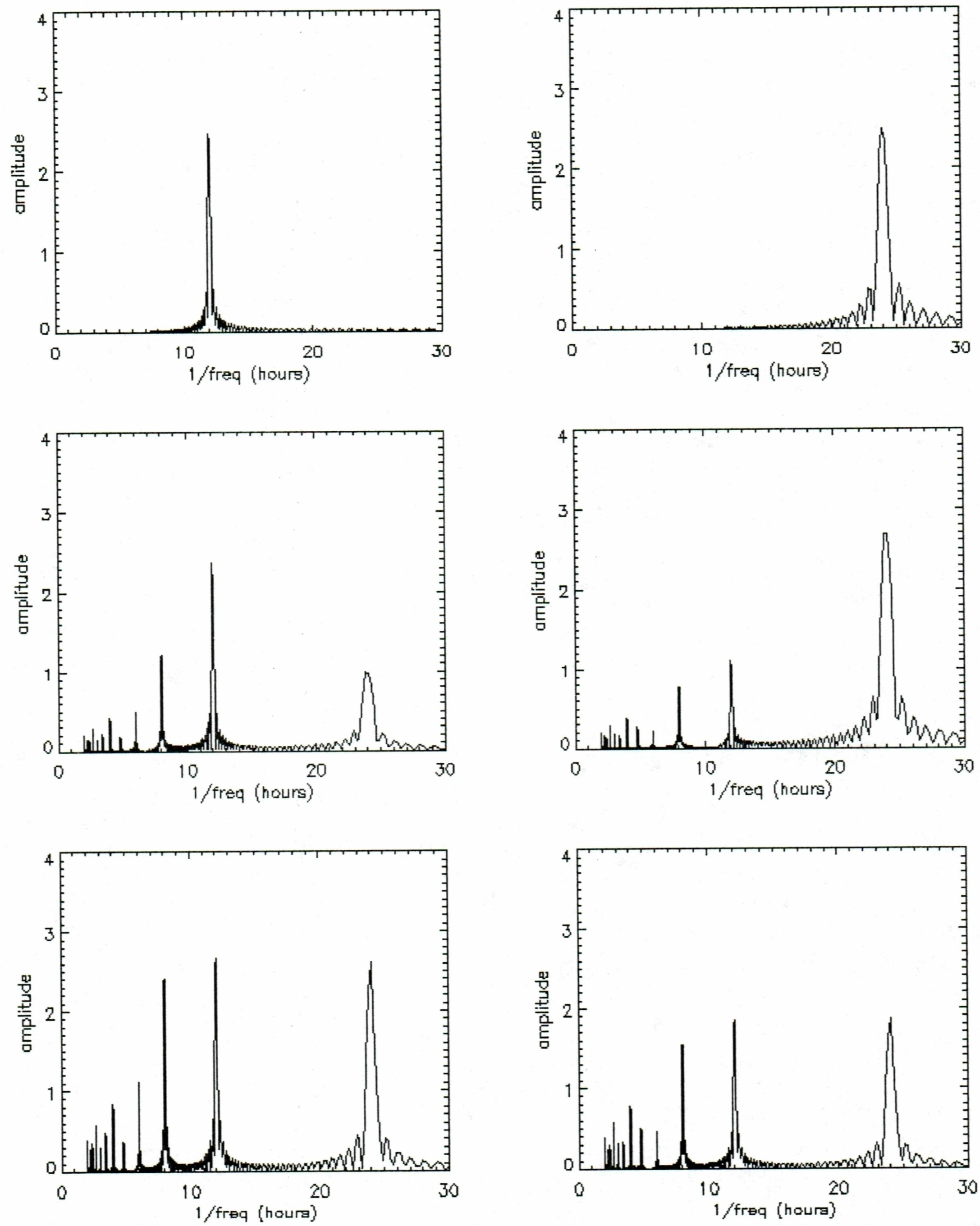



Figure 3-4 Lomb-Scargle periodogram for a 12 hour sinusoid (left column) and a 24 hour sinusoid (right column) with 24 hours of data (top), 16 hours of data (middle) and 8 hours of data (bottom) periodically selected.

was the case for the DFT analysis, undesired harmonics are also observed and can be comparable in amplitude to the desired harmonic. One benefit of the Lomb-Scargle over the DFT is that the amplitude of the desired harmonic is preserved for all cases except the case of the 24 hour harmonic windowed with a pulse of 8 hours. Again as expected, 8 hours of data out of 24 is not enough to faithfully retrieve the 24 hour harmonic. The relative sizes of the aliased harmonic and desired harmonic appear to be comparable to the DFT method.

Figure 3-5 shows the Lomb-Scargle periodogram of a synthetic data sequence containing both a 12 hour and 24 hour harmonic. These synthetic data were pre-conditioned by removing those data which correspond to actual missing data for the month of January 2001. The Lomb-Scargle method was applied to each height separately and then the individual results were combined to create the three dimensional plot of

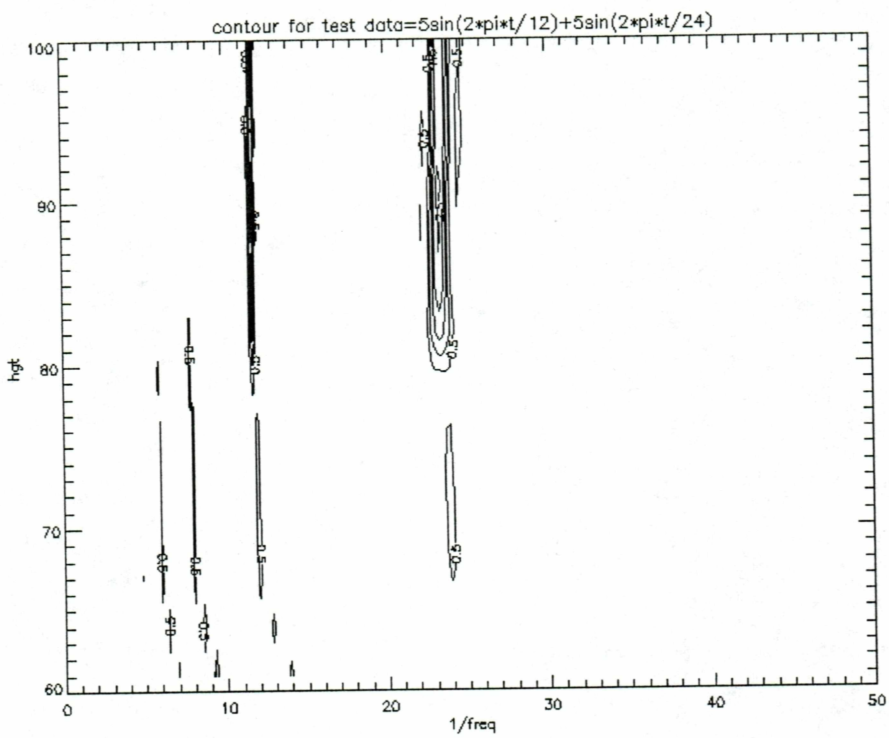



Figure 3-5 Contour of the Lomb-Scargle periodogram for our synthetic data, containing 12 hour and 24 hour harmonics, windowed with January 2001 gaps.

spectral amplitude versus height and period. One can see in this plot that the desired harmonics are faithfully reproduced throughout most of the height range where we typically acquire data. The only deviation is at the lower heights where the fitted harmonic appears to tend towards longer periods than the defined periods of the synthetic data. More investigation is required to understand this shift more fully.

Figure 3-6 and Figure 3-7 show the Lomb-Scargle periodograms for January, 2001, and July, 2001, data respectively. As with the synthetic data the Lomb-Scargle method was applied to each height separately and then the individual results were combined to create the three dimensional plot of spectral amplitude versus height and period. What is immediately obvious is that our actual data are considerably more spectrally diverse than our two harmonic synthetic data. January specifically has significant energy in a broad range of frequencies. One reason that January seems to be more spectrally diverse may be because, in this particular January, at the lower heights we have considerable periods of missing data. In effect what we may be observing is noise. More investigation is required to fully determine the cause of this difference. In January, 2001, you can still see the 12 hour and 24 hour tidal components. Also evident are components at other periods even at the upper heights. July, interestingly, does not show a strong harmonic at 12 hours. (Note that this is also seen in Figure 2-13). In fact based on this plot we might want to fit to a 10 hour harmonic instead.

The net result of our comparisons is that both the DFT and Lomb-Scargle methods can retrieve spectral amplitudes effectively if there are enough data and if the missing data points are randomly distributed. Possibly, the Lomb-Scargle has some advantage at the lower heights, where the missing data points are more systematically distributed. However, this is not a large advantage. Finally, we note that determining the dominant spectral components prior to any least squares fitting is potentially beneficial, so that we are not fitting to a harmonic that does not really exist.

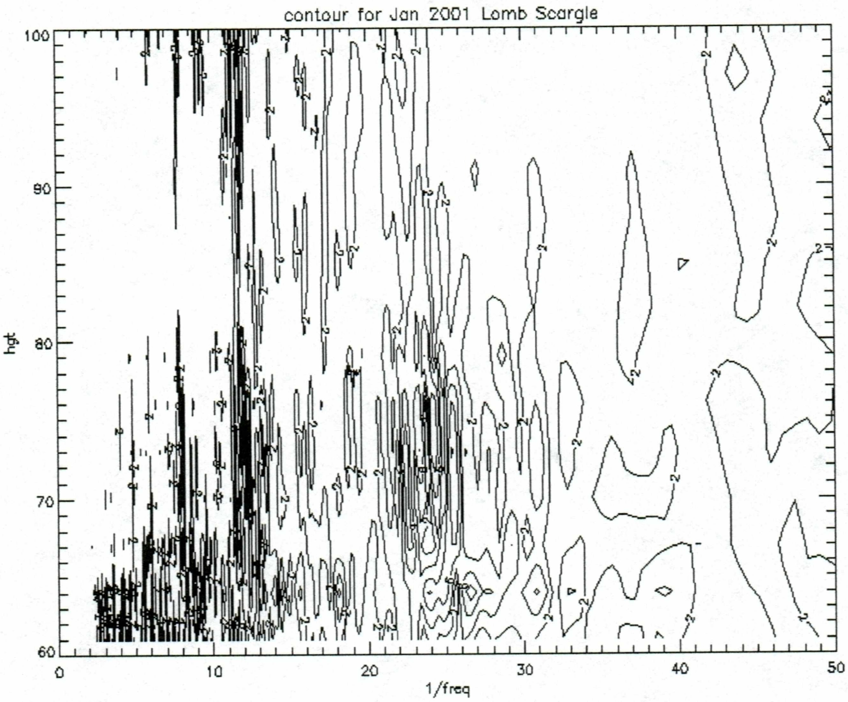



Figure 3-6 Lomb-Scargle periodogram of the January 2001 data.

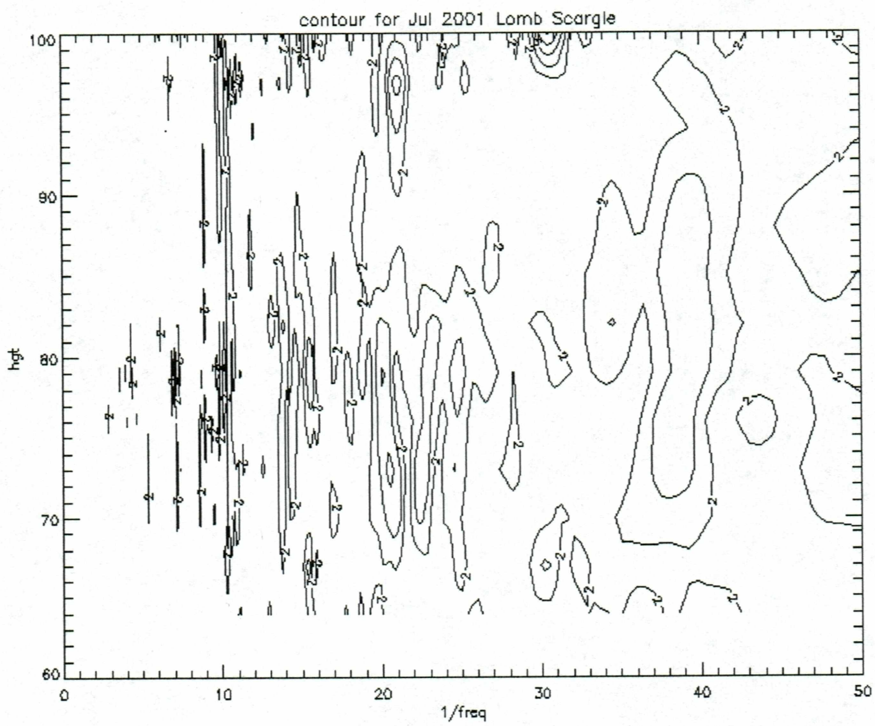



Figure 3-7 Lomb-Scargle periodogram of the July 2001 data.

# 4 Results of Data Analysis

In this chapter, the results of applying the time domain analysis using the automated processing system to the wind velocity data from 61 to 100 km height collected at Platteville, Colorado is presented and discussed. Data collected during the years 2000-2004 have been analyzed. These data are also compared with the GSWM-02 model data and data from other MF stations. Following is a discussion of the automated processing system and the results of that analysis.

In Chapter 2 we investigated problems associated with non-uniformly sampled data and determined that generating a month-day average provided us with robust results. Even in the month-day averages we do not have data for the entire 24 hour duration of the day at lower heights. Therefore we have decided to obtain a fit for the 24 hour period only when there are a minimum of 15 hours of data, and to obtain a fit for the 12 hour period only if there are at least 8 hours of data in a day. After examining the data using the LS fit method and the condition number variability depending on the amount of data present, we came to a conclusion of using the condition number as the criterion to decide whether the obtained tidal parameters are valid or not. We decided on using a condition number less than 10 as one of the criteria to assess if the obtained tidal parameters are valid or not.

In Chapter 3 we investigated the spectral methods to see if our assumed periods of 12 and 24 hours are accurate. For this we used the Lomb-Scargle Periodogram method to obtain the spectral amplitudes. Even though we were not able to exactly show the spectral peaks at the 24 hour and 12 hour periods, in most of the cases these were dominant. The FFT method also gave us similar results, but Lomb-Scargle seems to have some advantage in the lower heights, since we have more systematically distributed data at the lower heights, even though this advantage is not very great. We created a synthetic data set and looked at it using both the methods and various cases. From this we were able to say that our data consisted of diurnal and semidiurnal tidal components.

### 4.1 Automated Processing System

Figure 4-1 shows the flow of the basic automated data processing system. The goal is to acquire monthly mean winds and tidal amplitudes to compare with other MF stations and with atmospheric models. The first two blocks read in a month's worth of 5 minute velocity estimates and apply preliminary data rejection criteria, such as a minimum signal to noise ratio (SNR) and a range of heights of interest.

As we mentioned this is an automated processing system that can be used to generate the tidal parameters of the velocity wind estimates 24 hours a day and 7 days a week. The 5-minute velocity estimates are averaged into hourly bins. These hourly binned data are written onto hourly average data files. At the same time histogram plots are generated which are discussed in the next page. If we wish to generate or run the automated processing system for only a specific month, we can do that by entering the year and month for which data are desired. This system also looks out for any leap year so that the hourly averages generated are not corrupt.

For generating month-day averages a similar routine is used which averages data at each hour across the entire month. These month-day averages are also written into data files and saved for future use. Depending on our needs we can either use the hourly average data or the month-day data. For example if we want to compare our data with data from some other instrument or location, which is available only hourly, then we can use the hourly average data. If the data are available on a monthly basis, we have the month-day data for comparison. These hourly or month-day data are then passed to the tidal fitting routine to obtain the necessary tidal parameters for further study.

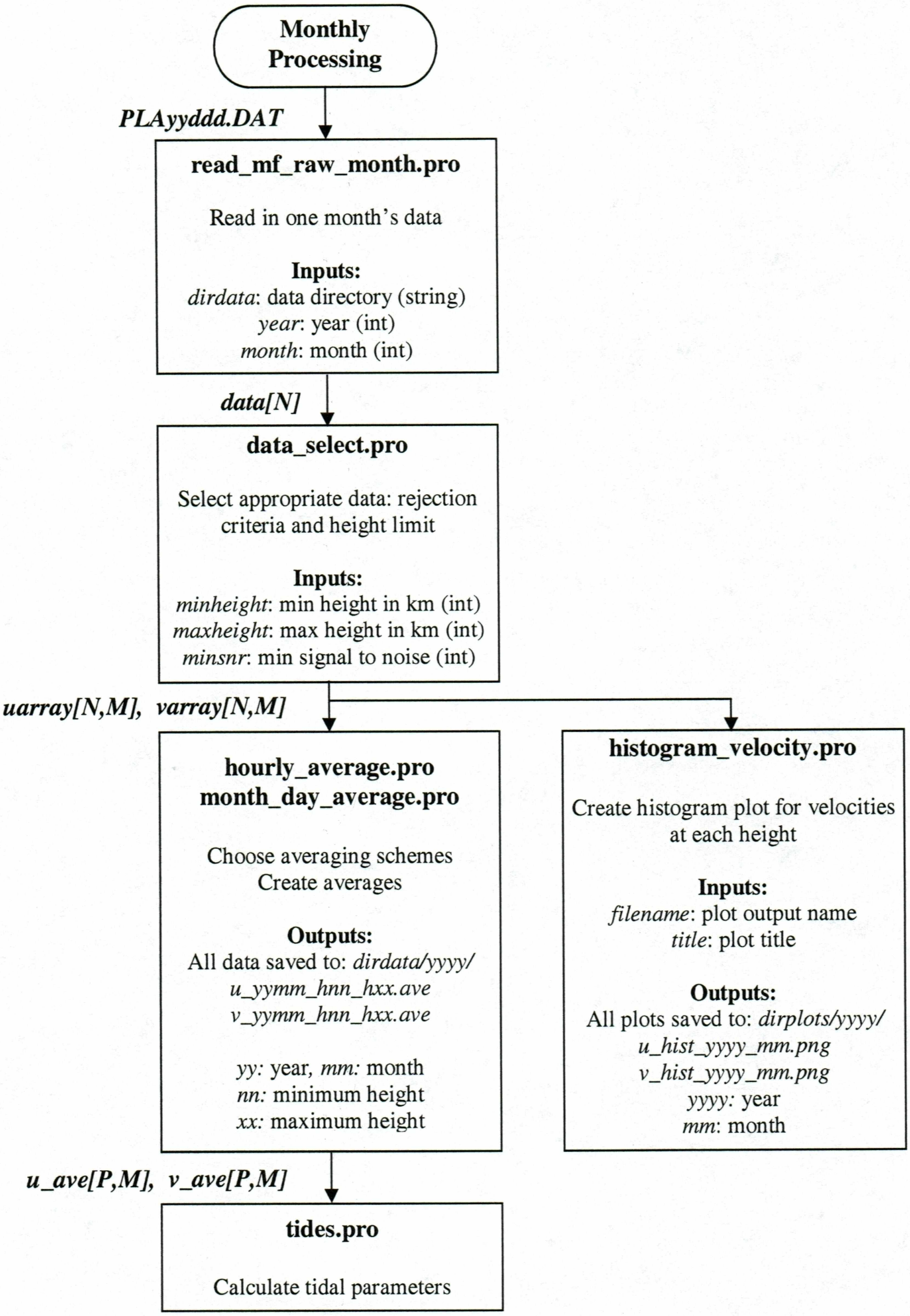



Figure 4-1 Basic data processing flow.

Diagnostic plots are created at various places in the processing stream. For example we plot histograms for each height of velocity estimates that have passed through the data selection routine. These histogram plots give us a preliminary view of the spread of velocity estimates at each height and the number of valid estimates that will be passed to the averaging routines. Figure 4-2 shows the histogram plot of zonal wind estimates for January 2001. In the right panel we see the number of velocity estimates for each height. As seen in Chapter 2 there are fewer velocity estimates at lower heights with the maximum number of velocity estimates being acquired at around 85 km. We can clearly see the lower level jet in the mean, peaking at around 73 km. Note also that the distribution of wind estimates is clearly not Gaussian. Frequently the distribution is significantly skewed and at the upper heights it is clearly bimodal.

