Electromagnetic Bias in Geosat Altimetry

by

Anne Marie Slinn

Submitted on October 9, 1990 in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering

ABSTRACT

Oceanographic satellites provide novel information about the world’s oceans. Satellite-borne radar altimeters produce an accurate measure of ocean wave heights, surface wind speeds, and the mean sea level to within a few centimeters. From these measurements information about current velocities and ocean circulation can be inferred. The largest remaining limitation to the accuracy of altimetric measurements, other than errors in the calculation of the satellite’s ephemeris, which are relatively easy to correct, is the electromagnetic (EM) bias.

The discrepancy between the mean reflecting surface of the ocean and the true mean sea level causes satellite altimeters to incorrectly estimate the sea level by a small amount, termed the EM bias. Direct measurements of the EM bias, during the SAXON–CLT experiment, produced an algorithm which estimates the EM bias as a function of the significant wave height, and the radar cross section (an altimetric measurement related to the wind speed). The SAXON bias algorithm is applied to the first year of Geosat data in the North Atlantic, and the resulting magnitudes, variations, and significance of the EM bias predictions are investigated.

Across the Gulf Stream it was found that the EM bias can contribute roughly 10% of the inferred geostrophic current velocities. In large winter storms the magnitude of the EM bias can reach up to a meter, although in such storms it was necessary to extrapolate the algorithm outside the range of direct measurements on which it was based. Repeat cycle (17-day) averages of the bias were determined and are presented in the form of color contour maps. Large basin-scale variations in the bias are evident, as well as a significant seasonal variation in the upper latitudes.

Thesis Supervisor: W. Kendall Melville
Title: Professor of Civil Engineering
# TABLE OF CONTENTS

Abstract...........................................................................................................................2  
Table of Contents...........................................................................................................3  
Chapter I. Introduction..............................................................................................4  
  1.1 Radar Altimetry ..................................................................................................5  
  1.2 Electromagnetic Bias .........................................................................................9  
  1.3 Geosat ...............................................................................................................13  
Chapter II. Geosat Data Processing ..........................................................................16  
  2.1 JPL Data Tapes ................................................................................................16  
  2.2 Editing and Averaging .......................................................................................17  
  2.3 Orbit Error Removal .........................................................................................19  
  2.4 Repeat Cycle Map Processing .........................................................................20  
Chapter III. Evaluation of EM Bias..................................................................21  
  3.1 In The Gulf Stream ...........................................................................................21  
  3.2 Near Storms ......................................................................................................26  
  3.3 Repeat Cycle Averages .....................................................................................37  
Chapter IV. Conclusion.........................................................................................62  
References..................................................................................................................65  
Acknowledgments........................................................................................................68  
Appendix A. Data Processing Programs.................................................................69
CHAPTER I. INTRODUCTION

The ability of oceanographic satellites to obtain global coverage of the oceans in a matter of weeks, and make observations at scales ranging from meters to entire basins is a potentially revolutionary improvement over the classical methods of ship-based and mooring-based measurements. Satellite-borne altimeters can directly measure the height of the satellite above the sea, and indirectly measure the waves, the wind, and the ocean currents. Determining current systems and velocities by the use of satellite altimetry promises to be a significant contribution to the study of ocean circulation and air-sea interaction.

This thesis provides some background on the fundamentals of radar altimetry, and the electromagnetic (EM) bias, which is a major source of error in altimetry measurements. The Geosat altimetric satellite is described, and an algorithm, which predicts the EM bias, is applied to Geosat data from the North Atlantic. The magnitude and significance of the predicted EM bias are studied in specific cases for which the bias may contribute a large error in altimetric measurements, and the seasonal variation of the EM bias is investigated.

First, in the Gulf Stream region, with its large current velocities and gradients, the contribution of the EM bias errors in the geostrophic velocity is estimated. Next, near two large storms, a summer (tropical) hurricane, and a winter extratropical storm, the change in magnitude of the EM bias is evaluated. And finally, the seasonal variability of the EM bias in the North Atlantic, during the first 14 months of Geosat, is presented using 21 contour maps of the bias, each map representing an average over a 17-day period.
1.1 Radar Altimetry

The radar altimeter works by transmitting a sharp pulse of radio waves at nadir to the ocean surface, and receiving the signal scattered back. The height of the altimeter above the ocean surface is measured by the time elapsed between transmitting a pulse, and receiving the return signal (Figure 1.1). Because the ocean surface is not flat, but covered with waves, the first return of a pulse comes from the wave crests. The duration of the rising return pulse then gives a measure of the distance between the wave crests and troughs, or the wave height. The maximum return power gives a measure of the roughness of the ocean surface. Because wind is related to the roughness of the ocean surface, the backscattered power can be used as an indirect measure of the surface wind speed.