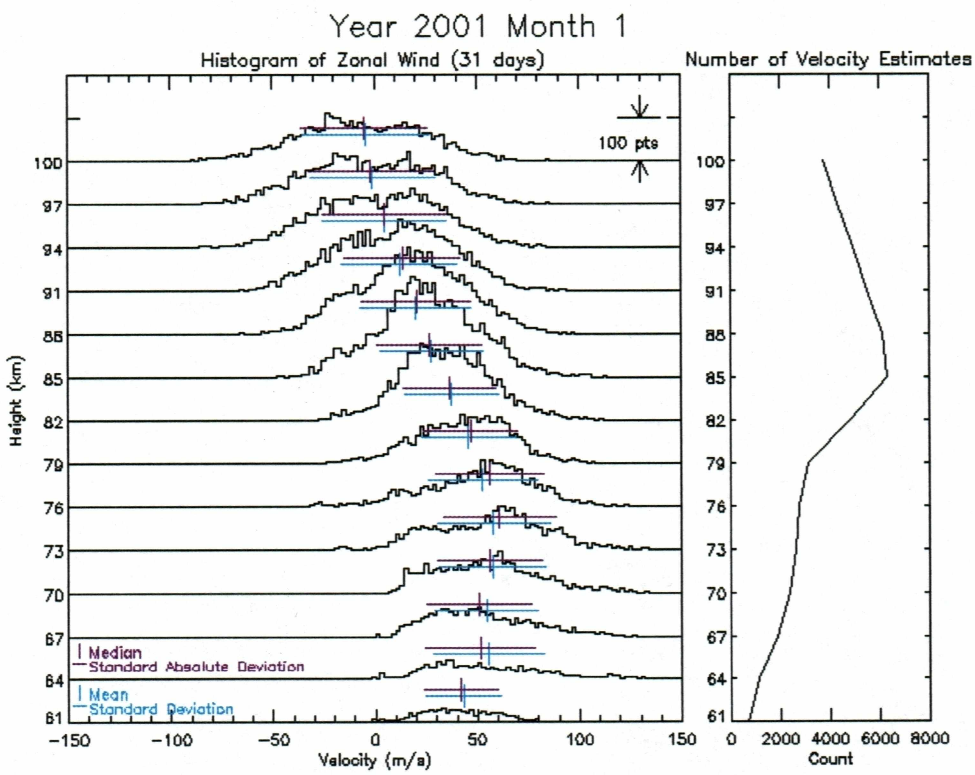



Figure 4-2 Histogram of zonal wind for January 2001.

After the initial data selection and averaging, the data are saved to a file for future processing and/or can be sent directly to the tidal analysis routine. Figure 4-3 shows the basic processing flow for our tidal analysis. We currently specify three regimes for calculating tidal fits depending on the coverage of data across a day. If there is less than 8 hours coverage we only calculate the mean wind at that height. If there is between 8 and 15 hours coverage then we calculate the mean and semidiurnal components of the wind. If there is greater than 15 hours coverage we calculate the mean, semidiurnal, and diurnal components of the wind. The data are further checked after the least squares fitting routine to verify that the condition number is less than 10.

The crux of the tidal analysis software is a routine *harmonic_ls_fit.pro* [Palo, 1997]. The user provides a matrix specifying the periods, $T_0 \ldots T_N$, of the sinusoids to which the data should be fit. This routine computes a harmonic least squares fit to the model $y(t) = DC + A\cos(2\pi w_i(t - tmax))$, where $w_i = 2\pi/T_i$. The routine returns a 4 by k matrix of the resulting fitted parameters and their corresponding errors. There are k=T+1 rows, where T is the number of requested sinusoids. The first row contains the DC fit. Each row contains the fit and standard deviation of the wave amplitude and phase (4 elements) for that particular sinusoid. The harmonic least squares analysis routine also returns a reconstructed version of the measured data based on the fitted parameters. This reconstructed data are passed to a plotting routine to provide diagnostic plots such as those shown in Figure 2-5 and Figure 2-6.

The fitted parameters are written to data files separately for the diurnal and semidiurnal tides. The routines used to obtain the above process are shown in Appendix A.

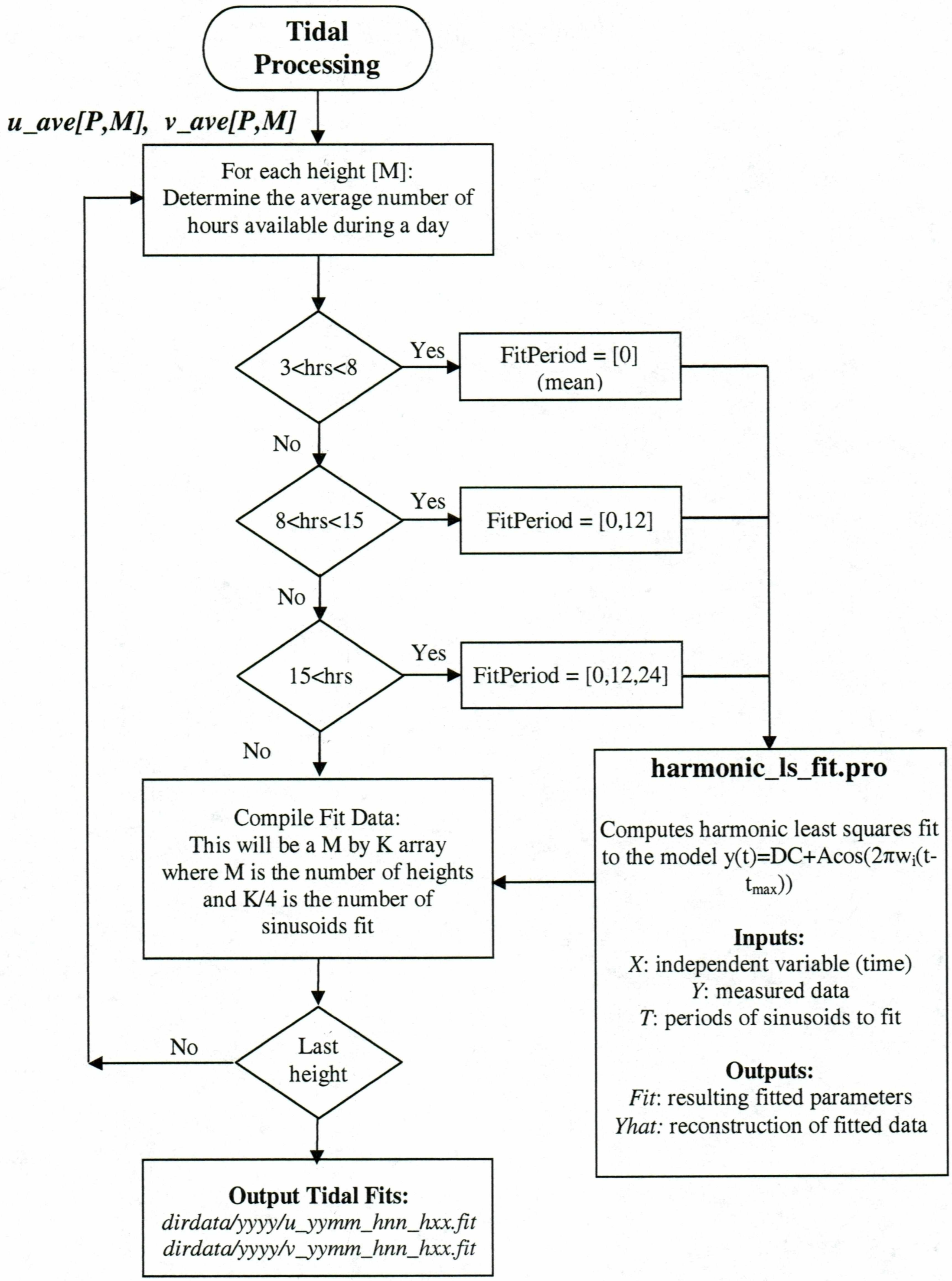



Figure 4-3 Tidal processing flow.

## 4.2 Mean Winds

The mean winds are obtained using the harmonic least squares routine fitted to a 12-hour period. If the number of hours is less than 8-hours at a particular height, then the mean wind velocity is obtained by calculating the average of the data. The mean is calculated only when there are at least a minimum of three data points available.

Figure 4-4 shows contour plots of the monthly mean zonal and meridional wind velocities for 2000-2004. The *x*-axis represents the month of the year and the *y*-axis represents the height in kilometers. For comparison we also show a contour plot of the zonal wind for one year derived from the GSWM-02 data. The red color represents positive (eastward/northward) winds and the blue color represents negative (westward/southward) winds. The contour lines are at 20 m/s increments.

For the months of January to March the zonal velocity is more than 30 m/s at lower heights and as we go up in height the velocity decreases. For the months from May to August the winds have negative values at lower heights of around 61 km. The velocities increase negatively to -40 m/s up to a height of 75 km and then the wind jet increases positively for the heights above 75 km. The zonal mean wind velocity is similar to that of the GSWM mean wind data. These trends are also seen at Saskatoon [*Manson et al*. 1985] and Urbana [*Franke and Thorsen*, 1993].

The meridional mean wind velocity is near 0 m/s, except that during August 2000, at a height of 85 km, the wind velocity went to 35 m/s. Also the velocity for all the years across all the months is consistent. Even at Saskatoon [*Manson et al*. 1985] the mean winds varied from -10 m/s to 10 m/s and during the months of October to December the velocities were around 5 m/s, similar to our velocities. At Urbana [*Franke and Thorsen*, 1993] the mean wind at heights of around 84 km was -10 m/s southward in June 1992.

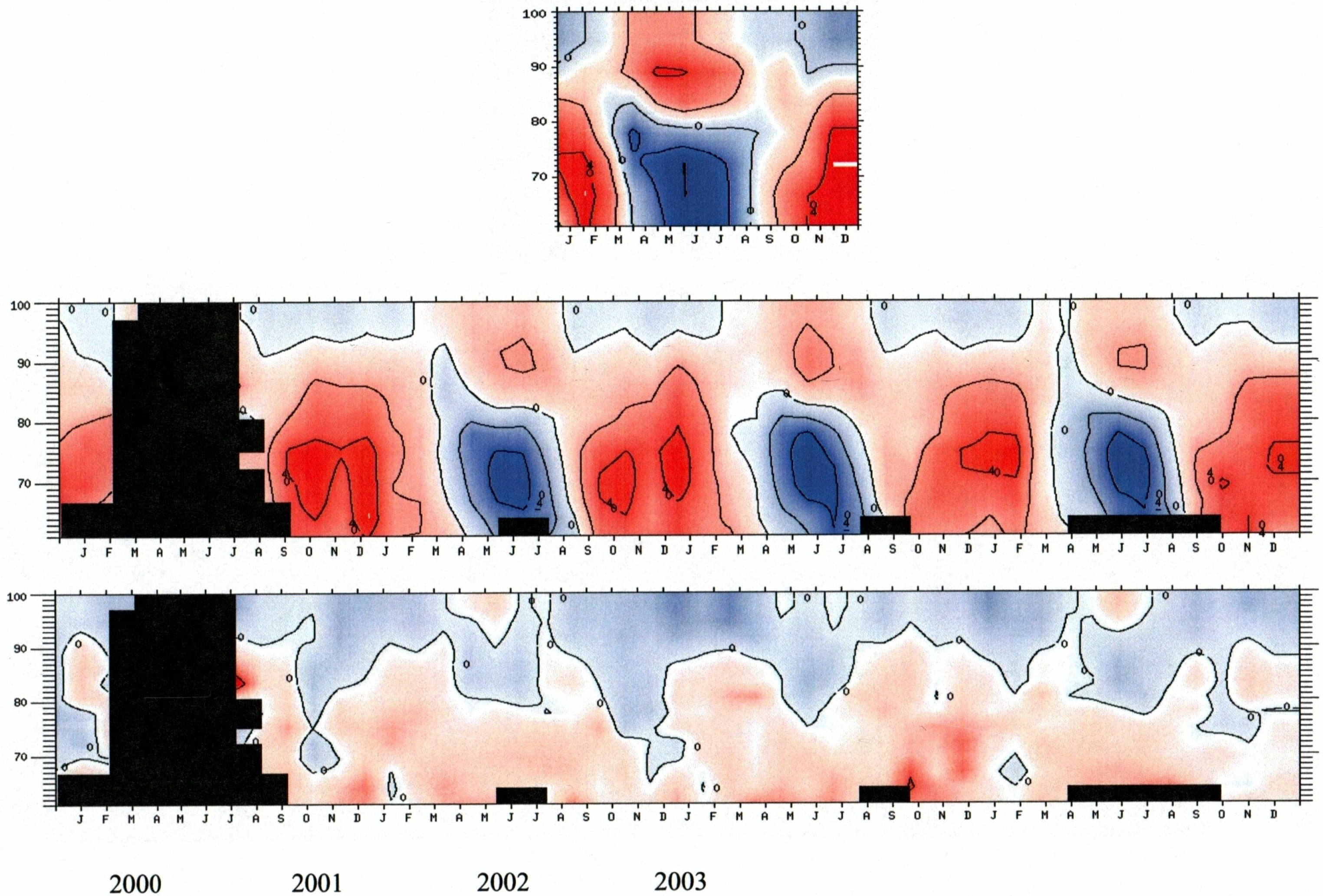



Figure 4-4 Mean wind speed contours for the year 2000-2004, GSWM mean wind for one year (top), zonal wind (middle), meridional wind (bottom).

## 4.3 Tidal Components

The tidal amplitudes and phases of diurnal and semidiurnal tides have been obtained using the harmonic least squares fitting method. As mentioned in Section 4.1 these parameters have been obtained using the criteria of the minimum number of hours for which the data are available and also the condition number criterion. The tidal parameters are written into data files as well as plotted to observe the characteristics of these parameters and compare them with other data. These tidal components are compared with MF data of other stations as well as GSWM-02 [*Hagan*, 2004] model data.

### 4.3.1 Diurnal Tide

The diurnal amplitudes and phases for the years 2000-'04 are shown in Figure 4-5 for the zonal winds and in Figure 4-6 for the meridional winds. The top panel shows the velocity and the lower panel shows the phase for each month. The diurnal amplitudes and phases are obtained using the condition of a minimum of 15 hours of data availability and the criterion of condition number less than 10. Hence we cannot see these parameters for all the heights since at lower heights we don't have enough data to obtain the fit parameters. The data are available below 79 km only for few months. The diurnal amplitude is never greater than 20 m/s except at the height of 100 km in June and July. For each month the amplitude and phase for all the years are almost the same except for the month of March, where there is a drastic difference for each year. This might be because of the season changing. Even in the month of August the amplitudes differ in each year at heights of around 82 km and above 97 km. The amplitude during the winter months is low and never greater than 12 m/s, whereas it went to nearly 20 m/s during the summer months. Above 84 km diurnal amplitude mostly decreases (except for the months May to August), whereas the GSWM diurnal amplitude increases. However, for most of the months, below 84 km if we have diurnal amplitude then it is similar to the GSWM data with a difference in the magnitudes. The diurnal amplitude for the summer months of April to August is increasing as we go up in altitude above 82 km. The vertical pattern of the diurnal amplitude at Saskatoon [*Manson and Meek*, 1986]. For the winter months the diurnal amplitude above 82 km shows a decreasing trend. The diurnal phase is mostly in agreement with the GSWM phase with similar structure. Except for few months, like May, the phase looks quite consistent. At the height of 100 km the diurnal amplitude is nearly 20 m/s for our data, whereas for the Urbana data it is around 8 m/s.

The 24-hour meridional wind amplitude is never greater than 20 m/s, except in March 2002, where the amplitude is 32 m/s at a height of 88 km. During March 2002 the velocity at mid heights of 82 km to 91 km is very large, around 30 m/s, and is never that high for any of the other year's data. This may be due to the season change effects during that particular year. The meridional diurnal amplitude does not agree well with that of the GSWM data. The amplitude at heights above 85 km is less than that of GSWM data. In the month of December the data for each year show a difference in velocity of around 3 to 4 m/s. At Urbana [*Franke and Thorsen*, 1993] during March 1992 the amplitude is around 10 m/s at a height of 88 km. The meridional diurnal amplitude at Urbana at a height of 93 km is around 10 m/s for April. The Platteville data is also around 10 m/s. The phase indicates a propagating diurnal wave. The phase of our data is mostly in agreement with the GSWM phase for almost all the years. The phase in October and November appears differently for all the years of 2000-2004. For Urbana during March 1992 the phase is -2 hours at a height of 90 km, whereas it is 6 hours for the Platteville data. Similarly for our data during March, the phase varied from 6 to 9 hours over different years. Also we can clearly see the diurnal nature of the data in the phase plots.

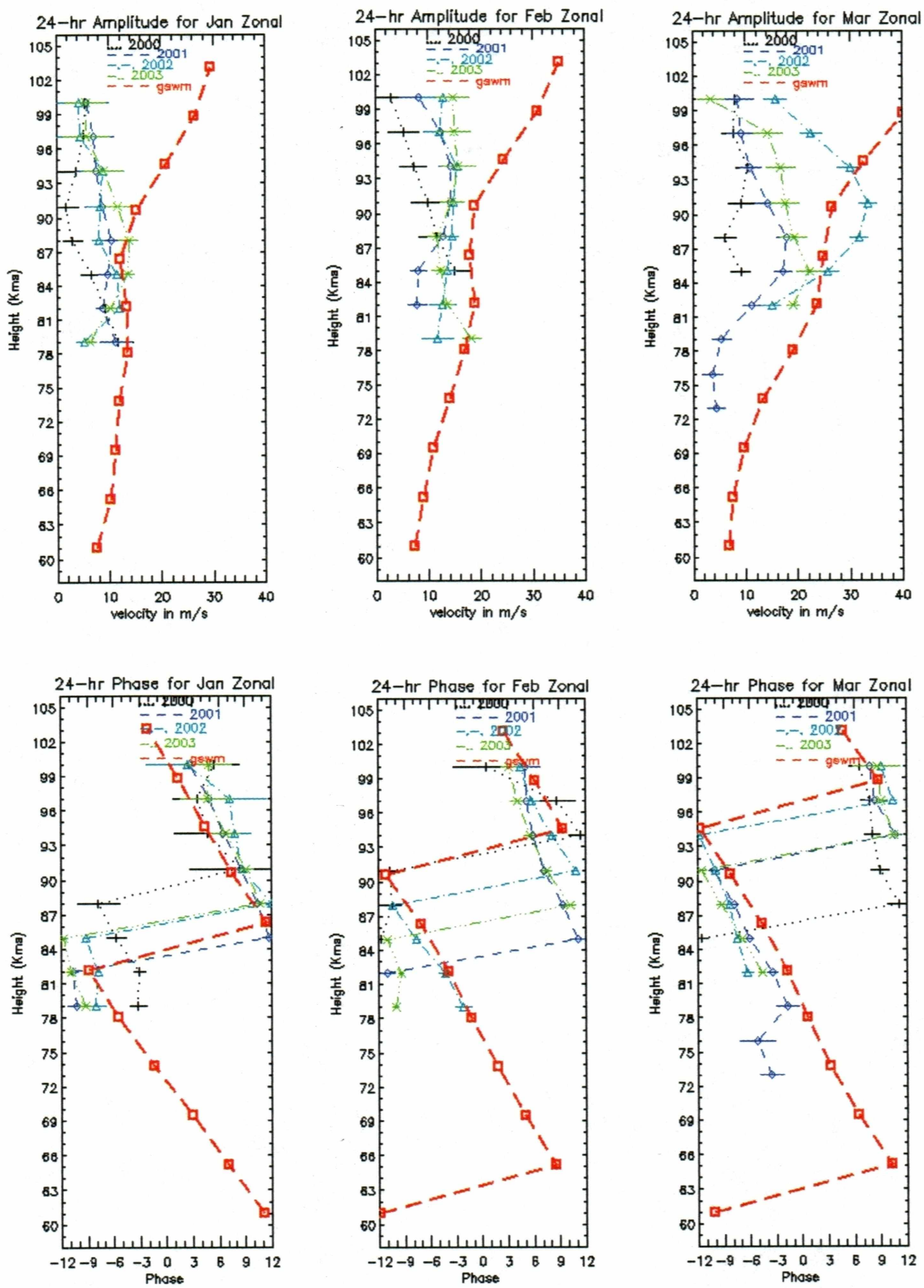



Figure 4-5 Zonal diurnal amplitudes and phases.

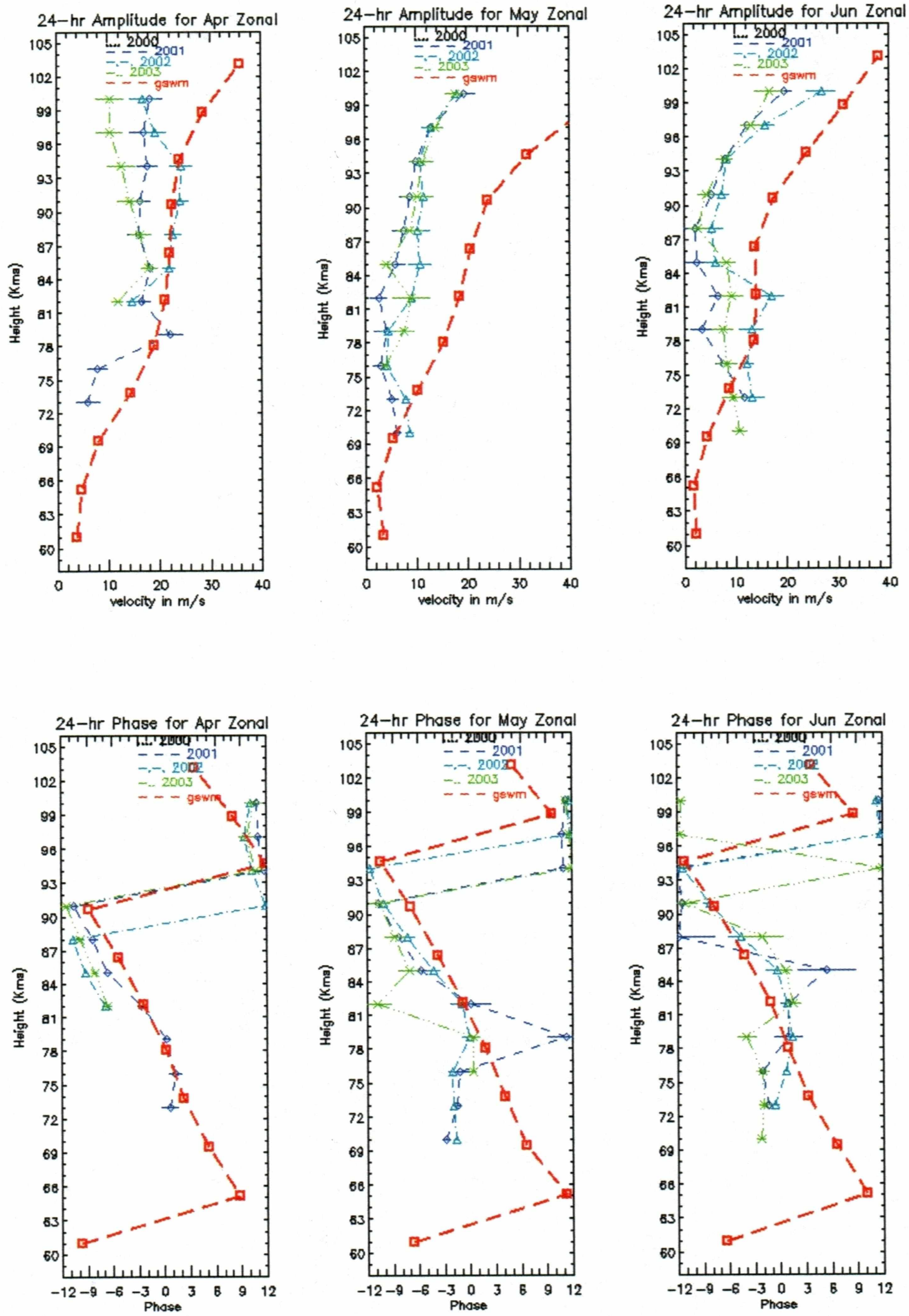



Figure 4-5 Zonal diurnal amplitudes and phases (continue).

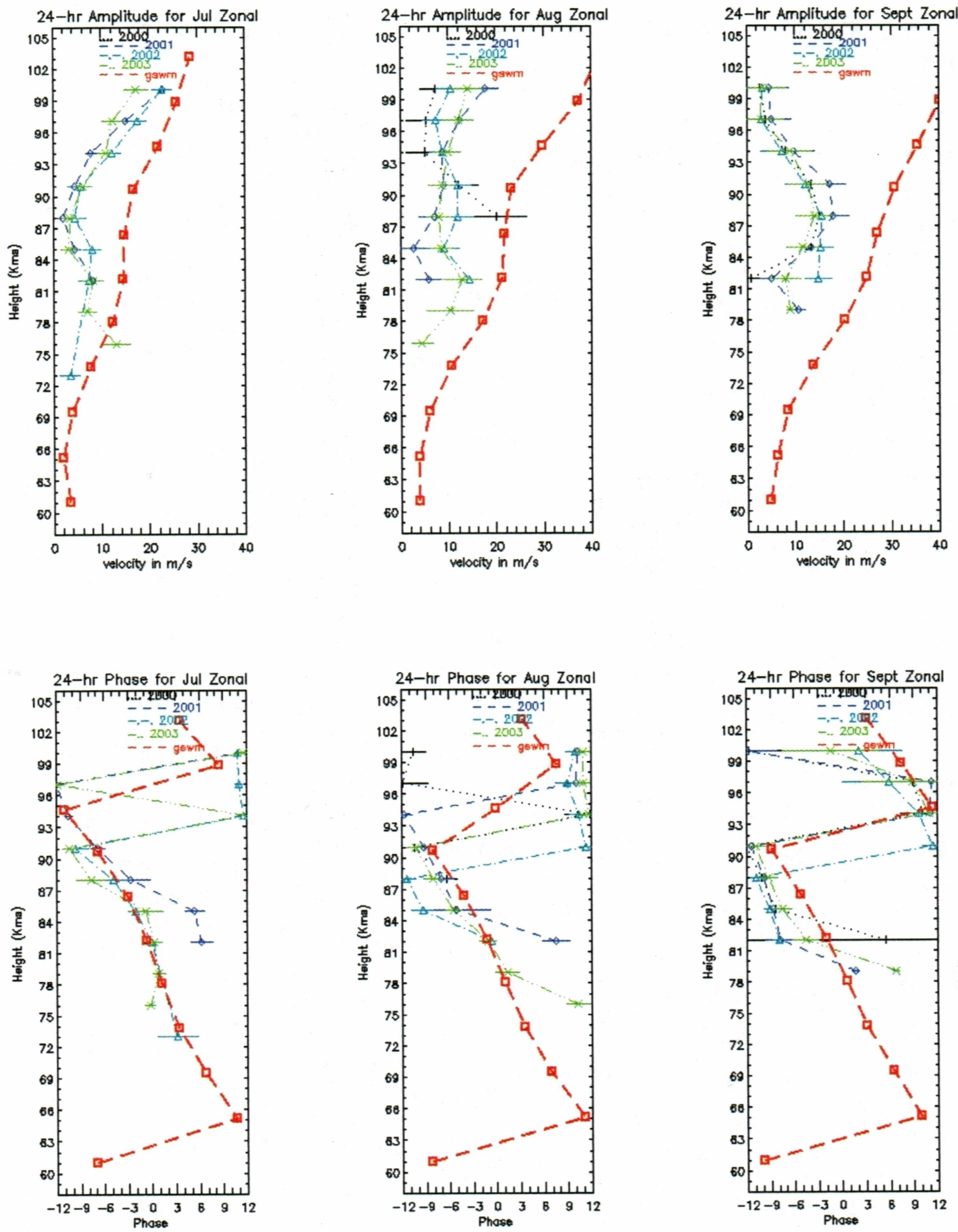



Figure 4-5 Zonal diurnal amplitudes and phases (continue).

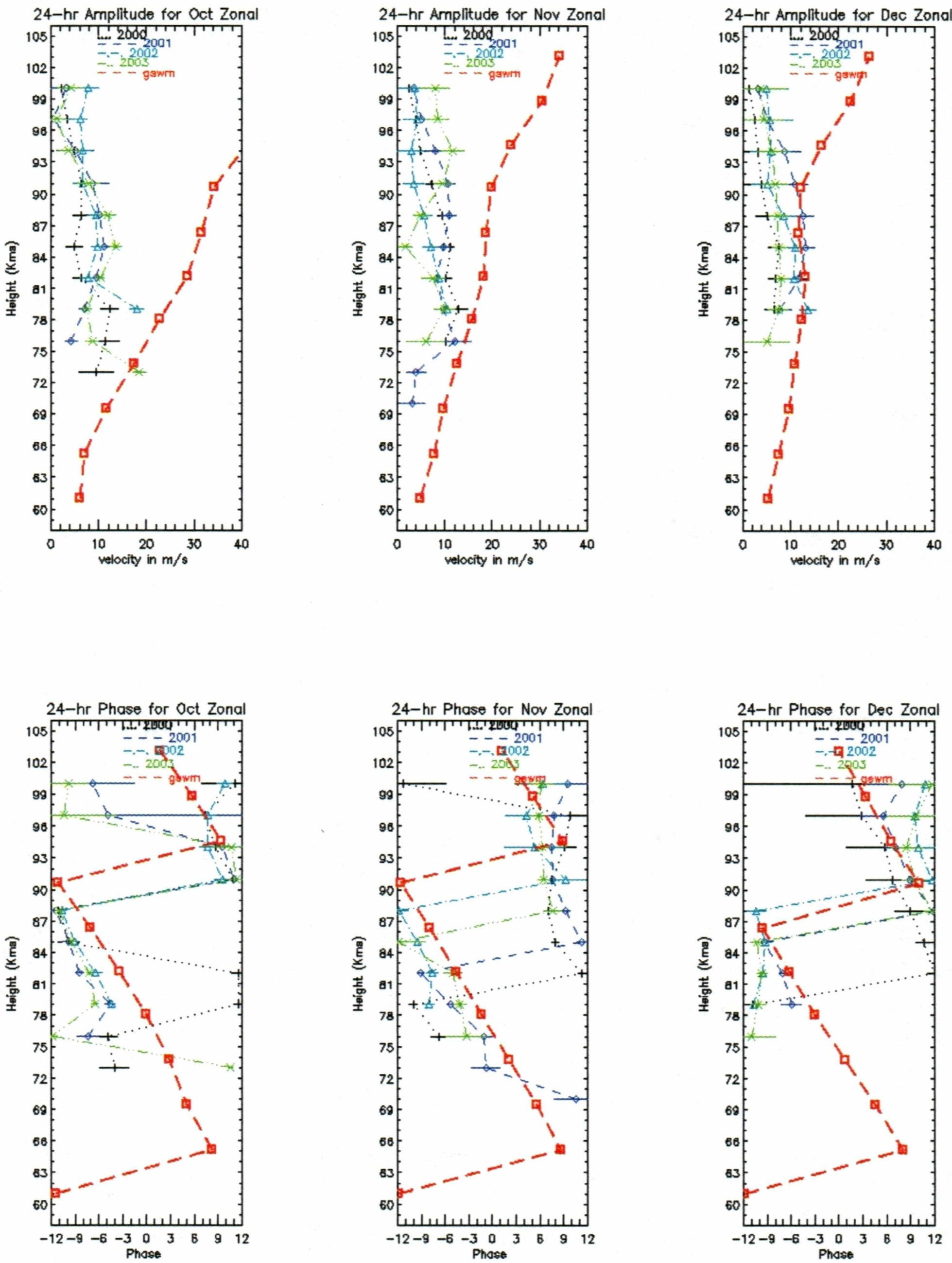



Figure 4-5 Zonal diurnal amplitudes and phases (continue).

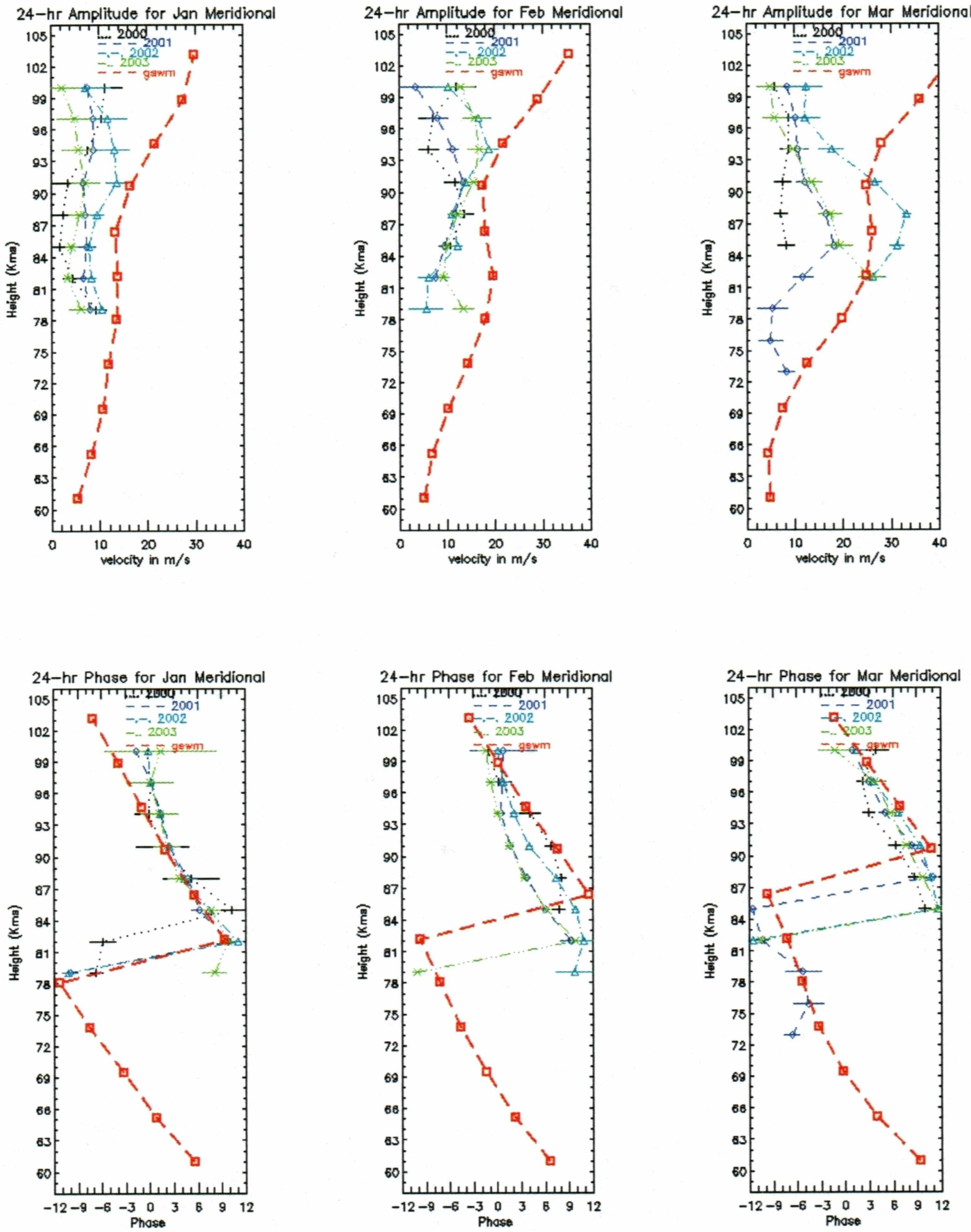



Figure 4-6 Meridional diurnal amplitudes and phases.

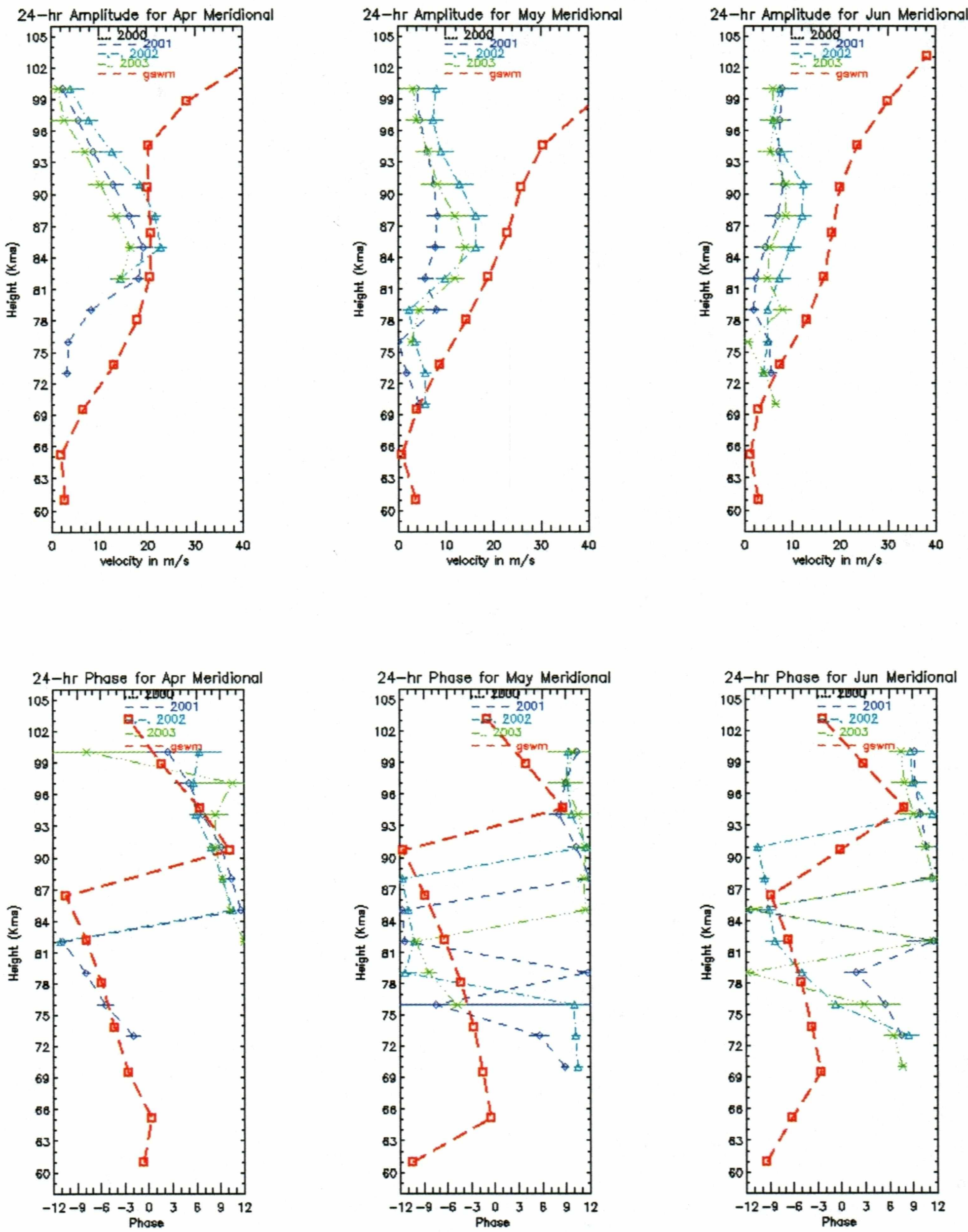



Figure 4-6 Meridional diurnal amplitudes and phases (continued).

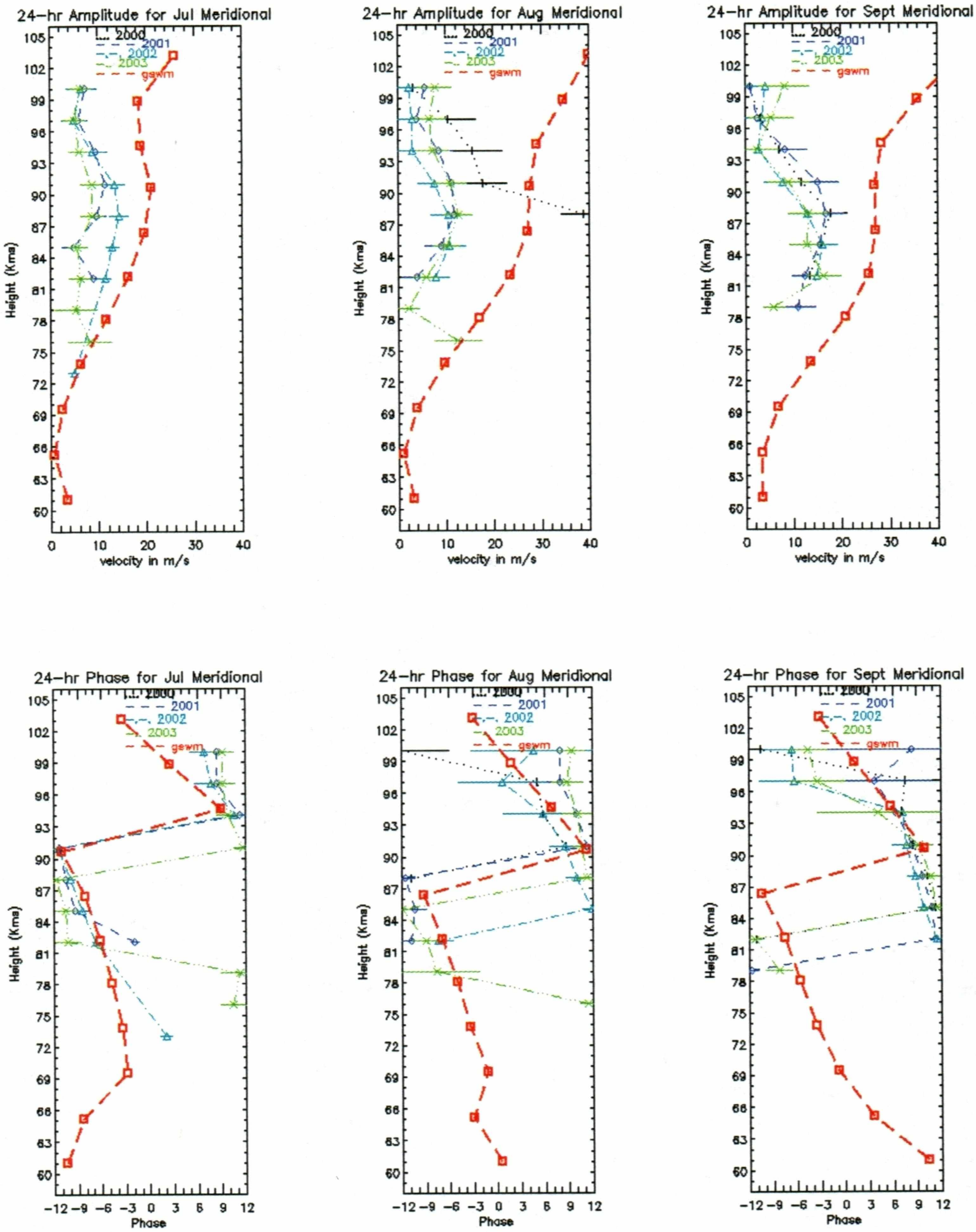



Figure 4-6 Meridional diurnal amplitudes and phases (continued).