Remote sensing methods have revolutionized our ability to observe the oceans. A satellite can achieve global coverage of the oceans in a matter of weeks, and make observations at scales ranging from meters to entire ocean basins. Perhaps the most important application of satellite altimetry is the measurement of ocean currents. Information about currents can be obtained from the measure of the height of the satellite above the ocean surface.

The ocean surface largely conforms to a gravitational equipotential surface, termed the geoid (Figure 1.2). Variations in the earth’s mass distribution cause the geoid to vary by approximately ± 100 m worldwide [Apel, 1987]. Ocean currents, tides, and atmospheric effects cause the ocean surface to deviate from the geoid. The departure of the ocean surface from the geoid, after correcting for tidal and atmospheric effects, is termed the sea surface topography. Strong current systems, such as the Gulf Stream and the Kuroshio, cause the sea surface topography to vary by roughly a meter [Pedlosky, 1987].
Figure 1.1. Radar altimeters determine the distance between the satellite and the sea surface from the elapsed time between transmitting and receiving a radar pulse. The significant wave height (SWH) is determined from the shape, or rise time, of the returned signal, and the magnitude of the returned signal power is used to infer the surface wind speed.

Figure 1.2. Measurement of sea surface topography, $\eta$, relative to geoidal heights. Satellite position with respect to center of mass, $R$, comes from orbit determination; height above sea level, $h$, derives from altimeter, and the difference gives surface elevation. The shape of the surface is due to variations in bottom topography, which produce geoid undulations, and to ocean currents, which produce the sea surface topography. The reference ellipsoid is the best smooth approximation to the geoid. [Adapted from Apel, 1987.]
The sea level is determined by differencing the height of the satellite above the sea surface, as measured by the satellite-borne radar altimeter, and the height of the satellite above the center of mass of the Earth, as measured by satellite tracking systems. With reasonable assumptions, the ocean current systems can be inferred using the geostrophic approximation, which balances the Coriolis acceleration and the slope of the sea surface topography (or pressure gradient). Under the geostrophic approximation, the slope of the sea surface relative to the geoid is linearly related to the magnitude of the surface current velocity [Apel, 1987]. The geostrophic equations are given by

\[
-f_v = -\frac{1}{\rho} \frac{\partial p'}{\partial x} = -g \frac{\partial \eta}{\partial x}
\]

\[
+f_u = -\frac{1}{\rho} \frac{\partial p'}{\partial y} = -g \frac{\partial \eta}{\partial y}
\]

where \( u_g \) and \( v_g \) are the velocity components in the \( x \) (East) and \( y \) (North) directions, \( f \) is the Coriolis parameter, \( f = 2\Omega \sin(lat) \), \( \Omega \) is the rotation rate of the earth, \( \rho \) is the water density, \( p' \) is the perturbation pressure, and \( \eta \) is the surface elevation. The surface current velocity can be thus be determined from the slope of the sea surface, as shown in Figure 1.3. Considering the fact that the largest ocean current systems cause the sea surface topography to vary by roughly a meter at most, it is clear that the measurements of sea level must be precise within a few centimeters.

The largest source of error in measuring the sea level is the inaccuracies of the ephemeris, or the position of the satellite with respect to the center of the Earth. Fortunately, this radial orbit error is easily removed on a regional basis, because the error is of a very large wavelength, equal to the satellite orbital circumference, which is much larger than the scales of interest in most studies. Other sources of error, termed the environmental path delays, are
caused by the gases (especially water vapor) in the troposphere, and the free electrons in the ionosphere affecting the propagation speed of the radar signal. These phenomena are well understood, and with independent measurements of the ionosphere and troposphere, the path delay errors can be accurately removed. The largest remaining source of error in measuring the sea level, and hence the determining the current velocities, is the electromagnetic bias [Melville et al., 1990].

Figure 1.3. The slope of the sea surface relative to the geoid \( (\partial \eta / \partial x) \) is directly related to surface geostrophic current \( v_s \). The slope of 1 m/100 km is typical of western boundary currents, such as the Gulf Stream. [Adapted from Stewart, 1985.]
1.2 Electromagnetic Bias

Owing to the fact that the trough of an ocean wave is in general *smoother* than the wave crest, with less small-scale roughness caused by wind, and *flatter*, with a larger local radius of curvature, the troughs tend to be better reflectors than the crests [Arnold, et al., 1989]. Hence, the mean sea level measured by radar altimeters is lower than the actual sea level, and this downward shift is referred to as the electromagnetic (EM) bias. Because the reflectivity of the ocean surface is related to the displacement of the surface from its mean value, the EM bias error is intrinsically correlated with the sea-state. Furthermore, because the sea-state is affected by the interactions between waves and current systems, the EM bias is also correlated with the spatial and temporal gradients in ocean currents. Therefore, the determination of the EM bias error is essential before ocean currents can be accurately resolved by satellite altimeters. A number of attempts have been made, using both direct and indirect methods, to determine the dependence of the bias on sea-state parameters.

Direct observation of EM bias began with the platform-based measurements by Yaplee et al. [1971]. Using a 10 GHz radar, measurements of the distance to the sea surface, and the radar reflectivity of the surface were made. At the same time, three wave poles, which surrounded the area observed by the radar, independently measured the wave height. Using this data, Jackson [1979] compared the distribution of the sea surface elevation, and the radar reflectivity as a function of deviation from the mean sea level. He concluded that the EM bias was approximately 5% of the significant wave height (SWH). The SWH (also referred to as $H_{1/3}$) is defined as the average of the heights of the highest one-third of the waves, or approximately four times the standard deviation of the wave height distribution [Apel, 1987].
Radars carried on low-flying aircraft were used in a number of later studies [Walsh et al., 1984; Choy et al., 1984; Hoge et al., 1984]. These studies showed the bias was a function of radar frequency, and ranged from 1% to 5% of SWH at 10 GHz. These studies also investigated the possible dependence of the EM bias on such parameters as the wavelength and wave slope, the wind speed, and properties of the sea surface displacement distributions (skewness and kurtosis). Although no definite correlations were apparent, the airborne radar studies have been considered inconclusive due to experimental errors caused by aircraft motion, and the reported differences in measured values for nearly identical conditions.

A recent platform-based study by Melville et al., [1990] during the Saxon-CLT experiment [Shemdin and McCormick, 1988] obtained direct measurements of the EM bias using a 14 GHz coherent scatterometer, which differs from a radar only in that it measures the return power scattered from a surface, but not the distance to the surface. Simultaneous measurements of the wave heights, environmental conditions, and returned power were recorded, and the EM bias were found to be a function of both SWH and wind speed. Since satellite altimeters infer the wind speed from a measurement of the backscattered power per unit area, hereafter referred to as \( \sigma_0 \), Melville et al. [1990] produced an algorithm correlating the bias with SWH and \( \sigma_0 \), given by

\[
B = (-0.0163 - 2.15 \sigma_0^2 - 0.00291 H_{1/3}) H_{1/3}
\]

with bias in m, \( \sigma_0 \) in dB, and \( H_{1/3} \) (or SWH) in m. The range of data collected during the Saxon experiment included hourly averages of SWH from 0.3 m to 2.9 m, wind speeds from 0.2 m/s to 15.3 m/s, and \( \sigma_0 \) values between approximately 9 and 17 dB. The resulting bias values they obtain are within the range of previously reported measurements.
Altimetric satellite measurements have also been used to predict the EM bias. At a given location, repeated measurements of the sea level are compared, and the temporal variability of the measured sea level is correlated with the sea-state. This method is somewhat indirect and includes a host of problems arising from the inclusion of instrumental errors, errors in other corrections applied to the altimeter height measurements, and other sources of sea level variability. Using Seasat data Born et al. [1982] found the EM bias, plus other included errors, to be approximately 7% of SWH. Douglas and Agreen [1983] found the EM bias, plus other errors, to be 6.4 \pm 0.6\% of SWH with Seasat data, and 1.9 \pm 1.1\% of SWH using Geos-3 data. Seasat instrument errors were independently estimated to be from 5 to 5.5\% of SWH [Hayne and Hancock, 1982; Lipa and Barrick, 1981], resulting in EM bias measurements of approximately 2\% of SWH for Seasat. Using Geosat data Cheney et al. [1989] predicted the EM bias to be 1\% of SWH, but cited a number of problems associated with this indirect method for obtaining a bias measurement. This result was refined by Nerem et al. [1990] who obtained a bias estimate of 3.6 \pm 1.5\% of SWH. And finally, Ray and Koblinsky [1990], also using Geosat data, reported a bias of 2.6 \pm 0.2\% of SWH, with an additional correlation of the bias with wind speed.