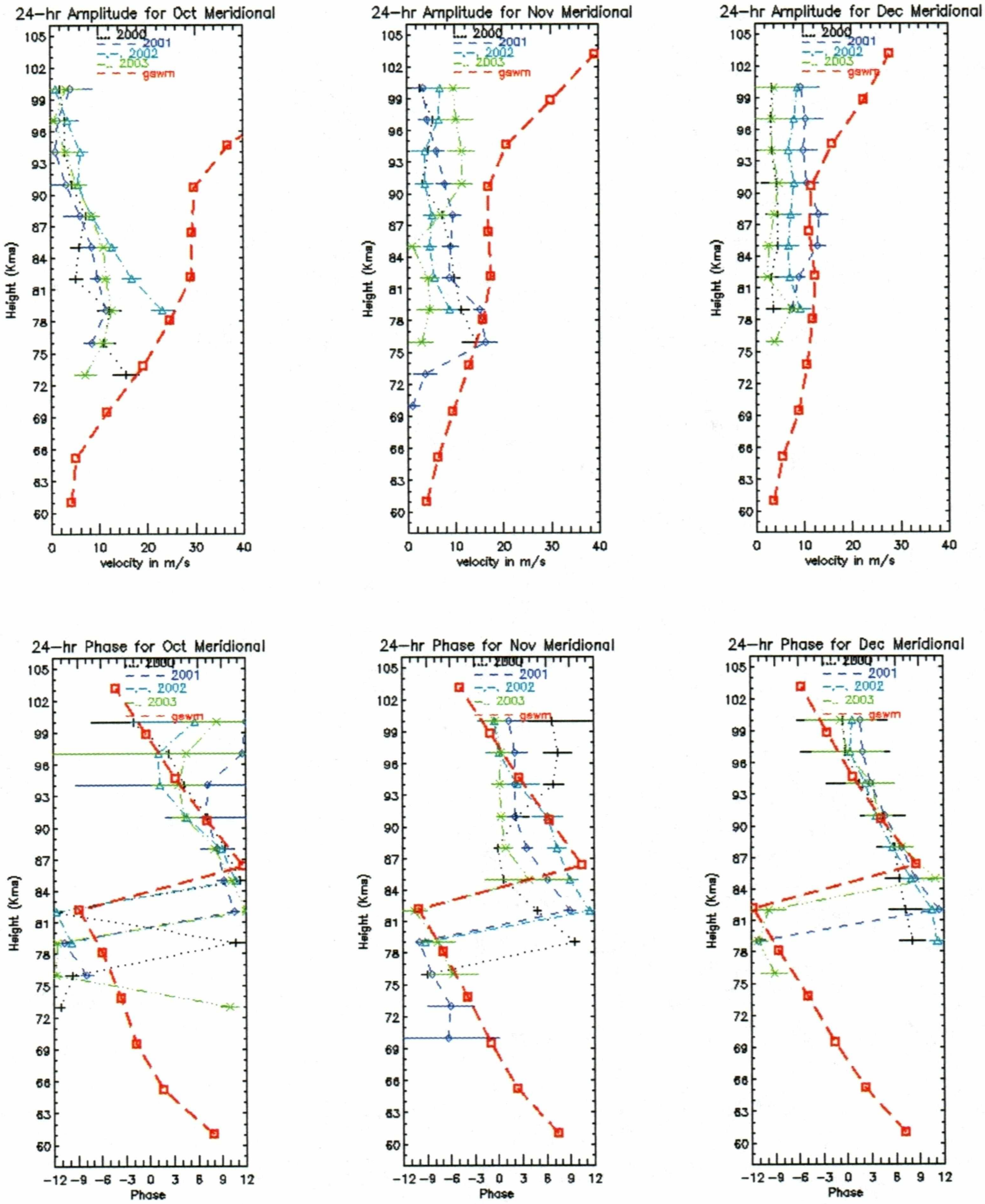



Figure 4-6 Meridional diurnal amplitudes and phases (continued).

### 4.3.2 Semidiurnal Tide

The semidiurnal tidal parameters are obtained for all the years from 2000-'04 meeting the criterion of a minimum of 8 hours of data availability and condition number less than 10. Figure 4-7 and Figure 4-8 show the zonal and meridional semidiurnal amplitudes and phases, respectively. The x-axis is the velocity in m/s in the top panel and the phase in hours in the bottom panel. The y-axis represents the heights in km in which we are interested.

The zonal semidiurnal tidal amplitude is increasing with height, although the increase is not large in magnitude. The GSWM data are almost overlapping our data. For the month of September, however, at heights above 82 km our data show an increasing trend, whereas the GSWM data don't show a significant increasing trend. For the remaining months both GSWM and our data show a similar trend over the heights. The August amplitude goes to a maximum of 28 m/s during August 2000. Similar to the semidiurnal amplitude during September [*Franke and Thorsen*, 1993], our amplitude shows an increasing trend, with a maximum velocity around 20 m/s and very small amplitudes around 4 m/s in October. For the year 2001 the amplitude in October is still greater than those of other years at heights above 84 km. During October 1991 above 84 km the semidiurnal amplitude over Urbana is around 12 m/s. At Saskatoon [*Manson, et al.* 1989] semidiurnal amplitude during October 1985 is 15 m/s above 80 km and increases with height. The zonal semidiurnal phase of each year is not exactly the same. Although for most of the months our phase is similar to the GSWM phase especially over upper heights above 85 km. The Urbana data for December 1991 shows a negative progression of phase above 84 km similar to that of our data during December. At Saskatoon during the same period of December 1985 [*Manson et al.*, 2002] semidiurnal phase shows a negative trend above 84 km with -3 hours at 82 km.

The meridional semidiurnal amplitude (Figure 4-8) has smaller magnitudes below 20 m/s, excepting only January 2002, when the magnitude went above 20 m/s. During October 2001 the semidiurnal amplitude above 82 km increases markedly when compared to other years and reaches a maximum of 22 m/s. In November above 79 km the amplitude of GSWM shows a decreasing and then an increasing trend whereas our data show a increasing trend with maximum velocity reaching 20 m/s. For the months of January to April the GSWM data increases in magnitude drastically above 85 km, whereas our velocity data remain less, in the range of 10 m/s. Other than that there is consistency for the semidiurnal amplitude with the GSWM amplitude. At Saskatoon [*Manson et al.*, 2002] during October 1985 the meridional semidiurnal amplitude is around 18 m/s above 82 km, and at Urbana it is around 5 m/s. The phase does look similar except for the months of April to September above 82 km. It can be seen that the phase shows a semidiurnal pattern. For most of the other months the phase at lower heights is not similar to that of the GSWM. During October the phase between the heights of 76 km and 91 km is not the same for all the years. For 2002, the phase is reversed across the heights, when compared to the phases obtained for the remaining years during the same period. The semidiurnal phase at Saskatoon during October 1985 shows a negative trend above 82 km, whereas our data show a positive trend. At Urbana also the data show a negative trend of phase.

The diurnal and semidiurnal amplitudes and phases for the years 2000-04 are shown along with the GSWM data. The differences in the GSWM data and our data may be due to many reasons. For example there may be some parameters that the GSWM model overestimates, or the data may be influenced by other parameters which have not been considered for the model. The GSWM data are not exactly at our heights, but a comparison is made about the path they follow. In most of the cases the trend shown by our data is similar to the trend shown by GSWM model data. Also, in few instances, our data were comparable to historical data from Saskatoon and Urbana.

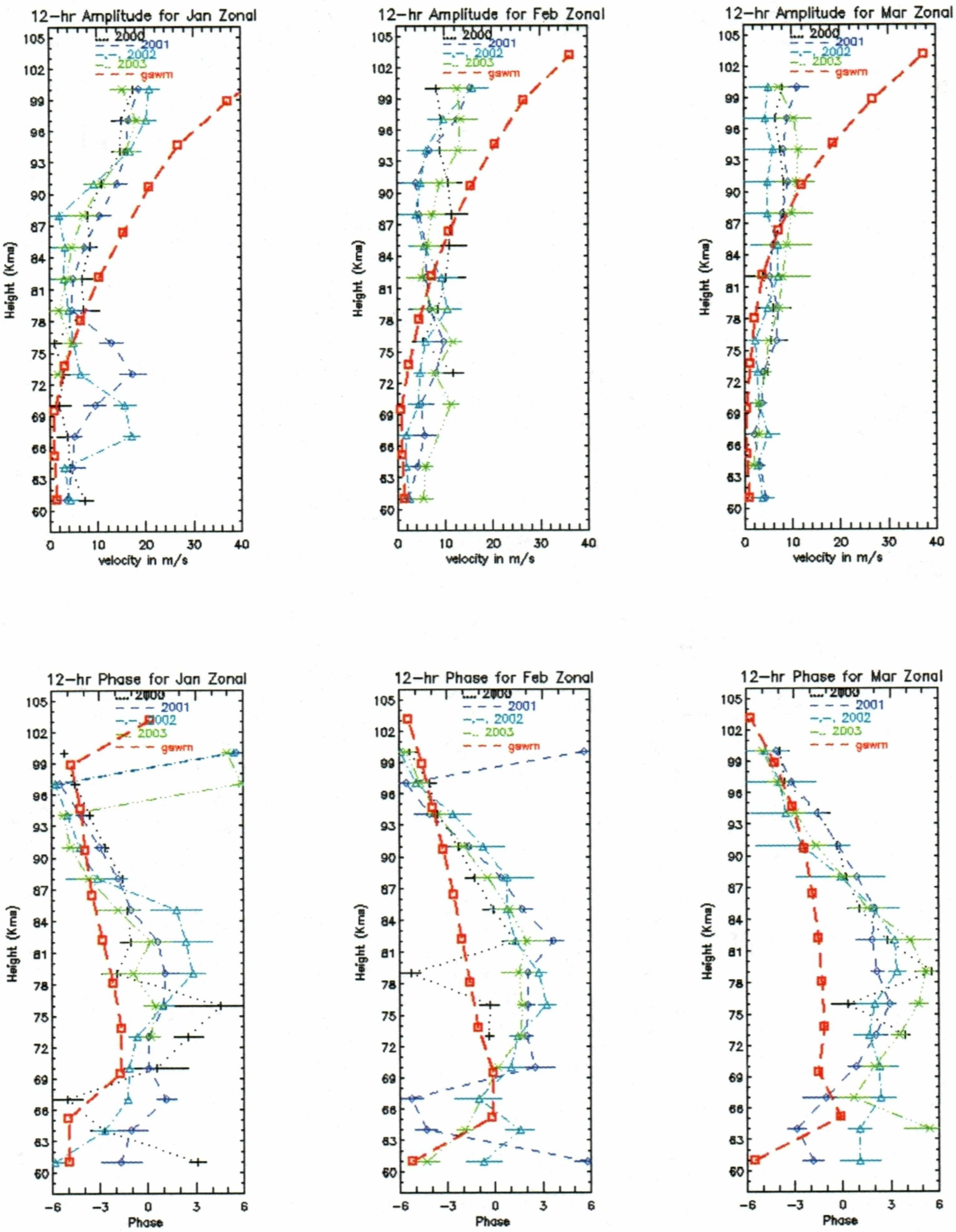



Figure 4-7 Zonal semidiurnal amplitudes and phases.

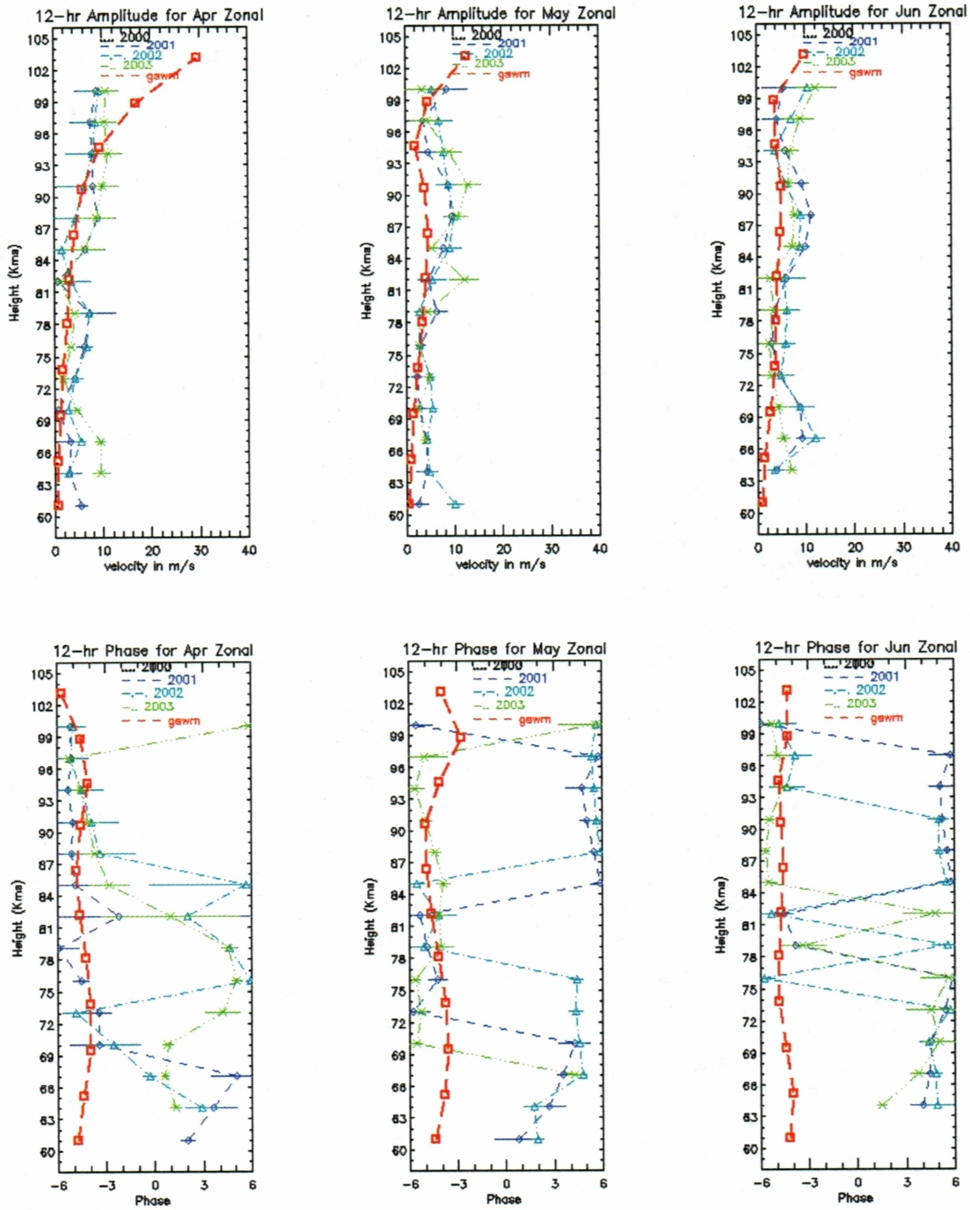



Figure 4-7 Zonal semidiurnal amplitudes and phases (continue)

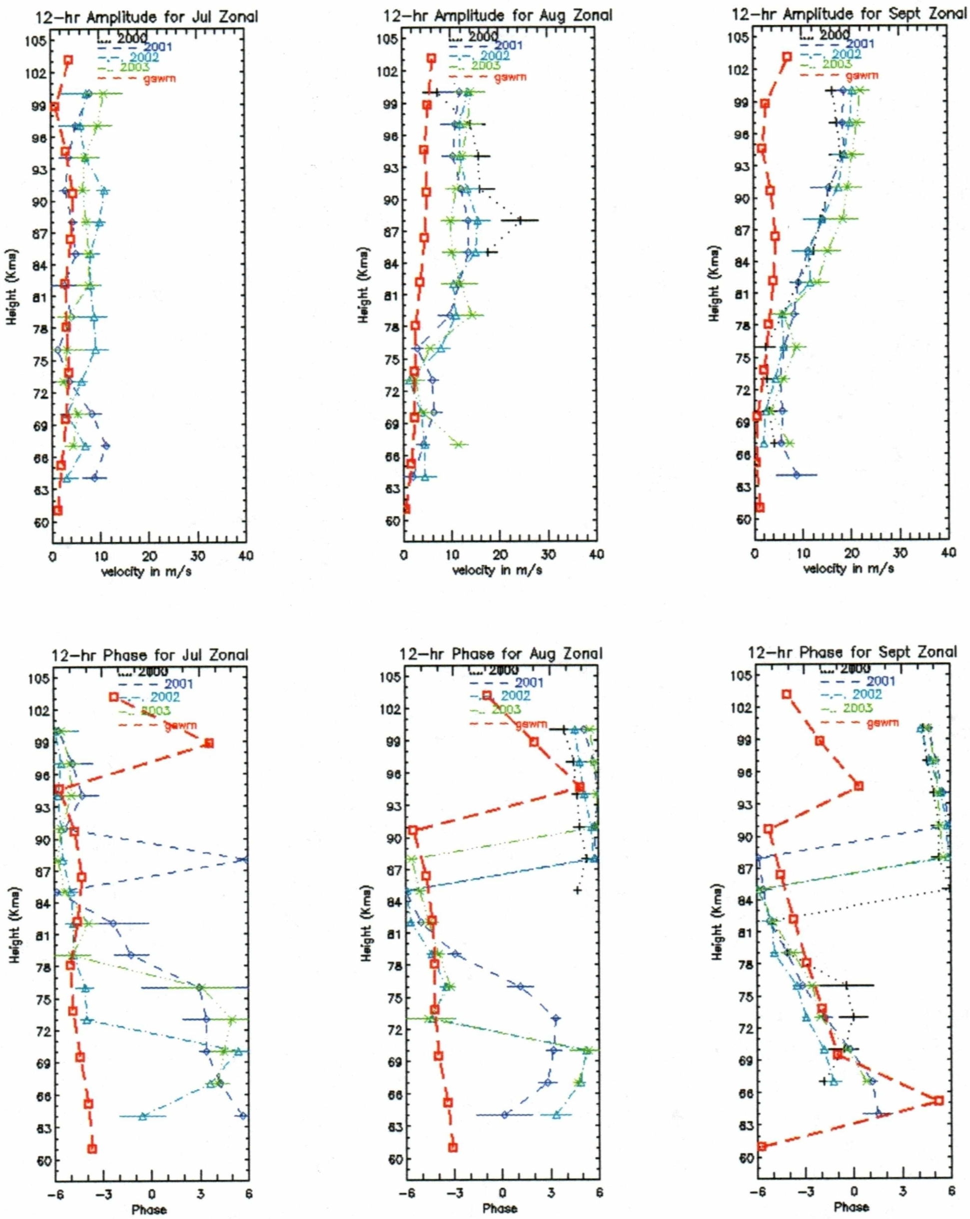



Figure 4-7 Zonal semidiurnal amplitudes and phases (continued).

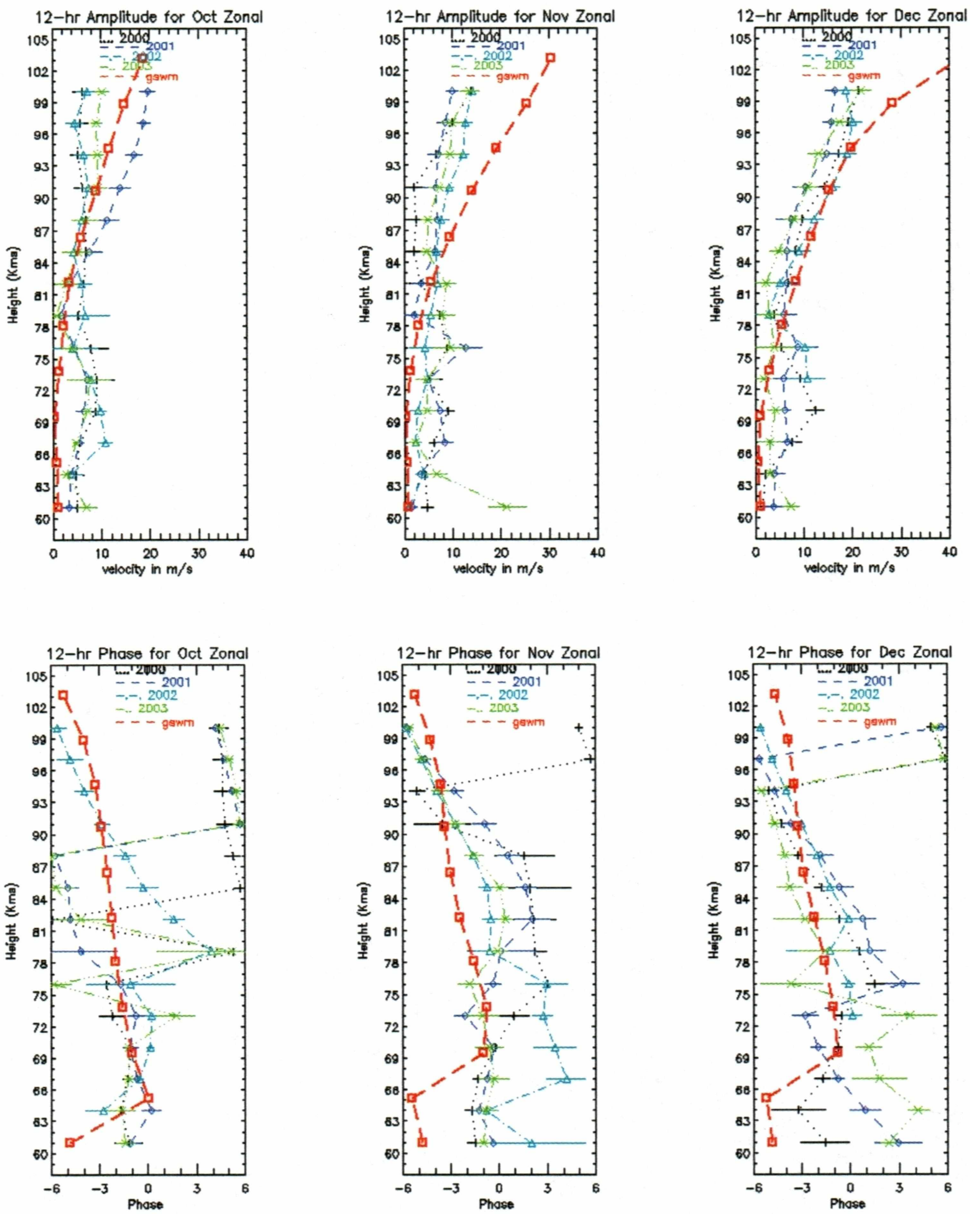



Figure 4-7 Zonal semidiurnal amplitudes and phases (continued).

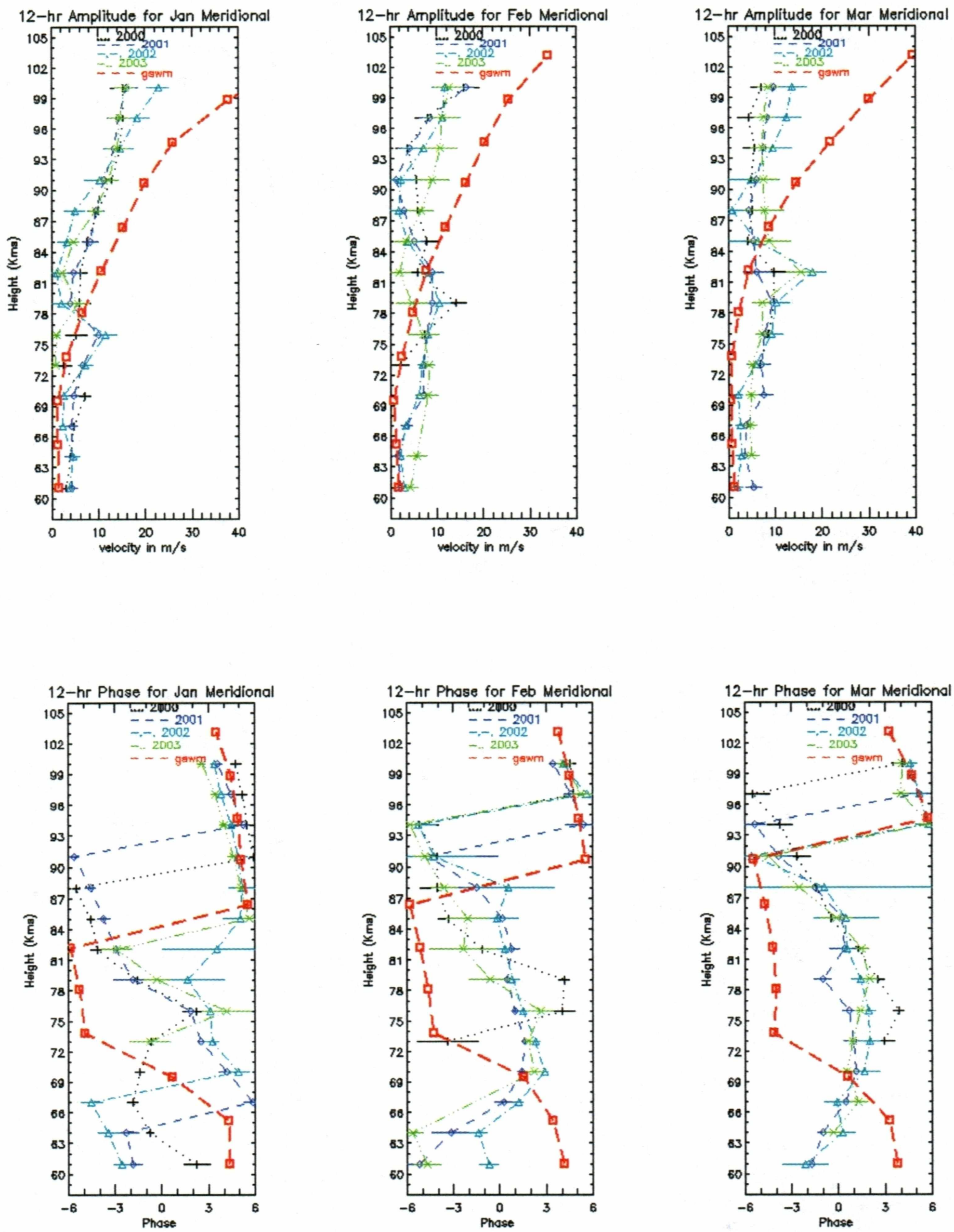



Figure 4-8 Meridional semidiurnal amplitudes and phases.

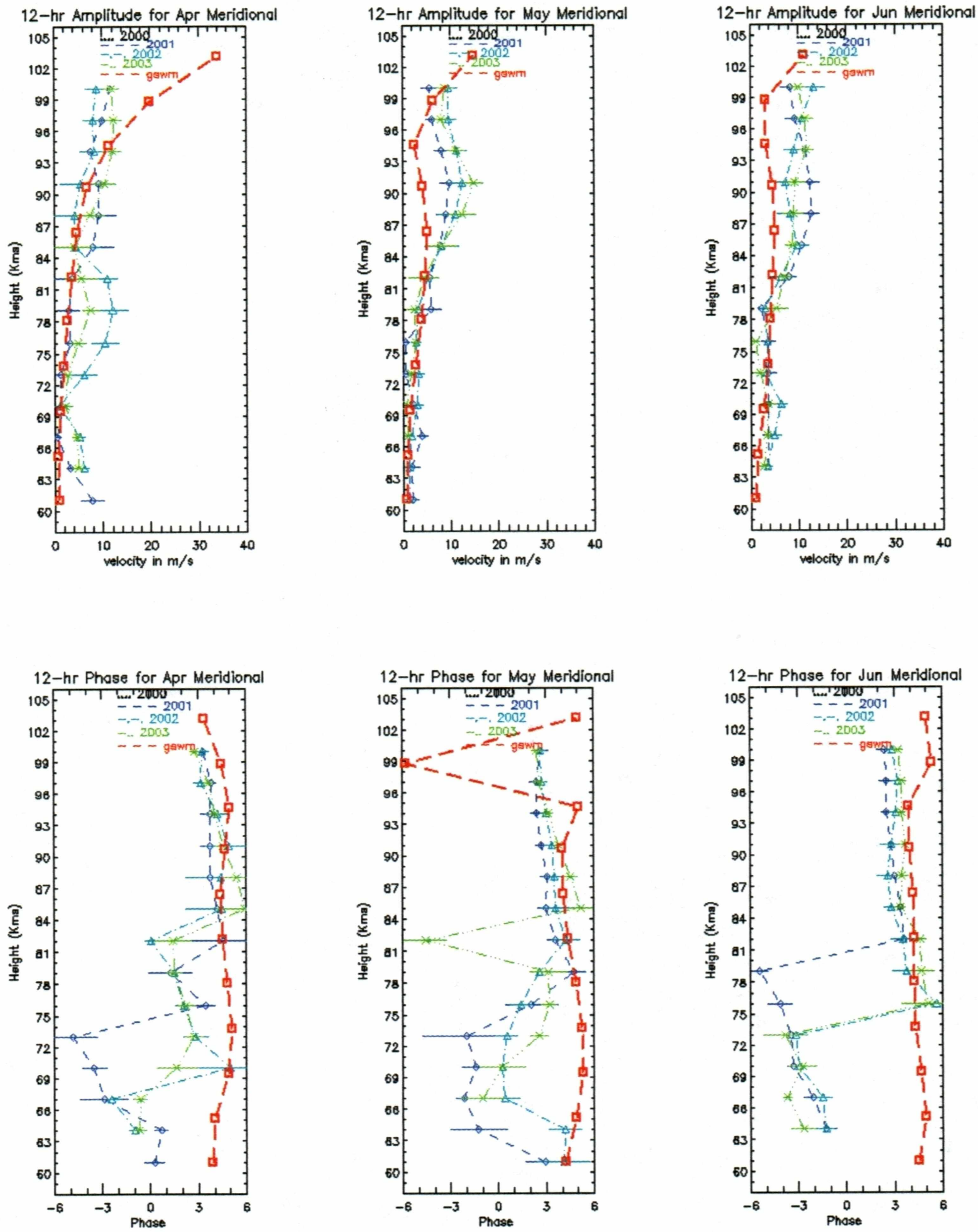



Figure 4-8 Meridional semidiurnal amplitudes and phases (continued).

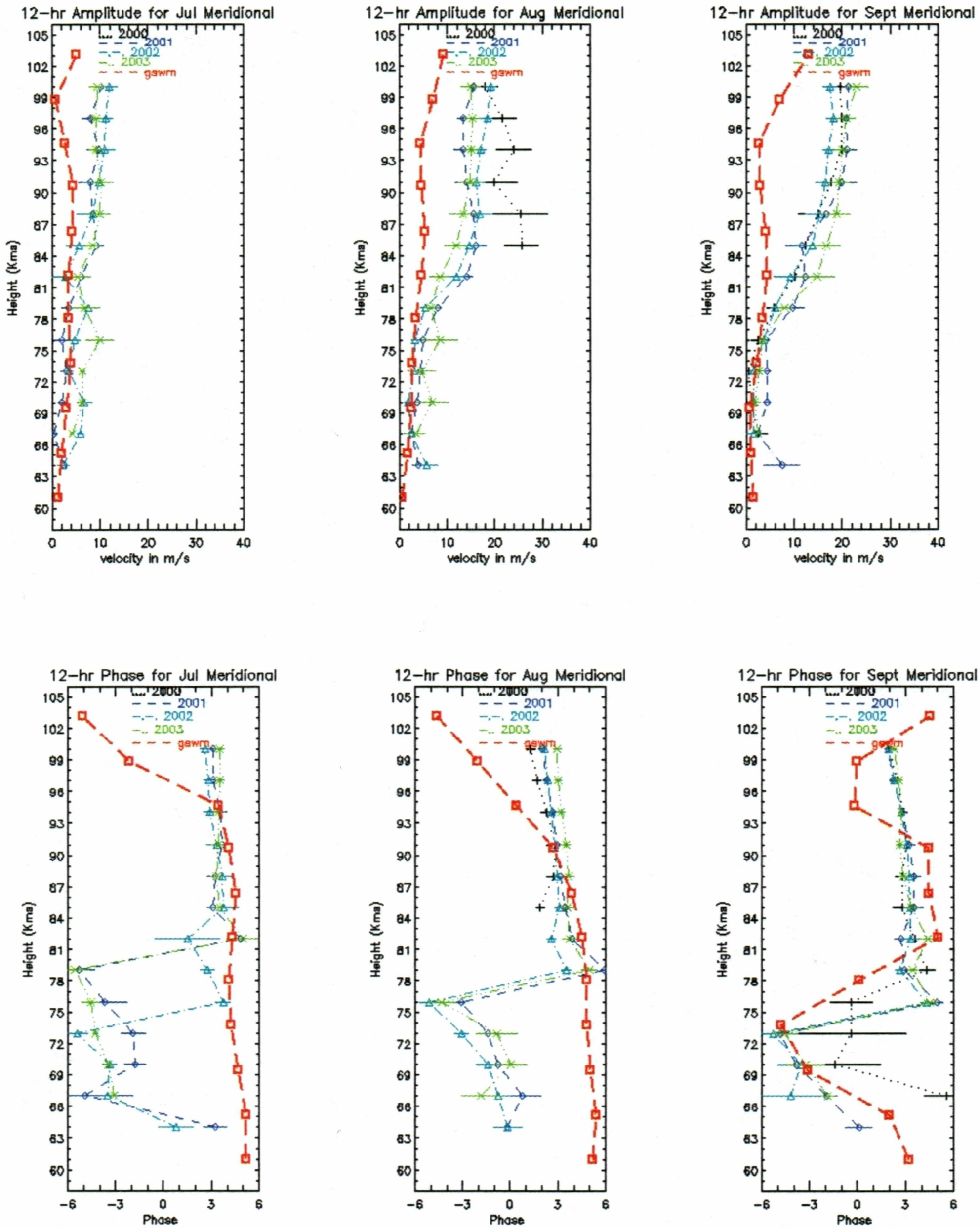



Figure 4-8 Meridional semidiurnal amplitudes and phases (continued).

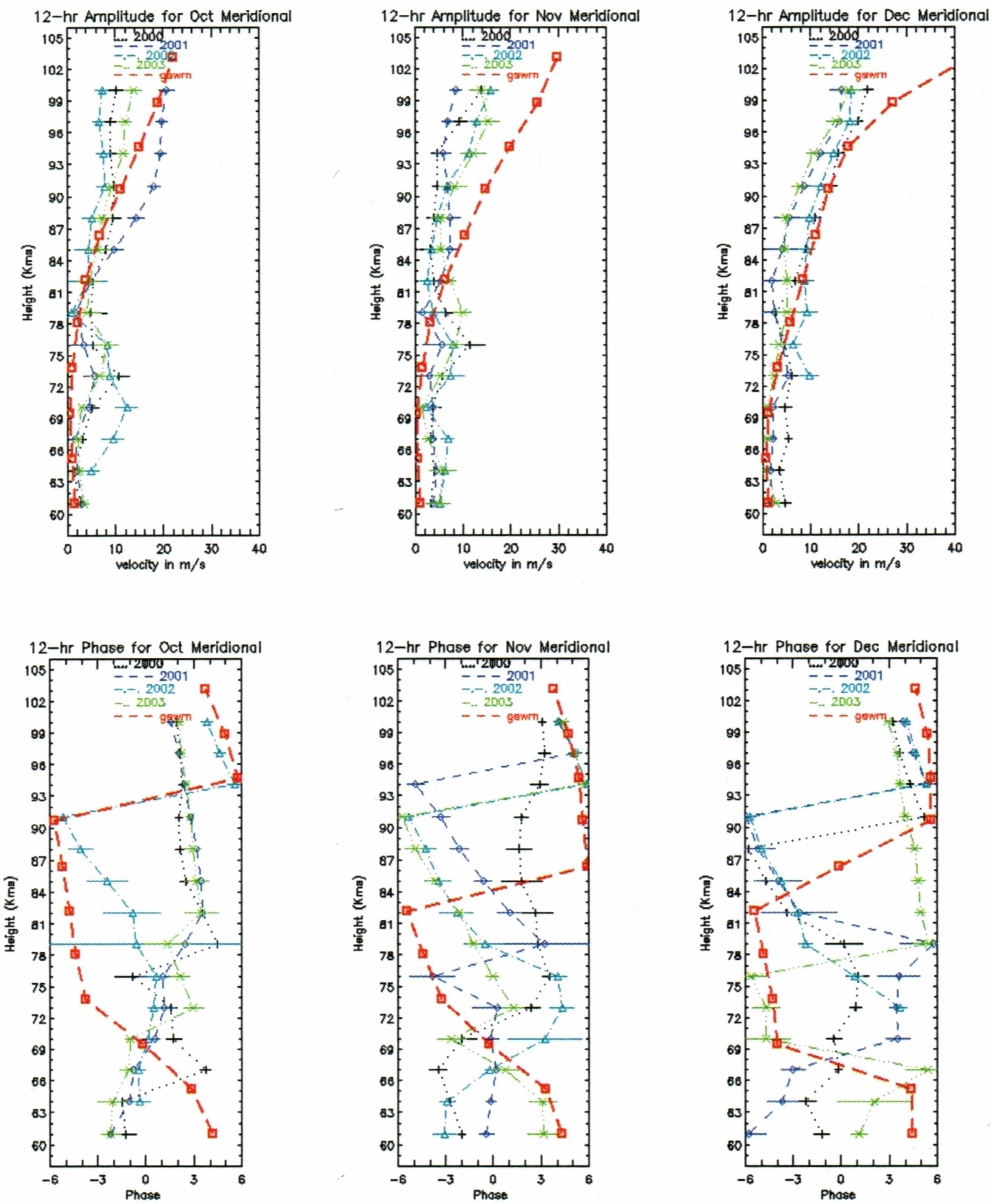



Figure 4-8 Meridional semidiurnal amplitudes and phases (continued).

# 5 Conclusions and Suggestions for Future Work

The MF Radar at Platteville Atmospheric Observatory has been used to collect the wind estimates for four years from 2000-’04. These wind estimates are processed in the time domain using the least squares fitting technique. The tidal amplitudes and phases along with the mean wind velocities have been obtained using this method and an analysis of these parameters is made.

The time domain analysis of the MF radar data are studied and the criteria and conditions where we can assume the obtained parameters to be valid are identified. The tidal parameters are obtained using the harmonic least squares fitting routine. The minimum number of hours of data needed is eight to obtain 12-hour parameters and fifteen to obtain the 24-hour tidal parameters. Similarly the condition number criterion of having a condition number less than ten is decided. Error analysis using the Monte Carlo method was done and compared with the least squares method. Both seem to be giving the same values and hence we have decided to use the errors obtained using the least squares fitting method.

We investigated the spectral domain techniques to analyze the MF data. The Lomb-Scargle periodogram method is studied to obtain the spectral peaks in contrast to the standard FFT method. The Lomb-Scargle method can be used for non-uniformly sampled data, when compared to the standard FFT method. Once we find a significant frequency which we believe is the actual frequency, then that significant frequency can be removed to see if any other frequencies are significant. Comparing the Fourier transform and Lomb-Scargle methods, it is possible that the Lomb-Scargle method might give better results, especially at lower heights where the data are more systematically distributed. It is always beneficial to know the dominant frequencies existing in the data before processing the data with the least squares fitting method. However, for our data, we were not able to find exactly 12 and 24 hour periods using the Lomb-Scargle method.

However, for most of the heights we do see that the 12 and 24-hour peaks are dominant and can be observed.

A quantitative study of the Platteville MF Radar data are done and compared with the data from Urbana (40°N, 88°W) and Saskatoon (52°N, 107°W). Along with these historical data, GSWM -02 model data are also compared and studied. In most parts of the year our data show a similar trend in amplitude and phase as GSWM-02 data. But in few months there is a drastic difference between these two data sets. That might be because of overestimation of parameters used in the model.

A thorough investigation into the Lomb-Scargle method must be made in the future. Once we are able to exactly locate the spectral frequencies contained in the data, then the tidal parameters of only those frequencies of interest can be obtained, using least squares method with more reliability. We can also compare our data from Platteville MF radar with those from other radars on the longitudinal circle.