Finally, a mention of theoretical studies of the EM bias should be made. The approximations of physical optics are used to model the backscatter of a radar pulse from the sea surface as the reflection from a collection of small mirror-like facets randomly distributed over the sea surface [Barrick, 1968]. Estimations of the distribution of the facets lead to predictions of the EM bias, such as the work by Barrick and Lipa [1985], who determined the bias should be 2 to 3\% of the SWH, but reported a large uncertainty. Although this result appears to agree with other investigations, some of the approximations used
in the theoretical approach may be too over-simplified to be render the result relevant [Rodriguez and Li, 1987]. A recent study by Arnold et al., [1990], using wave gauge measurements of the ocean surface, along with the methods of physical optics, found the bias to range from 2% to 5% of SWH, with a close agreement between the predictions and direct measurements of EM bias at two different frequencies. Table 1.1 gives a summary of bias predictions.

Table 1.1. Predictions of EM bias.

<table>
<thead>
<tr>
<th>Investigator(s)</th>
<th>Method</th>
<th>Bias (% of SWH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yaplee et al. (1971), Jackson (1979)</td>
<td>Platform meas.</td>
<td>5%</td>
</tr>
<tr>
<td>Walsh et al. (1984) [36GH]</td>
<td>Air-borne meas.</td>
<td>1.1±0.4%</td>
</tr>
<tr>
<td>Choy et al. (1984) [10GHz]</td>
<td>Air-borne meas.</td>
<td>3.3±1.0%</td>
</tr>
<tr>
<td>Hoge et al. (1984) [UV]</td>
<td>Air-borne meas.</td>
<td>(-)1.4±0.8%1</td>
</tr>
<tr>
<td>Melville et al. (1990)</td>
<td>Platform meas.</td>
<td>1.3% to 5.8%2</td>
</tr>
<tr>
<td>Born et al. (1982)</td>
<td>Seasat data</td>
<td>7.5±2.8%</td>
</tr>
<tr>
<td>Douglas and Agreen (1983)</td>
<td>Seasat data</td>
<td>6.4±0.6%</td>
</tr>
<tr>
<td>Geosat data</td>
<td>Geosat data</td>
<td>1.9±1.1%</td>
</tr>
<tr>
<td>Cheney et al. (1989)</td>
<td>Geosat data</td>
<td>1%</td>
</tr>
<tr>
<td>Nerem et al. (1990)</td>
<td>Geosat data</td>
<td>3.6±1.5%</td>
</tr>
<tr>
<td>Ray and Koblinsky (1990)</td>
<td>Geosat data</td>
<td>2.6±0.2%2</td>
</tr>
<tr>
<td>Barrick and Lipa (1985)</td>
<td>Theoretical</td>
<td>2% to 3%</td>
</tr>
<tr>
<td>Arnold et al. (1990)</td>
<td>Theoretical+Platf.</td>
<td>2% to 5%</td>
</tr>
</tbody>
</table>

1 Using ultraviolet frequencies, Hoge et al. found the measured mean sea level was above the true value, i.e. the EM bias effect was in the opposite direction as the effect observed at other frequencies.
2 The EM bias was found to have an additional dependence on surface wind speed.

Although the predictions for EM bias are varied, it is advantageous to apply a bias correction to satellite altimeter measurements, and consider the resulting effect and implications of such a correction. The algorithm developed by Melville et al. [1990] during the SAXON experiment is based on the most extensive set of direct observations of bias presently available. Despite a few shortcomings, data collected by the Geosat satellite represents the most complete set of satellite altimeter measurements presently available.
Therefore, applying the bias algorithm from Saxon to Geosat data will give the best available prediction of the spatial scales and gradients of bias, as well as seasonal and latitudinal dependence of bias variations.