**References**


Andrews, D., J. Holton and C. Leovy, Middle Atmosphere Dynamics, *Academic Press*, 1987

Bradley, E., The Jackknife, the Bootstrap and Other Resampling Plans, Society for Industrial and Applied Mathematics, 1982

Briggs, B.H., G.J. Phillips, and D.H. Shinn, The Analysis of Observations on Spaced Receivers of the Fading of Radio Signals, *Proc. Phys. Soc.* B, 63, 106-121, 1950

Chapman, S., and R. S. Lindzen, Atmospheric Tides, Gordon and Breach Science Publishers, Inc., 1970

David, J.C., and J. M. Forbes, On the Extraction of Tidal Information From Measurements Covering a Fraction of a Day, *Geophysical Research Letter*, 10, 580-582, 1983

Ferraz-Mello, Estimation of Periods from Unequally Spaced Observations, *The Astronomical Journal*, 86, 619, April 1981

Forbes, J.M., Middle Atmosphere Tides, *Jour. of Atm. and Terr. Phys.,* 46, 1049-1067, 1984

Forbes, J.M., Tidal and Planetary Waves, In The Upper Mesosphere and Lower Thermosphere: A Review of Experiment and Theory, *Geophysical Monograph,* 67-87, 1987

Franke, S. J. and D. Thorsen, Mean Winds and Tides in the Upper Middle Atmosphere at Urbana (40N,88W) During 1991-1992, *JGR,* 98, 18607-18615, 1993

Fritts, D.C and M.J. Alexander, Gravity Wave Dynamics and Effects in the Middle Atmosphere, *Rev. Geophys.*, 41, 1003, 2003

Hagan, M., GSWM-98: Results for migrating solar tides, *Journal of Geophysical Research,* 104, 6813-6827, 1999

Hagan, M., GSWM : Global Scale Wave Model, 2003

http://sisko.colorado.edu/TIMED/GSWM.html Accesed: April, 2004

Lomb, N. R., Least-Squares Frequency Analysis of Unequally Spaced Data, *Astrophysics and Space Science*, 39, 447-462, 1976

Manson, A.H., C.E. Meek, M.J. Smith and G.J. Fraser, Direct Comparisons of Prevailing Winds and Tidal Wind Fields (24-, 12-h) in the Upper Middle Atmosphere (60-105 km) during 1978-1980 at Saskatoon (52°N, 107°W) and Christchurch (44°S, 173°E), *Jour. of Atm. and Terr. Phys.,* 47, 463-476,1985.

Manson, A.H., Y. Luo, and C. Meek, Global Distributions of Diurnal and Semi-diurnal Tides: Observations from HRDI-UARS of the MLT Region, *Report #1, University of Saskatchewan, Saskatoon,* CANADA, 2002

Manson, A.H., C. Meek, M. Hagan, J. Koshyk, S. Franke, D. Fritts, C. Hall, W. Hocking, K. Igarashi, J. MacDougall, D. Riggin, and R. Vincent, Seasonal Variations of the Semi-diurnal and Diurnal Tides in the MLT: Multi-year MF Radar Observations from 2-70 N, Modeled Tides (GSWM,CMAM), *Annales Geophysicae* 20, 661-677, 2002

Manson, A.H. and C. E. Meek, Dynamics of the Middle Atmosphere at Saskatoon (52°N, 107° W): a Spectral Study during 1981, 1982, *Jour. of Atm. and Terr. Phys.,* 488, 1039-1055, 1986

Meek, C.E., An Efficient Method for Analyzing Ionospheric Drifts Data, *Jour. of Atm. and Terr. Phys.,* 42, 835-839, 1980

Palo, S. Routine for Calculating Harmonic Least Square Fit, IDL algorithm file, harmonic_ls_fit.pro, 1997

Palo, S. E. email correspondence, 2003

Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in C The Art of Scientific Computing, Cambridge University Press, 1992

Scargle, J. D., Studies in Astronomical Time Series Analysis II. Statistical Aspects of Spectral Analysis of Unevenly Spaced Data, *Astrophysical Journal,* 263, 835-853, 1982

Solomon, S. Stratospheric Ozone Depletion: A review of Concepts and History, *Rev. Geophys.* 37, 1999

Thorsen, D., Platteville Atmospheric Observation, 2000 http://grison.colorado.edu/RadarStations/Platteville/Platte2MHzMF.html Accessed: August, 2004

# Appendix A

The following code is used to obtain the Monte Carlo error estimation.

```
*****

PRO hour_ave_ls_fit_montetest, dir, year, month, dirplots,dir_out,PLA =
pla, PKR =pkr
;
; This procedure processes one months worth of MF raw data, generates ;
;hourly average data and
; produces the fit parameters and calculates the standard deviation
;using Monte Carlo method
; dir  = Directory where the raw data are located
; year = desired year (must be 4 or 2 digits)
; month = desired month (must be a number between 1, 12)
; dirplots = Directory where plots will be written
; dir_out = Directory where the output U and V array fit parameters
;files are saved

; OUTPUTS:

; The fit parameters of the hourly average data are written


;************* Check for PLA or PKR **************
 PLA = 'pla'
; PKR = 'pkr'
 dir = '/home/vemula/idl/platmf/Data/rawdata/'
 year = 2001
 month = 01
 dir_out = '/home/vemula/idl/platmf/processed_data/avefiles/'
;month_dayfiles/'
 dirplots = '/home/vemula/idl/platmf/Plots/' ;/month_ave/'
 long = 104.6

syear2 = STRING((year-year/100*100),format='(I2.2)')

;*********** Adjusting the longitude depending on the station *******
; IF PLA then KEYWORD_SET(PLA) gives 1 else 0

      IF((KEYWORD_SET(pla) and KEYWORD_SET(pkr)) EQ 1) THEN BEGIN
            PRINT, ' Entered both PLA and PKR '
            result = DIALOG_MESSAGE('BOTH PLA AND PKR ARE ENTERED
',/ERROR)
      ENDIF

      IF (KEYWORD_SET(pla)) THEN BEGIN
        long = 104.6
      ENDIF
```

```
      IF (KEYWORD_SET(pkr)) THEN BEGIN
        long = 147.49
      ENDIF

;************* Read in a months worth of data **************
;
  days = read_mf_raw_month(dir,year,month,data)
  IF(days EQ 0) THEN BEGIN
    PRINT,'No data read!'
    RETURN
  ENDIF

  syear = STRTRIM(STRING(year),2)
  smonth = STRTRIM(STRING(month),2)
  sdays = STRTRIM(STRING(days),2)

;************* Rejection criteria **************************
;
  minheight = 60
  maxheight = 100
  minsnr = 2.

  ;************* Limit data do a specific height range *********
  ;************* and reform into height-time matrix ************
  ;
  heightindex = WHERE((data[0].height.height GE minheight) AND $
            (data[0].height.height LE maxheight))
  hgtarray  = data[0].height[heightindex].height
  heightsize = SIZE(heightindex,/N_ELEMENTS)

  timesize  = SIZE(data,/N_ELEMENTS)
  tm = {TimeRec,day:0, hour:0, min:0}
  timearray = Replicate(tm,timesize)
  timearray.day = data.day
  timearray.hour = data.hour
  timearray.min = data.min

  errarray = data.height[heightindex].err

  SNRarray = data.height[heightindex].NF
    index = WHERE(SNRarray GE 1., count)
    IF(count GT 0) THEN SNRarray[index] = 0.999
    index = WHERE(SNRarray LT 1, count)
    SNRarray = SNRarray/(1.-SNRarray)

  uarray  = data.height[heightindex].u
  varray  = data.height[heightindex].v

;************ Select only valid data "err = 0" ************
;
  index = WHERE(errarray NE 0, count)
  IF(count NE 0) THEN BEGIN
    uarray[index] = !VALUES.F_NAN
```

```
    varray[index] = !VALUES.F_NAN
  ENDIF

;************ Select only robust data "snr > 2" ************
;

  index = WHERE(SNRarray LT minsnr,count)
  IF(count NE 0) THEN BEGIN
    uarray[index] = !VALUES.F_NAN
    varray[index] = !VALUES.F_NAN
  ENDIF

  umin = MIN(uarray,/NAN)
  umax = MAX(uarray,/NAN)
  vmin = MIN(varray,/NAN)
  vmax = MAX(varray,/NAN)

;***** To convert from UT to SLT *******
;
   tmp_time = time_convert(long,timearray)

      IF(tmp_time NE 0) THEN BEGIN
        PRINT, 'ERROR : time_convert'
        RETURN
      ENDIF

;***** To get the hourly averages *******

   tmp = hour_average(uarray,timearray,u_avg,t_avg)
      IF(tmp NE 0) THEN BEGIN
      PRINT,'ERROR: hour_average'
      RETURN
      ENDIF


;**** Extracting actual data from the u_avg array(mean+dev+numpts)
******

 sz = size(u_avg)
 actual_data = fltarr(14,sz[2])
j=0
      FOR i=0,39,3 DO BEGIN
            actual_data[j,*] = u_avg[i,*]
            j=j+1
      ENDFOR
 time_svd = t_avg.day*24+t_avg.hour+(t_avg.min)/60 ; timearray for the
LS fits in hours

;**** This part is used to produce the LS fits ******

 periods = [12]   ; need to change the period for 12 or 24
 fit_data_12 = fltarr(14,sz[2])  ; array with the fitted data (not the
parameters)
```

```
 data_hours_12 = fltarr(14)  ; array with the number of data points at
each height
 fits_12 = fltarr(8,14)
 fitsarray_12 = fltarr(6,14)
 fits_montecarlo_12 = fltarr(6,14)
 condnum_montecarlo_12 = fltarr(14)
 condnum_12 = fltarr(14)
 height_array_12 = hgtarray

 j=0
 k=0

 FOR i = 0,13,1 DO BEGIN
   tmpdatafit_12 = actual_data[i,*]
   tmphours_12 = WHERE(FINITE(tmpdatafit_12) eq 1, count_12)  ;
tmphours has the values of hours when data are available

   scount = STRING(count_12)
   IF (count_12 eq 0) THEN BEGIN
        tmphours_12 = 0
        data_hours_12(j) = 0
        fits_12[*,k] = !Values.(1)
        condnum_12(j) = !Values.(1)
      ENDIF

      fit_data_12[j,*] = 'NaN'
      IF (count_12 GT 0) THEN BEGIN
        hours_12 = time_svd(tmphours_12)
        tmp_datafit_12 = tmpdatafit_12[tmphours_12]

result=harmonic_ls_fit(hours_12,tmp_datafit_12,periods,a_fit,b_yhat)

        condnum_12(j) = result
        fit_data_12[j,[tmphours_12]] = b_yhat
        fits_12[*,k] = reform(a_fit, [8,1])
        data_hours_12(j) = N_ELEMENTS(hours_12)

result_monte=monte_carlo_hr_ave(hours_12,tmp_datafit_12,periods,fits_mo
ntecarlo)
        fits_montecarlo_12[*,k] = fits_montecarlo[0:5,0]
        condnum_montecarlo_12(j) = fits_montecarlo[6,0]
      ENDIF

      j=j+1
      k=k+1

  ENDFOR
  fitsarray_12 = [fits_12[0:1,*],fits_12[4:7,*]] ; Extracting the actual
fits removing zeros

;***** writing the fitted data of 12-hr Monte Carlo evaluation of
Standard Deviation ****
 U_V = 'u'
 zonal_meridional = 'Zonal'
```

```
 smonth = STRING(month,format='(I2.2)')

 IF (periods EQ 12) THEN BEGIN
      write_u_v_fit_ave_12hr_monte,fits_montecarlo_12,height_array_12,d
ir_out,syear,syear2, $
      smonth, U_V, zonal_meridional,condnum_montecarlo_12,data_hours_12
 ENDIF ELSE BEGIN
       PRINT,'Error! Entered periods =[24] '
    ENDELSE

;******** writing the fitted data of the 12-hr tide **********
write_u_v_fit_ave_12hr,fitsarray_12,height_array_12,dir_out,syear,syear
2,smonth,U_V, $

 periods = [24]
 fit_data_24 = fltarr(14,sz[2])  ; array with the fitted data (not ;the
parameters)
 data_hours_24 = fltarr(14)  ; array with the number of data points at
;each height
 fits_24 = fltarr(8,14)
 fitsarray_24 = fltarr(6,14)
 fits_montecarlo_24 = fltarr(6,14)
 condnum_montecarlo_24 = fltarr(14)
 condnum_24 = fltarr(14)
 height_array_24 = hgtarray

 j=0
 k=0

 FOR i = 0,13,1 DO BEGIN
  tmpdatafit_24 = actual_data[i,*]
  tmphours_24 = WHERE(FINITE(tmpdatafit_24) eq 1, count_24)
; tmphours_24 has the values of hours when data are available

   scount = STRING(count_24)
   IF (count_24 eq 0) THEN BEGIN
       tmphours_24 = 0
       data_hours_24(j) = 0
       fits_24[*,k] = !Values.(1)
       condnum_24(j) = !Values.(1)
      ENDIF

      fit_data_24[j,*] = 'NaN'
      IF (count_24 GT 0) THEN BEGIN
        hours_24 = time_svd(tmphours_24)
        tmp_datafit_24 = tmpdatafit_24[tmphours_24]
result = harmonic_ls_fit(hours_24,tmp_datafit_24,periods,a_fit,b_yhat)

        condnum_24(j) = result
        fit_data_24[j,[tmphours_24]] = b_yhat
        fits_24[*,k] = reform(a_fit, [8,1])
        data_hours_24(j) = N_ELEMENTS(hours_24)
```

```
result_monte =
monte_carlo_hr_ave(hours_24,tmp_datafit_24,periods,fits_montecarlo)
        fits_montecarlo_24[*,k] = fits_montecarlo[0:5,0]
        condnum_montecarlo_24(j) = fits_montecarlo[6,0]
      ENDIF

      j=j+1
      k=k+1

 ENDFOR
 fitsarray_24 = [fits_24[0:1,*],fits_24[4:7,*]] ; Extracting the actual
;fits removing zeros

 IF (periods EQ 24) THEN BEGIN
      write_u_v_fit_ave_24hr_monte,fits_montecarlo_24,height_array_24,d
ir_out,syear,syear2,$
      smonth,U_V, zonal_meridional,condnum_montecarlo_24,data_hours_24
 ENDIF ELSE BEGIN
       PRINT,'Error! Entered periods =[12] '
    ENDELSE

print, 'hour_ave fits are generated '
END

*******

PRO write_u_v_fit_ave_24hr_monte,fitarray,height,dir,syear,syear2,
smonth,U_V,zonal_meridional,condnum,data_hours

; This is used to write the fit parameters of the 24-hr tide
; fitarray : The array generated after the fits
; height : height array

filename =
dir+syear+'/montecarlo_ls_fit/'+U_V+'_'+syear2+'_'+smonth+'_ave.fit_24_
nc_monte'
header=[['8',''],$
      ['******************************************',''],$
      ['Processed MF Radar Winds:',zonal_meridional],$
      ['Site : Platteville',''],$
      ['LS Fitted data for Hourly Average data files using montecarlo
method',''],$
      ['Year :',syear],$
      ['Month :',smonth],$
      ['Format: ',''],$
      ['Height DC-Amp DC-dev 24Amp 24Amp.dev 24Ph 24Ph.dev condnum
hours',''],$
      ['****These are fit values with [24] as period and no min.
condition of data hours(points)',''],$
      ['******************************************','']]

sz = size(fitarray)
out_array = FLTARR(sz[1]+3,sz[2])
out_array[0,*] = height
```

```
out_array[1:sz[1],*] = fitarray
out_array[sz[1]+1,*] = condnum
out_array[sz[1]+2,*] = data_hours
format_array = '(F6.0,6(F30.4),F30.1,F6.0)' ; used when 12 hours period
is used

;Writing the fits to the file ***********
write_to_fit,filename,header,out_array,format_array

END

*******

PRO write_u_v_fit_ave_12hr_monte,fitarray,height,dir,syear,syear2,
smonth,U_V,zonal_meridional,condnum,data_hours

; fitarray : The array generated after the fits
; height : height array

filename =
dir+syear+'/montecarlo_ls_fit/'+U_V+'_'+syear2+'_'+smonth+'_ave.fit_12_
nc_monte'

header=[['8',''],$
       ['*****************************************',''],$
       ['Processed MF Radar Winds:',zonal_meridional],$
       ['Site : Platteville',''],$
       ['LS Fitted data for Hourly Average data files using montecarlo
method',''],$
       ['Year :',syear],$
       ['Month :',smonth],$
       ['Format: ',''],$
       ['Height DC-Amp DC-dev 12Amp 12Amp.dev 12Ph 12Ph.dev condnum
hours',''],$
       ['****These are fit values with [12] as period and no min.
condition of data hours(points)',''],$
       ['*****************************************','']]

sz = size(fitarray)
out_array = FLTARR(sz[1]+3,sz[2])
out_array[0,*] = height
out_array[1:sz[1],*] = fitarray
out_array[sz[1]+1,*] = condnum
out_array[sz[1]+2,*] = data_hours
;format_array = '(F6.0,10(F30.1),F33,F6.0)' ; used for 12 and 24 hour
period
format_array = '(F6.0,6(F15.4),F25.1,F6.0)' ; used when 12 hours period
is used

;Writing the fits to the file ***********
write_to_fit,filename,header,out_array,format_array

END
```

```
********

pro write_to_fit,filename,header,outarray,farray

; To write the fitted data into a file

asz= size(outarray)

OPENW,unit,filename,/GET_LUN

;********** Printing the header into the file ************

PRINTF,unit,header

;********** Printing the actual fitted data into the file *********

FOR i=0,asz[2]-1 DO BEGIN
PRINTF,unit,FORMAT=farray,outarray[*,i]
ENDFOR

;********** Closing all the logical units used ***********
CLOSE, unit
FREE_LUN,unit

END

********

FUNCTION monte_carlo_hr_ave, hours,data, periods,fits_montecarlo
;This Procedure is used for the random number generation and the Monte
;Carlo error estimation
;It generates the random numbers from 0 to 743 and takes those random
;numbers as the index
;of the actual data and does it for 100 repetitions

;dir = '/home/vemula/idl/platmf/processed_data/avefiles/'

;year = 2001
;month = 01

sz = size(data)
rnd_arr = fltarr(sz[1],100)
fit_para = fltarr(8,100)
condnum = fltarr(100)
std_dev = fltarr(3,1)
fits_montecarlo = fltarr(8,1)

;*** random number generation and selection of 100%(744) of random
;placed data from the orginal data set

rnd_arr = randomu(0,[sz[1],100],/uniform)
rnd_arr = rnd_arr*(sz[1]+1)
rnd_arr = FIX(rnd_arr)
```

```
; The fit_para array has the fit parameters for 12 hr period

FOR i=0,99,1 DO BEGIN
      tmpdatafit = data[rnd_arr[*,i]]
      tmphours = hours[rnd_arr[*,i]]

result = harmonic_ls_fit(tmphours,tmpdatafit,periods,a_fit,b_yhat)

      fit_para[*,i] = reform(a_fit,8,1)
      condnum[i] = result
ENDFOR

; the std. dev of the mean values is calculated
 avg = TOTAL(fit_para[0,*])/100
 sum_diff = TOTAL([fit_para[0,*]-avg]^2)
 std_dev[0,0] = SQRT((1.0/(100-1))*sum_diff)

; The std. dev of the 12hr and 24-hr amp and phase is calculated
 i=1
      avg = TOTAL(fit_para[4,*])/100
      sum_diff = TOTAL([fit_para[4,*]-avg]^2)
      std_dev[1,0] = SQRT((1.0/(100-1))*sum_diff)
IF (periods EQ 24 and fit_para[6,0] GT 10) THEN BEGIN
      adjust = WHERE((fit_para[6,*] LT 0), cn)
      IF (cn EQ 0) THEN BEGIN
        avg = TOTAL(fit_para[6,*])/100
        sum_diff = TOTAL([fit_para[6,*]-avg]^2)
        std_dev[2,0] = SQRT((1.0/(100-1))*sum_diff)
      ENDIF ELSE BEGIN
        fit_para[6,adjust] = fit_para[6,adjust]+24
        avg = TOTAL(fit_para[6,*])/100
        sum_diff = TOTAL([fit_para[6,*]-avg]^2)
        std_dev[2,0] = SQRT((1.0/(100-1))*sum_diff)
      ENDELSE
 ENDIF ELSE BEGIN
      avg = TOTAL(fit_para[6,*])/100
      sum_diff = TOTAL([fit_para[6,*]-avg]^2)
      std_dev[2,0] = SQRT((1.0/(100-1))*sum_diff)
 ENDELSE

fits_montecarlo[0] = MEAN(fit_para[0,*],/NaN)
fits_montecarlo[1] = std_dev[0,0]
fits_montecarlo[2] = MEAN(fit_para[4,*],/NaN)
fits_montecarlo[3] = std_dev[1,0]
fits_montecarlo[4] = MEAN(fit_para[6,*],/NaN)
      IF (fits_montecarlo[4] GT 12) THEN fits_montecarlo[4] =
      fits_montecarlo[4]-24
      fits_montecarlo[5] = std_dev[2,0]
      fits_montecarlo[6] = MEAN(condnum,/NaN)

RETURN,1

END
```