1.3 Geosat

The U.S. Navy satellite Geosat (Geodetic Satellite) was the third satellite altimeter flown which made direct measurements of the sea level and its variation. In 1975, Geos-3 generated a number of important geophysical and oceanographic advances, but suffered because of poor ground coverage. Seasat, flown in 1978, carried significantly improved instrumentation along an orbit which provided improved coverage, but it suffered a short circuit after only three months, which rendered it inoperable. After 18 months of classified operation, in late September 1986 Geosat completed a series of maneuvers which altered its orbit to match the orbit flown by Seasat. Geosat collected oceanographic data along repeating ground tracks for the next 3 years, until its failure in January 1990. Plans for future altimetric satellites, such as the joint French-American Topex/Poseidon mission scheduled for mid-1991, are already underway.

The Geosat Exact Repeat Mission (ERM) travelled along an exact repeat orbit, meaning each ground track is repeated (within 1 km) every 17.05 days, or after 244 revolutions (Figure 1.4). Geosat’s altitude was approximately 800 km, and the spacing between ground tracks is approximately 150 km at the equator. In collaboration with the Johns Hopkins University Applied Physics Laboratory, NOAA generated and distributes the Geosat ERM Geophysical Data Records (GDR’s) [Cheney et al., 1987]. The GDR’s include 34 channels of data sampled every second, or approximately every 7 km. The 1-per-second values include time, latitude, longitude, orbit height, sea surface height, geoid
height, SWH, $\sigma_0$, several diagnostics, and the corrections applied for tides and path delays.

Although Geosat data has been overwhelming successful in furthering our understanding of the ocean, it also has three major shortcomings [Douglas and Cheney, 1990]. First, unlike Seasat, Geosat had no on-board instrumentation to measure tropospheric water vapor. The altimetric path delay corrections were derived from climatological values based on three years of satellite radiometer data, in conjunction with a global model of atmospheric water vapor, updated every 12 hours by the Fleet Numerical Oceanographic Center (FNOC). Secondly, the ephemeris calculations are based on tracking data collected from only the continental U.S. and Hawaiian stations, resulting in large (3 to 4 m) ephemeris orbit errors. Finally, Geosat suffered from large excursion of the spacecraft attitude from nadir (the point directly below the satellite). An attitude of greater than 1° (from nadir) can cause erroneous altimeter measurements, caused by the nadir “footprint” at the ocean surface not being fully illuminated by the radar pulse. Furthermore, when the attitude was greater than about 1.1°, a signal distortion occurred whenever the satellite passed over a transition from land to water, which resulted in large data gaps [Cheney et al, 1988].
Figure 1.4. Geosat ground tracks for a complete repeat cycle. Ascending tracks angle upward to the left, and descending tracks are downward to the left. Note data gaps often occur as the satellite passes from over land to over water, if the attitude is greater than 1.1°
CHAPTER II. GEOSAT DATA PROCESSING

2.1 JPL Data Tapes

A version of NOAA's Geosat Oceans Geophysical Data Records (GDR's), which had been edited and condensed by the Jet Propulsion Laboratory (JPL) [Zlotnicki et al., 1989a] was obtained from Dr. C. Wunsch at MIT. The orbits, atmospheric fields, tides, inverse barometer effect, and tropospheric path delays included with the GDR's [Cheney et al., 1987] had already been applied. The included variables, altimetric sea surface height (H), significant wave height (SWH), and radar cross section ($\sigma_0$), had been edited by JPL by deleting data over land or ice, deleting out-of-range data ($\sigma_0 > 35$ dB, $H < -140$ m, $H > 10$ m), and detecting and removing spikes (if the 10-per-second GDR values of H, and SWH, averaged over 1 second intervals had a standard deviation of greater than 15 cm). Other possibly erroneous data, including locations where the satellite attitude exceeded 1°, were flagged but not deleted.

The remaining data which did not fail the above editing criteria were separated into 244 ascending and 244 descending files corresponding to the ground track extending between 70°N and 70°S. Files were named by the equatorial crossing longitude of the ground track (e.g., 306.4A refers to the ascending track which crosses at 306.4°). Each file contains the data records from each repeat pass of the satellite along that track, with an identifying header separating repeats. Each data point record includes the time, latitude, longitude, the three variables H, SWH and $\sigma_0$, and a one byte flag variable which marks the occurrence of certain conditions. The flag conditions include: satellite attitude > 1°; 1-second standard deviation of H > 10 cm; automatic gain control < 16 dB, or > 40 dB; dh(SWH/attitude) [which is the height bias resulting from a combination of SWH and attitude] or dh(FM)
[which is the height bias resulting from linear compression of the altimeter pulse] out of range (GDR flag 2); FNOC interpolation >12 hrs (GDR flag 12); ionosphere model problems (GDR flag 13).

JPL interpolated all data to a set of reference latitudes, which was chosen to include the equator, and give an along-track time step of 1 s (~ 7 km). Thus, all tracks had data points at the same latitudinal locations, while the longitudes were dependent on the track. The slight departures (± 1 km) of the satellite from the exact repeat orbit caused the longitudes of any given track to vary slightly between repeats cycles. Included with the JPL data tapes was a program [called GETDATA], which could be used to extract the data of interest (which variables), from a given location (between latitudinal and longitudinal bounds), between given dates. [Note: All programs referred to are included in Appendix A.]

2.2 Editing and Averaging

The area chosen to study is the North Atlantic, from 0° to 100°W and 0° to 65°N, during the first 24 repeat cycles, from November 8, 1986 to December 31, 1987. The data were further edited by deleting all records flagged for an attitude greater than 1.0° from nadir, as recommended by Cheney et al., [1988], and discarding any record missing a value of either σ₀ or SWH, caused by failing any previous editing scheme. The location of other flagged data, along with the flag type, were stored for later reference, but the data were not discarded. The SAXON bias algorithm was used to compute a value of the EM bias, B, for each data record. [The editing and bias calculations were performed using the program BIAS].

Although some Geosat data lies outside the range of sea-states encountered during the SAXON experiment (0 < SWH < 3 m, 9 < σ₀ < 17 dB),
especially at the upper latitudes, and during the winter months, the algorithm was applied to all the data, without limitation. Therefore, the bias values obtained in the extrapolated regions are viewed as predictions which are dependent on the validity of extrapolation. Furthermore, the altimeter aboard Geosat operated at 13.5 GHz, while the SAXON algorithm was based on measurements at 14 GHz, but this small frequency difference is expected to have a negligible influence on the bias. After computing the bias, the complete data record: time, latitude, longitude, H, SWH, σ₀, and B, was put into a file identified by the track name, and the repeat cycle number.

Two different averaging techniques were utilized during the processing steps described in the following sections. The dynamic height profiles of the Gulf Stream tracks, and the tracks used in the analysis of the change in bias near storms, were smoothed using a 35 km (or 5 point) running boxcar-mean [the program BOXAVE]. Thus, at each data point location, the value of the variable is averaged together with its nearest 4 neighbors (two on each side) and the resulting average at each location is recorded. The running boxcar method is primarily used for smoothing data along a continuous segment. The second method, termed block-averaging, is used mainly to reduce the quantity of data. The data reduction necessary to produce the color contour maps utilized a 50 km block-average [the program BLKAVE]. Because there were often gaps in the data, the block was defined using the satellite sample time information, and averages were actually computed over 7-second intervals, which is approximately 50 km (or 7 points) if the sampling is continuous. The average value of the variable over the 50 km block was determined, and then recorded at the center location of the 50 km segment.
2.3 Orbit Error Removal

Geostrophic ocean current velocity is directly related to the slope of the sea surface, relative to the geoid [Pedlosky, 1987]. The difficulties involved in accurately determining a detailed geoid [Nerem et al., 1990], and hence the inability to define an exact surface to which the sea surface topography can be referenced, leads to the method of collinear differencing. The exact repeat nature of Geosat causes each ground track to be repeated within 1 km every 17 days. By subtracting the altimetric height (H) of one repeat from another, all time-invariant components of H, including the geoid and the permanent sea surface topography are removed. The resulting residual, termed the dynamic sea surface height, is on the order of a meter in the region where the Gulf Stream meanders from its mean position. The longitudinal variations of ±1 km result in a sea level error of less than 2 cm in this region [Brenner et al., 1990].