********

```
PRO plot_hrave_fits_monte,dir,dir_plot,year,month, U=u, V=v

; This procedure is used to plot the tidal parameters of 24-hr and 12-
;hr tides with monte carlo error estimation

; INPUTS :
; The directory from which the data need to be read
; year,month

; OUTPUTS :
; The directory to which the plots need to be written

 WINDOW, RETAIN = 2
 loadct,39
 dir = '/home/vemula/idl/'
 dir_plot = '/home/vemula/idl/platmf/Plots/month_ave/fittedplots/'
 U ='u'
 year = 2001
 month = 1

; Converting the year and month into strings

   syear = STRTRIM(STRING(year),2)
      smonth = STRING(month,format='(I2.2)')
      syear2 = STRING((year-year/100*100),format='(I2.2)')


; File names set for reading the 24 and 12 hour parameters obtained
;from raw , hourly averages, month_day and GSWM data files

; Naming the hourly average fit parameter data files with montecarlo
;std. dev to read
 filename_24_hr_ave_monte =
dir+'platmf/processed_data/avefiles/'+syear+'/montecarlo_ls_fit/u_' $
            +syear2+'_'+smonth+'_ave.fit_24_nc_monte'

 filename_12_hr_ave_monte =
dir+'platmf/processed_data/avefiles/'+syear+'/montecarlo_ls_fit/u_' $
            +syear2+'_'+smonth+'_ave.fit_12_nc_monte'

; Naming the hourly average fit parametes data files to be read

 filename_24_hr_ave =
dir+'platmf/processed_data/avefiles/'+syear+'/nocutoff/ls_fit_sltadjust
/u_'$
            +syear2+'_'+smonth+'_ave.fit_24_nc'

 filename_12_hr_ave =
dir+'platmf/processed_data/avefiles/'+syear+'/nocutoff/ls_fit_sltadjust
/u_'$
            +syear2+'_'+smonth+'_ave.fit_12_nc'
```

```
; Constants for reading the data
 header = 0
 headerstr = ''
 data_24_ave = fltarr(9,14)    ; arrays with hourly average fit
                               ;parameters
 data_12_ave = fltarr(9,14)
 data_24_ave_monte = fltarr(9,14)
 data_12_ave_monte = fltarr(9,14)

; Reading the hourly average fit paramters (24-hr )data from the files

OPENR, unit, filename_24_hr_ave, /GET_LUN, ERROR = err
      ;****** Checking whether the file exists or not ****
      IF (err ne 0) then begin
         PRINT,'File Not Found for ', smonth, ' ', syear
       return
      ENDIF ELSE BEGIN
         PRINT,'Reading data' , filename_24_hr_ave
      ENDELSE


 READF,unit, header
      FOR i =0,header+2 DO BEGIN
        READF,unit,headerstr
      ENDFOR
 READF,unit,data_24_ave
CLOSE, /ALL

; **** Reading the 12-hr hourly average fit parameters
 OPENR, unit, filename_12_hr_ave, /GET_LUN, ERROR = err
      ;****** Checking whether the file exists or not ****
      IF (err ne 0) then begin
         PRINT,'File Not Found for ', smonth, ' ', syear
       return
      ENDIF ELSE BEGIN
         PRINT,'Reading data' , filename_12_hr_ave
      ENDELSE


 READF,unit, header
      FOR i =0,header+2 DO BEGIN
        READF,unit,headerstr
      ENDFOR
 READF,unit,data_12_ave
CLOSE, /ALL

; Reading the hourly average fit paramters (24-hr )data from the monte
;carlo files

OPENR, unit, filename_24_hr_ave_monte, /GET_LUN, ERROR = err
      ;****** Checking whether the file exists or not ****
      IF (err ne 0) then begin
         PRINT,'File Not Found for ', smonth, ' ', syear
       return
```

```
      ENDIF ELSE BEGIN
         PRINT,'Reading data' , filename_24_hr_ave_monte
      ENDELSE


READF,unit, header
      FOR i =0,header+2 DO BEGIN
        READF,unit,headerstr
      ENDFOR
READF,unit,data_24_ave_monte
CLOSE, /ALL

; **** Reading the 12-hr hourly average fit parameters
OPENR, unit, filename_12_hr_ave_monte, /GET_LUN, ERROR = err
      ;****** Checking whether the file exists or not ****
      IF (err ne 0) then begin
         PRINT,'File Not Found for ', smonth, ' ', syear
       return
      ENDIF ELSE BEGIN
         PRINT,'Reading data' , filename_12_hr_ave_monte
      ENDELSE


READF,unit, header
      FOR i =0,header+2 DO BEGIN
        READF,unit,headerstr
      ENDFOR
READF,unit,data_12_ave_monte
CLOSE, /ALL

; Constant for reading each parameter of hourly average into an array

mean_ave = fltarr(14)
mean_ave_dev = fltarr(14)
amp_24_ave = fltarr(14)
amp_24_ave_dev = fltarr(14)
pha_24_ave = fltarr(14)
pha_24_ave_dev = fltarr(14)
amp_12_ave = fltarr(14)
amp_12_ave_dev = fltarr(14)
pha_12_ave = fltarr(14)
pha_12_ave_dev = fltarr(14)
condnum_24_ave = fltarr(14)
condnum_12_ave = fltarr(14)
height_ave = fltarr(14)

; Constant for reading each parameter of monte carlo hourly average
;fits into an array

mean_ave_monte = fltarr(14)
mean_ave_dev_monte = fltarr(14)
amp_24_ave_monte = fltarr(14)
amp_24_ave_dev_monte = fltarr(14)
pha_24_ave_monte = fltarr(14)
```

```
pha_24_ave_dev_monte = fltarr(14)
amp_12_ave_monte = fltarr(14)
amp_12_ave_dev_monte = fltarr(14)
pha_12_ave_monte = fltarr(14)
pha_12_ave_dev_monte = fltarr(14)
condnum_24_ave_monte = fltarr(14)
condnum_12_ave_monte = fltarr(14)
height_ave_monte = fltarr(14)

; Each array of hourly average fits with its respective array values

height = data_12_ave[0,*]
condnum_24_ave = data_24_ave[7,*]
condnum_12_ave = data_12_ave[7,*]
tmp_24_ave = WHERE (condnum_24_ave lt 10, count)
tmp_12_ave = WHERE (condnum_12_ave lt 10, count)
hgt_24_ave = data_24_ave[0,tmp_24_ave]
mean_ave = data_12_ave[1,*]
mean_ave_dev = data_12_ave[2,*]
amp_24_ave = data_24_ave[3,tmp_24_ave]
amp_24_ave_dev = data_24_ave[4,tmp_24_ave]
pha_24_ave = data_24_ave[5,tmp_24_ave]
pha_24_ave_dev = data_24_ave[6,tmp_24_ave]
hgt_12_ave = data_12_ave[0,tmp_12_ave]
amp_12_ave = data_12_ave[3,tmp_12_ave]
amp_12_ave_dev = data_12_ave[4,tmp_12_ave]
pha_12_ave = data_12_ave[5,tmp_12_ave]
pha_12_ave_dev = data_12_ave[6,tmp_12_ave]
; Each array of monte carlo hourly average fits with its respective
;array values

height = data_12_ave_monte[0,*]
condnum_24_ave_monte = data_24_ave_monte[7,*]
condnum_12_ave_monte = data_12_ave_monte[7,*]
tmp_24_ave_monte = WHERE (condnum_24_ave_monte lt 10, count)
tmp_12_ave_monte = WHERE (condnum_12_ave_monte lt 10, count)
hgt_24_ave_monte = data_24_ave_monte[0,tmp_24_ave_monte]
mean_ave_monte = data_12_ave_monte[1,*]
mean_ave_dev_monte = data_12_ave_monte[2,*]
amp_24_ave_monte = data_24_ave_monte[3,tmp_24_ave_monte]
amp_24_ave_dev_monte = data_24_ave_monte[4,tmp_24_ave_monte]
pha_24_ave_monte = data_24_ave_monte[5,tmp_24_ave_monte]
pha_24_ave_dev_monte = data_24_ave_monte[6,tmp_24_ave_monte]
hgt_12_ave_monte = data_12_ave_monte[0,tmp_12_ave_monte]
amp_12_ave_monte = data_12_ave_monte[3,tmp_12_ave_monte]
amp_12_ave_dev_monte = data_12_ave_monte[4,tmp_12_ave_monte]
pha_12_ave_monte = data_12_ave_monte[5,tmp_12_ave_monte]
pha_12_ave_dev_monte = data_12_ave_monte[6,tmp_12_ave_monte]

;*** Plotting the DC- Amplitude for the raw, hourly average, month_day
;data

window,retain=2,xsize = 300, ysize =512
position = [0.2,0.1,0.75,0.95]
```

```
PLOT,[-60,60], height,yticks=14,ytickinterval=3, $
yrange=[min(height)-3,max(height)+6],ystyle=1,yminor=3, position =
position, $
ytitle = "Height (Kms)",title= 'DC ' +smonth+' '+syear+'zonal', $
xtitle = "velocity in m/s ",/NODATA, xticklen=0,xminor=4, color=0,
background=255

; Oplotting the actual hourly average mean data

OPLOT, mean_ave,height, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT, mean_ave,height, linestyle = 2,thick = 1.5, color = 60

;**** Plotting the deviation *******

mean_devpos_ave = [mean_ave]+([mean_ave_dev]/2)
mean_devneg_ave = [mean_ave]-([mean_ave_dev]/2)
      FOR i=0,N_ELEMENTS(height)-1 DO BEGIN

      OPLOT,[mean_devneg_ave[i],mean_devpos_ave[i]],[height[i],height[i
      ]],color= 60
      ENDFOR

tvlct,r,g,b,/get
image = tvrd()
; this line is used when we write the plot using the monte carlo error
;data
write_png,dir_plot+syear+'/montecarlo_ls_fit/u_'+syear2+'_'+smonth+'_DC
.png',image,R,G,B
;;write_png,dir_plot+syear+'/ls_fit_sltadjust/u_'+syear2+'_'+smonth+'_D
;C.png',image,R,G,B

; Plotting the standard deviations obtained from LS FIT and Monte Carlo
method
window,retain=2,xsize = 300, ysize =512
 position = [0.2,0.1,0.75,0.95]
 PLOT,[-5,5], height,yticks=14,ytickinterval=3, $
 yrange=[min(height)-3,max(height)+6],ystyle=1,xstyle=1,yminor=3,
position = position, $
 ytitle = "Height (Kms)",title= 'DC- deviation ' +smonth+'
'+syear+'zonal', $
 xtitle = "velocity in m/s ",/NODATA, xticklen=0,xminor=4, color=0,
background=255

  ; Oplotting the actual hourly average mean data deviation
 OPLOT, mean_ave_dev,height, PSYM = 4, thick = 1.5, COLOR = 60
 OPLOT, mean_ave_dev,height, linestyle = 2,thick = 1.5, color = 60

; Oplotting the monte carlo hourly average mean data deviation
 OPLOT, mean_ave_dev_monte,height, PSYM = 2, thick = 1.5, COLOR = 150
 OPLOT, mean_ave_dev_monte,height, linestyle = 4,thick = 1.5, color =
150
 xyouts,[100],[467.5],['- - - std. dev '], color =[60],
alignment=0,/DEVICE
```

```
xyouts,[100],[459],['- . . monte carlo '], color =[150],
alignment=0,/DEVICE
 tvlct,r,g,b,/get
 image = tvrd()
; this line is used when we write the plot using the monte carlo error
;data and actual std deviation obtained from SVD

write_png,dir_plot+syear+'/montecarlo_ls_fit/u_'+syear2+'_'+smonth+'_DC
_dev.png',image,R,G,B


;*** Plotting the 24-hr hourly average fits with the monte carlo errors

 WINDOW, retain=2
 !P.MULTI = [0,2,1]
 position = [0.1,0.1,0.35,0.95]
; Plotting the amplitudes
 PLOT,[0,30],height,yticks=14,ytickinterval=3, $
 yrange=[min(height)-3,max(height)+6],ystyle=1,yminor=3, position =
position, $
 ytitle = "Height (Kms)",title= '24-hr Amplitude for ' +smonth+'
'+syear+'zonal', $
 xtitle = "velocity in m/s ",xticklen=0,xminor=5, color=0,
background=255,/nodata

;****Oplotting the hourly average parameter
OPLOT,amp_24_ave,hgt_24_ave, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT,amp_24_ave,hgt_24_ave, linestyle = 2,thick = 1.5, color = 60

;**** Plotting the deviation *******
amp_devpos_24_ave = [amp_24_ave]+([amp_24_ave_dev]/2)
amp_devneg_24_ave = [amp_24_ave]-([amp_24_ave_dev]/2)

      FOR i=0,N_ELEMENTS(tmp_24_ave)-1 DO BEGIN

      OPLOT,[amp_devneg_24_ave[i],amp_devpos_24_ave[i]],[hgt_24_ave[i],
      hgt_24_ave[i]],color= 60
      ENDFOR

 position = [0.6,0.1,0.85,0.95]
; Plotting the Phases
PLOT,[-12,12],height,yticks=14,ytickinterval=3,
xticks=6,xstyle=1,xtickinterval=3, $
yrange=[min(height)-3,max(height)+6],ystyle=1,yminor=3, xminor = 3,
position = position, $
ytitle = "Height (Kms)",title= '24-hr Phase for ' +smonth+'
'+syear+'zonal', $
xtitle = "Phase ",xticklen=0, color=0, background=255,/nodata

;****Oplotting the hourly average parameter
OPLOT,pha_24_ave,hgt_24_ave, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT,pha_24_ave,hgt_24_ave, linestyle = 2,thick = 1.5, color = 60

;**** Plotting the deviation *******
```

```
pha_devpos_24_ave = [pha_24_ave]+([pha_24_ave_dev]/2)
pha_devneg_24_ave = [pha_24_ave]-([pha_24_ave_dev]/2)

       FOR i=0,N_ELEMENTS(tmp_24_ave)-1 DO BEGIN

       OPLOT,[pha_devneg_24_ave[i],pha_devpos_24_ave[i]],[hgt_24_ave[i],
       hgt_24_ave[i]],color= 60
       ENDFOR

;*** Writing the 24-hr parameters plot ****
tvlct,r,g,b,/get
image = tvrd()
; this line is used when we write the plot using the monte carlo error
;data
;
write_png,dir_plot+syear+'/montecarlo_ls_fit/u_'+syear2+'_'+smonth+'_24
-hr.png',image,R,G,B
;;
write_png,dir_plot+syear+'/ls_fit_sltadjust/u_'+syear2+'_'+smonth+'_24-
hr.png',image,R,G,B

;*** Plotting the 24-hr hourly average fits deviation with the monte
carlo errors

WINDOW, retain=2
!P.MULTI = [0,2,1]
position = [0.1,0.1,0.35,0.95]
; Plotting the amplitudes
PLOT,[-5,5],height,yticks=14,ytickinterval=3, $
yrange=[min(height)-3,max(height)+6],ystyle=1,xstyle=1,yminor=3,
position = position, $
ytitle = "Height (Kms)",title= '24-hr Amplitude deviations for '
+smonth+' '+syear+'zonal',$
xtitle = "velocity in m/s ",xticklen=0,xminor=5, color=0,
background=255,/nodata

;****Oplotting the hourly average parameter
OPLOT,amp_24_ave_dev,hgt_24_ave, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT,amp_24_ave_dev,hgt_24_ave, linestyle = 2,thick = 1.5, color = 60

;****Oplotting the hourly average monte carlo deviation parameter
OPLOT,amp_24_ave_dev_monte,hgt_24_ave,PSYM = 2, thick = 1.5, COLOR=60
OPLOT,amp_24_ave_dev_monte,hgt_24_ave, linestyle = 4,thick = 1.5, color
= 150
xyouts,[100],[467.5],['- - - std dev '], color =[60],
alignment=0,/DEVICE
xyouts,[100],[459],['- . . monte carlo'], color =[150],
alignment=0,/DEVICE

position = [0.6,0.1,0.85,0.95]
; Plotting the Phases
PLOT,[-10,10],height,yticks=14,ytickinterval=3,
xticks=6,xstyle=1,xtickinterval=3, $
```

```
yrange=[min(height)-3,max(height)+6],ystyle=1,yminor=3, xminor = 3,
position = position, $
ytitle = "Height (Kms)",title= '24-hr Phase deviations for ' +smonth+'
'+syear+'zonal', $
xtitle = "Phase ",xticklen=0, color=0, background=255,/nodata

;****Oplotting the hourly average parameter
OPLOT,pha_24_ave_dev,hgt_24_ave, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT,pha_24_ave_dev,hgt_24_ave, linestyle = 2,thick = 1.5, color = 60

;****Oplotting the hourly average monte carlo deviation parameter
OPLOT,pha_24_ave_dev_monte,hgt_24_ave_monte, PSYM = 2, thick = 1.5,
COLOR = 150
OPLOT,pha_24_ave_dev_monte,hgt_24_ave_monte, linestyle = 4,thick = 1.5,
color = 150
xyouts,[425],[467.5],['- - - std dev '], color =[60],
alignment=0,/DEVICE
xyouts,[425],[459],['- . . monte carlo'], color =[150],
alignment=0,/DEVICE

;*** Writing the 24-hr parameters plot ****
tvlct,r,g,b,/get
image = tvrd()
; this line is used when we write the plot using the monte carlo error
;data

write_png,dir_plot+syear+'/montecarlo_ls_fit/u_'+syear2+'_'+smonth+'_24
-hr_dev.png',image,R,G,B

;**** Plotting the 12-hr hourly average fit parameters with monte carlo
errors

!P.MULTI = [0,2,1]
position = [0.1,0.1,0.35,0.95]
;*** Plotting the 12-hr amplitudes of raw,hourly,month_day,GSWM
parameters

PLOT,[0,30],height,yticks=14,ytickinterval=3, $
yrange=[min(height)-3,max(height)+6],ystyle=1,yminor=3, position =
position, $
ytitle = "Height (Kms)",title= '12-hr Amplitude for ' +smonth+'
'+syear+'zonal', $
xtitle = "velocity in m/s ",xticklen=0,xminor=5, color=0,
background=255,/nodata

;****Oplotting the hourly average parameter
OPLOT,amp_12_ave,hgt_12_ave, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT,amp_12_ave,hgt_12_ave, linestyle = 2,thick = 1.5, color = 60

;**** Plotting the deviation *******
amp_devpos_12_ave = [amp_12_ave]+([amp_12_ave_dev]/2)
amp_devneg_12_ave = [amp_12_ave]-([amp_12_ave_dev]/2)

        FOR i=0,N_ELEMENTS(tmp_12_ave)-1 DO BEGIN
```

```
      OPLOT,[amp_devneg_12_ave[i],amp_devpos_12_ave[i]],[hgt_12_ave[i],
      hgt_12_ave[i]],color= 60
      ENDFOR

position = [0.6,0.1,0.85,0.95]
; Plotting the Phases of 12-hr raw,hourly,month_day,GSWM parameters
PLOT,[-6,6],height,yticks=14,ytickinterval=3, $
yrange=[min(height)-3,max(height)+6],ystyle=1,yminor=3, xstyle
=1,xminor =2,position = position, $
ytitle = "Height (Kms)",title= '12-hr Phase for ' +smonth+'
'+syear+'zonal', $
xtitle = "Phase ",xticklen=0, color=0, background=255,/nodata

;****Oplotting the hourly average parameter
OPLOT,pha_12_ave,hgt_12_ave, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT,pha_12_ave,hgt_12_ave, linestyle = 2,thick = 1.5, color = 60

;**** Plotting the deviation *******
pha_devpos_12_ave = [pha_12_ave]+([pha_12_ave_dev]/2)
pha_devneg_12_ave = [pha_12_ave]-([pha_12_ave_dev]/2)

      FOR i=0,N_ELEMENTS(tmp_12_ave)-1 DO BEGIN

      OPLOT,[pha_devneg_12_ave[i],pha_devpos_12_ave[i]],[hgt_12_ave[i],
      hgt_12_ave[i]],color= 60
      ENDFOR

;*** Writing the 12-hr parameters plot ***
tvlct,r,g,b,/get
image = tvrd()
; this line is used when we write the plot using the monte carlo error
;data
write_png,dir_plot+syear+'/montecarlo_ls_fit/u_'+syear2+'_'+smonth+'_12
-hr.png',image,R,G,B

;**** Plotting the 12-hr hourly average fit parameters with monte carlo
errors

!P.MULTI = [0,2,1]
position = [0.1,0.1,0.35,0.95]

PLOT,[-5,5],height,yticks=14,ytickinterval=3, $
yrange=[min(height)-3,max(height)+6],ystyle=1,xstyle=1,yminor=3,
position = position, $
ytitle = "Height (Kms)",title= '12-hr Amplitude deviations for '
+smonth+' '+syear+'zonal', $
xtitle = "velocity in m/s ",xticklen=0,xminor=5, color=0,
background=255,/nodata

;****Oplotting the hourly average parameter
OPLOT,amp_12_ave_dev,hgt_12_ave, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT,amp_12_ave_dev,hgt_12_ave, linestyle = 2,thick = 1.5, color = 60
```

```
;****Oplotting the hourly average monte carlo deviation parameter
OPLOT,amp_12_ave_dev_monte,hgt_12_ave_monte, PSYM = 2, thick = 1.5,
COLOR = 150
OPLOT,amp_12_ave_dev_monte,hgt_12_ave_monte, linestyle = 4,thick = 1.5,
color = 150
xyouts,[100],[467.5],['- - - std dev '], color =[60],
alignment=0,/DEVICE
xyouts,[100],[459],['- . . monte carlo'], color =[150],
alignment=0,/DEVICE

position = [0.6,0.1,0.85,0.95]
; Plotting the Phase of 12-hr hourly parameters
PLOT,[-6,6],height,yticks=14,ytickinterval=3, $
yrange=[min(height)-3,max(height)+6],ystyle=1,yminor=3, xstyle
=1,xminor =2,position = position, $
ytitle = "Height (Kms)",title= '12-hr Phase deviations for ' +smonth+'
'+syear+'zonal', $
xtitle = "Phase ",xticklen=0, color=0, background=255,/nodata

;****Oplotting the hourly average deviation parameter
OPLOT,pha_12_ave_dev,hgt_12_ave, PSYM = 4, thick = 1.5, COLOR = 60
OPLOT,pha_12_ave_dev,hgt_12_ave, linestyle = 2,thick = 1.5, color = 60

;****Oplotting the hourly average monte carlo deviation parameter
OPLOT,pha_12_ave_dev_monte,hgt_12_ave_monte, PSYM = 2, thick = 1.5,
COLOR = 150
OPLOT,pha_12_ave_dev_monte,hgt_12_ave_monte, linestyle = 4,thick = 1.5,
color = 150
xyouts,[425],[467.5],['- - - std dev '], color =[60],
alignment=0,/DEVICE
xyouts,[425],[459],['- . . monte carlo'], color =[150],
alignment=0,/DEVICE

;*** Writing the 12-hr parameters plot ***
tvlct,r,g,b,/get
image = tvrd()
; this line is used when we write the plot using the monte carlo error
;data

write_png,dir_plot+syear+'/montecarlo_ls_fit/u_'+syear2+'_'+smonth+'_12
-hr_dev.png',image,R,G,B
;write_png,dir_plot+syear+'/ls_fit_sltadjust/u_'+syear2+'_'+smonth+'_12
-hr.png',image,R,G,B

!P.MULTI=0
END

**** Code used for Lomb-Scargle Periodogram method

PRO test_data

; This is used to create the test data which can be used for every
program
LOADCT,39
```

```
WINDOW,xsize=700,ysize=850,retain=2 ; used to set the size of the plot
!P.MULTI = [0,2,3,0,1]
pi = !DPI
num = 744
t = indgen(num)
n = n_elements(t)
theta_12 = 0
theta_24 = 0
theta_8 = 0
;data_t =
sin(2*pi*t/12+theta_12);+sin(2*pi*t/24+theta_24)+sin(2*pi*t/8+theta_8)
 FOR dt_sel = 0,1 DO BEGIN
  IF (dt_sel EQ 0) THEN BEGIN
      data_t = 5*sin(2.0*pi*t/12.0)  ;+5*sin(2*pi*t/12)
      PRINT,'The data signal is data_t = 5*sin(2.0*pi*t/12.0)'
  ENDIF ELSE BEGIN
      data_t = 5*sin(2.0*pi*t/24.0)
      PRINT,'The data signal is data_t = 5*sin(2.0*pi*t/24.0)'
  ENDELSE
sz = size(data_t,/n_elements)
t_interval = 1.0  ; interval
N21 = N/2 +1
f = indgen(N)
f[n21] = n21-n+findgen(n21-2)
f=f/(n*t_interval)   ; compute to freq.
;*** taking the absolute of frequencies to have only positive freqs.
indexfreq = WHERE(f GT 0,cnt)
posfreq = f[indexfreq]
;*** calculating the freq's of interest
freq = fltarr(num)
pnow = 1.0/(num-1)  ;767
 FOR i=0,(num-1) DO BEGIN
      freq[i] = pnow
      pnow = pnow+1.0/(num-1)
 ENDFOR

 result = fft(data_t,/double)
 result_freq = result[indexfreq] ;taking the values of FFT's respective
to the positive freqs

charsz = 2.25
plot,[0,30],[0,3],color=0,background = 255, charsize=charsz, $
title='FFT- data = 5*sin(2*pi*t/24) for 744hrs ', $
xtitle='1/freq (hours)',ytitle='amplitude',/nodata
OPLOT,1/posfreq,abs(result_freq),color=0
tvlct,r,g,b,/get
image=tvrd()
write_png,'/home/vemula/idl/platmf/test_data_plots/nov2nd2004/fft_24_74
4_random.png',image,r,g,b

; this part of Lomb-Scargle is used without masking of the test data
 res = my_lomb_scargle(t,data_t,sz,lomb_freq_100,lomb_pow_100);used to
get all plots in one page
```

```
; this part is used for masking the test data

rnd = randomn(0,[744],/uniform)
mask = WHERE (rnd LT 0.765, count) ;0.286=221points; 80% of data = LT
0.765 ; 60% of data=LT 0.588

num_hr = n_elements(mask)
n_hrs = STRTRIM(STRING(num_hr),2)
datamask_fft = fltarr(744)
datamask_fft[*] = 0
datamask_fft[mask] = data_t[mask]
result_datamask = fft(datamask_fft,/double)
result_datamaskfreq = result_datamask[indexfreq]
renorm = ((abs(result_datamaskfreq)*744.0)/num_hr) ; renormalizing the
power

plot,[0,30],[0,3],color=0,background = 255, charsize=charsz,$
title='FFT- datamask = 5*sin(2*pi*t/24) for '+n_hrs+'(744hrs) with rnd
data masked-zeros', $
xtitle='1/freq (hours)',ytitle='amplitude',xticklen=-0.02,/nodata
oplot,1/posfreq,abs(result_datamaskfreq),color=0
OPLOT,1/posfreq,renorm,color=150,linestyle=2
tvlct,r,g,b,/get
image = tvrd()

write_png,'/home/vemula/idl/platmf/test_data_plots/nov2nd2004/fft_24_'+
n_hrs+'(744)_random.png', image,r,g,b

datamask_lomb = data_t[mask]
hours_lomb = t[mask]
sz = size(datamask_lomb,/n_elements)

res =
my_lomb_scargle(hours_lomb,datamask_lomb,sz,lomb_freq_80,lomb_pow_80);u
sed to get all plots in one page

;*** this part is used for masking the test data for 30% of data

rnd_30 = randomn(0,[744],/uniform)
mask_30 = WHERE (rnd_30 LT 0.286, count) ;0.286=221points; 80% of data
= LT 0.765 ; 60% of data=LT 0.588
num_hr_30 = n_elements(mask_30)
n_hrs_30 = STRTRIM(STRING(num_hr_30),2)
datamask_fft_30 = fltarr(744)
datamask_fft_30[*] = 0
datamask_fft_30[mask_30] = data_t[mask_30]
result_datamask_30 = FFT(datamask_fft_30,/double)
result_datamaskfreq_30 = result_datamask_30[indexfreq]
renorm_30 = ((abs(result_datamaskfreq_30)*744.0)/num_hr_30) ;
renormalizing the power
plot,[0,30],[0,3],color=0,background = 255, charsize=charsz,$
title='FFT- datamask = 5*sin(2*pi*t/24) for '+n_hrs_30+'(744hrs) with
rnd data masked-zeros', $
```

```
xtitle='1/freq (hours)',ytitle='amplitude',xticklen=-0.02,/nodata
oplot,1/posfreq,abs(result_datamaskfreq_30),color=0
OPLOT,1/posfreq,renorm_30,color=150,linestyle=2
tvlct,r,g,b,/get
image = tvrd()

write_png,'/home/vemula/idl/platmf/test_data_plots/nov2nd2004/fft_24_'+
n_hrs+'(744)_random.png', image,r,g,b

datamask_lomb_30 = data_t[mask_30]
hours_lomb_30 = t[mask_30]
sz = size(datamask_lomb_30,/n_elements)

res =
my_lomb_scargle(hours_lomb_30,datamask_lomb_30,sz,lomb_freq_30,lomb_pow
_30);used to get all plots in one page

charsz = 2.25
PLOT,[0,30],[0,3],color=0,background = 255, charsize = charsz, $
xtitle='1/freq (hours)',ytitle='amplitude',/nodata
OPLOT,1/lomb_freq_100,lomb_pow_100,color=0

;*** To plot the 80% hours of data Lomb Scargle
PLOT,[0,30],[0,3],color=0,background = 255, charsize = charsz, $
xtitle='1/freq (hours)',ytitle='amplitude',/nodata
OPLOT,1/lomb_freq_80,lomb_pow_80,color=0

;*** To plot the 30% hours of data Lomb Scargle
PLOT,[0,30],[0,3],color=0,background = 255, charsize = charsz, $
xtitle='1/freq (hours)',ytitle='amplitude',/nodata
OPLOT,1/lomb_freq_30,lomb_pow_30,color=0

ENDFOR

tvlct,r,g,b,/get
image=tvrd()

write_png,'/home/vemula/idl/platmf/test_data_plots/nov19th2004/lomb_ran
dom.png',image,r,g,b


END

********

PRO testdata_periodicgaps

; This is used to create the test data and selected hours of data are
;processed
; to obtain the FFT and Lomb Periodograms which can be used for every
;program
; Also, a renormalizing factor of N/n is applied(N=total no: of data
;points,n=no: of real data)
```

```
WINDOW,xsize=700,ysize=850,retain=2 ; used to set the size of the plot
!P.MULTI = [0,2,3,0,1]
loadct,39
pi = !DPI
num = 744
FOR dt_sel = 0,1 DO BEGIN
   IF (dt_sel EQ 0) THEN BEGIN
      t = indgen(24,31)  ;findgen(24,31)  ;indgen(24*31)
      time_mask = fltarr(24,31)
      N = n_elements(t)
      t_reform = reform(t,1,744)
      data_t = 5*sin(2.0*pi*t_reform/12.0)
;+5*sin(2.0*pi*t_reform/24.0)
      PRINT,'The data are data_t = 5*sin(2.0*pi*t_reform/12.0)'
   ENDIF ELSE BEGIN
      t = indgen(24,31)  ;findgen(24,31)  ;indgen(24*31)
      time_mask = fltarr(24,31)
      N = n_elements(t)
      t_reform = reform(t,1,744)
      data_t = 5*sin(2.0*pi*t_reform/24.0)
      PRINT,'The data are data_t = 5*sin(2.0*pi*t_reform/24.0)'
   ENDELSE
theta_12 = 0
theta_24 = 0
theta_8 = 0
sz = size(data_t,/n_elements)
freq = fltarr(num)
pnow = 1.0/(num-1)  ;767
 FOR i=0,(num-1) DO BEGIN
      freq[i] = pnow
      pnow = pnow+1.0/(num-1)

 ENDFOR

;*** Calculating the frequencies of respectives FFT's
t_interval = 1.0  ; interval
N21 = N/2 +1
f = indgen(N)
f[n21] = n21-n+findgen(n21-2)
f=f/(n*t_interval)   ; compute to freq.

;*** taking the absolute of frequencies to have only positive freqs.
for FFT plots
indexfreq = WHERE(f GT 0,cnt)
posfreq = f[indexfreq]        ; taking the positive freq's

;*** This part is used for 24 hrs of data

time_mask[0:23,*] = t[0:23,*]
t_mask = reform(time_mask,1,744)

;*** for calculating the FFT ***

datamask_24_fft = fltarr(744)
```

```
datamask_24_fft = data_t
result_24 = FFT(datamask_24_fft,/double)
result_24_freq = result_24[indexfreq]
result_24_norm = (abs(result_24_freq)*744.0)/744.0

PLOT,[0,30],[0,4],color=0,background = 255, charsize = charsz, $
title='FFT- data = 5*sin(2*pi*t/24) for 744 of 744hrs(defined) ', $
xtitle='1/freq (hours)',ytitle='amplitude',xticklen=-0.02,/nodata
OPLOT,1/posfreq,abs(result_24_freq),color=0
OPLOT,1/posfreq, result_24_norm, color=150, linestyle=2
tvlct,r,g,b,/get
image=tvrd()
write_png,'/home/vemula/idl/platmf/test_data_plots/nov4th2004/fft_12_74
4(744).png',image,r,g,b

datamask_24 = data_t

;*** The masked hours of data are used to obtain the LombScargle
periodograms

sz_24 = size(datamask_24,/n_elements)
power_12 = fltarr(6)
power_24 = fltarr(6)
res =
my_lomb_scargle(t_mask,datamask_24,sz_24,lomb_freq_24,lomb_pow_24,pow_1
2,pow_24)

;*** this part is for 16 hrs of data ***

time_mask_16 = fltarr(24,31)
time_mask_16[4:19,*] = t[4:19,*]
tm_mask_16 = reform(time_mask_16,1,744)
mask_16 = WHERE(tm_mask_16 ne 0 , count)
t_mask = t_reform[mask_16]

;*** for calculating the FFT ***

datamask_16_fft = fltarr(744)
datamask_16_fft[mask_16] = data_t[mask_16]
result_16 = FFT(datamask_16_fft,/double)
result_16_freq = result_16[indexfreq]
result_16_norm = (abs(result_16_freq)*744.0)/496

PLOT,[0,30],[0,4],color=0,background = 255, charsize = charsz, $
title='FFT- data = 5*sin(2*pi*t/24) for 496 of 744hrs(defined) ', $
xtitle='1/freq (hours)',ytitle='amplitude',xticklen=-0.02,/nodata
OPLOT,1/posfreq,abs(result_16_freq),color=0
OPLOT,1/posfreq, result_16_norm, color=150, linestyle=2
tvlct,r,g,b,/get
image=tvrd()
write_png,'/home/vemula/idl/platmf/test_data_plots/nov4th2004/fft_12_49
6(744)_defined.png',image,r,g,b

datamask_16 = data_t[mask_16]
```

```
;*** The masked hours of data are used to obtain the LombScargle
periodograms

sz_16 = size(datamask_16,/n_elements)
res =
my_lomb_scargle(mask_16,datamask_16,sz_16,lomb_freq_16,lomb_pow_16,pow_
12,pow_24)

;; power_12[2] = pow_12
;; power_24[2] = pow_24

;*** this part is used for 8hrs of data ****
 time_mask_8 = fltarr(24,31)
 time_mask_8[8:15,*] = t[8:15,*]
 tm_mask_8 = REFORM(time_mask_8,1,744)
 mask_8 = WHERE(tm_mask_8 ne 0 , count)
 t_mask = t_reform[mask_8]

;*** for calculating the FFT ***

 datamask_8_fft = fltarr(744)
 datamask_8_fft[mask_8] = data_t[mask_8]
 result_8 = FFT(datamask_8_fft,/double)
 result_8_freq = result_8[indexfreq]
 result_8_norm = (abs(result_8_freq)*744.0)/248

PLOT,[0,30],[0,4],color=0,background = 255, charsize = charsz, $
title='FFT- data = 5*sin(2*pi*t/24) for 248 of 744hrs(defined) ', $
xtitle='1/freq (hours)',ytitle='amplitude',/nodata
OPLOT,1/posfreq,abs(result_8_freq),color=0
OPLOT,1/posfreq, result_8_norm, color=150, linestyle=2
tvlct,r,g,b,/get
image=tvrd()
write_png,'/home/vemula/idl/platmf/test_data_plots/nov4th2004/fft_12_24
8(744)_defined.png',image,r,g,b
datamask_8 = data_t[mask_8]

;*** The masked hours of data are used to obtain the FFT and
LombScargle periodograms
sz_8 = size(datamask_8,/n_elements)
power_12 = fltarr(6)
power_24 = fltarr(6)
 res =
my_lomb_scargle(mask_8,datamask_8,sz_8,lomb_freq_8,lomb_pow_8,pow_12,po
w_24)

;; power_12[0] = pow_12
;; power_24[0] = pow_24
charsz = 2.25
PLOT,[0,30],[0,4],color=0,background = 255, charsize = charsz, $
xtitle='1/freq (hours)',ytitle='amplitude',/nodata
OPLOT,1/lomb_freq_24,lomb_pow_24,color=0
```

```
;*** To plot the 16 hours of data Lomb Scargle
PLOT,[0,30],[0,4],color=0,background = 255, charsize = charsz, $
xtitle='1/freq (hours)',ytitle='amplitude',/nodata
OPLOT,1/lomb_freq_16,lomb_pow_16,color=0

;*** To plot the 8 hours of data Lomb Scargle
PLOT,[0,30],[0,4],color=0,background = 255, charsize = charsz, $
xtitle='1/freq (hours)',ytitle='amplitude',/nodata
OPLOT,1/lomb_freq_8,lomb_pow_8,color=0
ENDFOR

END

********

FUNCTION my_lomb_scargle,x,y,n,lomb_freq,lomb_pow,pow_12,pow_24,jmax
; This routine has been written based on the routine from Numerical
recipes in fortraninteger
; Given n data points and x,y are the time and the data arrays
;INTEGER,jmax,n,nout,np,NMAX
;REAL,hifac,ofac,prob,px(np),py(np),x(n),y(n)

NMAX = n ;768      ;744  ; 8928 is used for the raw data since this is
the max. data points for raw data
TWOPID = 2*!DPI    ;6.283185307179586
ofac =      4      ; 0.5
fhi = 0.5          ;0.1
T = max(x)-min(x)
fc = n/(2.0*T)
hifac = fhi/fc
nout = 0.5*ofac*hifac*n
n_hrs = STRTRIM(STRING(n),2)  ; string of numbers of data hours

; Constants Used

wpr = DBLARR(NMAX)
wr = DBLARR(NMAX)
wpi = DBLARR(NMAX)
wi = DBLARR(NMAX)
px = fltarr(nout)
py = fltarr(nout)
py_var = fltarr(nout)
lomb_freq = fltarr(nout)
lomb_pow = fltarr(nout)
ave = MEAN(y,/DOUBLE,/NaN)
var = variance(y,/DOUBLE,/NaN)
xmax = x(1)
xmin = x(1)
 FOR j=0,n-1 DO BEGIN
      IF (x(j) gt xmax) THEN BEGIN
      xmax=x(j)
      ENDIF ELSE BEGIN
       IF (x(j) LT xmin) THEN BEGIN
        xmin = x(j)
```

```
        ENDIF
       ENDELSE
 ENDFOR

xdif = xmax-xmin
xave = 0.5*(xmax+xmin)
pymax = 0.0
pnow = 1./(xdif*ofac)
  FOR j=0,n-1 DO BEGIN
      arg = TWOPID*((x(j)-xave)*pnow)
      wpr(j) = -2.d0*sin(0.5d0*arg)^2
      wpi(j) = sin(arg)
      wr(j) = cos(arg)
      wi(j) = wpi(j)
  ENDFOR
  FOR i = 0,nout-1 DO BEGIN
      px(i) = pnow

      sumsh = 0
      sumc = 0
      FOR j=0,n-1 DO BEGIN
        c = wr(j)
        s = wi(j)
        sumsh = sumsh+s*c
        sumc = sumc+(c-s)*(c+s)
      ENDFOR
      wtau = 0.5*atan(2.*sumsh,sumc)
      swtau = sin(wtau)
      cwtau = cos(wtau)
      sums = 0.
      sumc = 0.
      sumsy = 0.
      sumcy = 0.
      FOR j = 0,n-1 DO BEGIN
        s = wi(j)
        c = wr(j)
        ss = s*cwtau-c*swtau
        cc = c*cwtau+s*swtau
        sums = sums+ss^2
        sumc = sumc+cc^2
        yy = y(j)-ave
        sumsy = sumsy+yy*ss
        sumcy = sumcy+yy*cc
        wtemp = wr(j)
        wr(j) = (wr(j)*wpr(j)-wi(j)*wpi(j))+wr(j)
        wi(j) = (wi(j)*wpr(j)+wtemp*wpi(j))+wi(j)
      ENDFOR

      py(i) = 0.5*(sumcy^2/sumc+sumsy^2/sums)    ;/var
      IF ((py(i) GT pymax) OR (py(i) EQ pymax)) THEN BEGIN
        pymax = py(i)
        jmax = i
      ENDIF
      pnow = pnow+1./(xdif*ofac)
```

```
ENDFOR

print,var
py_var = py/var
expy = exp(-(DOUBLE(pymax/var)))
effm = 2.*nout/ofac     ; estimate of number of independent frequencies
prob = effm*expy

IF (prob GT 0.01) THEN BEGIN
      prob =1.-(1.-expy)^effm
ENDIF

;*** These are the freq's and powers returned to main pro from Lomb
Scargle
 lomb_freq = px
 lomb_pow = SQRT((py/n))
;*** This is used to extract the power of the 24hr and 12hr peaks
RETURN, prob

END
********
```