Inaccuracies in the orbit prediction models introduce a large radial orbit error (~ 3 m) in the altimetric height measurements [Sandwell and Zhang, 1989]. Fortunately these errors are of a wavelength equal to the orbital circumference, and are easily removed by modeling the error as a low order polynomial on a regional basis. First, for each track, a long-term average of the mean sea level was determined by averaging the first 24 repeat cycle values of altimetric height measurements (H) at each data point location along the track. Next, the long-term average was subtracted from the track data of the repeat cycle of interest [using the program RMEAN] giving the dynamic sea surface height profile. Finally, to remove the orbit error (often referred to as bias and tilt) a second degree polynomial was fitted and removed from 2500 km along-track segments of this residual [using the program RPOLY]. The segment length of 2500 km was chosen so that the long
wavelength errors would be removed, but the mesoscale (50–1000 km) variations of the dynamic topography would be preserved [Zlotnicki et al., 1989b].

2.4 Repeat Cycle Map Processing

Color contour mapping is a very useful (and standard) method for presenting the abundance of data collected from satellite observations. Data from the first 24 repeat cycles of Geosat were processed to be plotted on color contour maps, with each map representing an average over the 17 day repeat cycle, with a spatial resolution of 1° x 1° (latitude, longitude). The maps were produced using plotting software developed at Woods Hole Oceanographic Institution (WHOI) [Caruso and Dunn, 1989].

All track data were first block-averaged in the along-track direction in 50 km segments and the average value of the variable was placed at the center location of the 50 km segment [using the program BLKAVE]. This was done to make the along-track data density comparable to the between-track spacing before attempting to interpolate to a uniform space grid. Next, to reduce the evidence of track patterns in the final contours, a two-step gridding process using Laplacian interpolation was used. To fill in the diamond shaped patterns formed by crossing tracks, the averaged track data were first gridded to four 2° x 2° grids, each offset by ±0.5° x ±0.5° [using the program ZGRID]. These four grids were then used as input to the second interpolation [again using ZGRID], which resulted in a uniform 1° x 1° grid. The final contour maps were smoothed using a 5 x 5 pixel median filter included in the WHOI plotting software.
CHAPTER III. EVALUATION OF EM BIAS

3.1 In The Gulf Stream

The ascending Geosat tracks cross the Gulf Stream nearly orthogonal to its mean path, and parallel tracks are 115 km apart and separated by 3 days. As a test of the processing, a comparison of the dynamic sea surface height profiles of three such tracks (306.4A, 307.9A, 309.4A), during repeat cycle 13 (May 31, June 3, and June 6, 1987) was made with published results by Kelly and Gille [1990]. For each track, a mean sea surface height profile was computed from the first 24 repeat cycles of the track. To remove the geoid, and other time-invariant components, the long-term average profile of the track was subtracted from the individual height profile of repeat 13, creating a difference profile. A second degree polynomial was fitted and removed, and the resulting dynamic height residuals were then smoothed using a 35 km running boxcar mean, and are shown in Figure 3.1. The values agree with Kelly and Gille's within 5%, and the differences can be attributed to slightly different techniques used to remove wavelength error, and using a one year long-term average, instead of three years (personal communication, Kelly, July 1990).

As shown by Kelly and Gille [1990], the location of the maximum surface current velocity and transport correspond with the position of the relative maximum gradient in the dynamic height profiles. The location of the maximum slope which occurs within the Gulf Stream (determined by the location of warm water observed on an IR image, see Kelly and Gille [1990]) is marked with a X for each track in Figure 3.1. The EM bias for these three tracks was computed (as previously described) and smoothed, again using a 35 km running boxcar. The dynamic height profile and bias for each track are
shown in Figure 3.2. All three tracks show significant gradients in the bias across the Gulf Stream.

The maximum geostrophic velocity of each of the three tracks was determined using the geostrophic approximation [equation 1.1] with the maximum slope of the dynamic height profile. The maximum velocities for Tracks 306.4A, 307.9A, and 309.4A were found to be 0.9 m/s, 1.2 m/s, and 1.6 m/s respectively. To estimate a bias error contribution, the maximum gradients in the bias were used with the geostrophic approximation, which gives velocities arising from the bias of 0.06 m/s, 0.24 m/s, and 0.13 m/s for the three tracks. Therefore, the EM bias gradients introduce an error in the geostrophic velocities ranging from 7% to 20%. Note also, as shown in Figures 3.2b and 3.2c, the bias and dynamic height gradients may be in either the same or opposite directions. Figure 3.3 shows the corresponding SWH and \( \sigma_0 \) for each track.
Figure 3.1. Dynamic height residuals for three tracks crossing the Gulf Stream. The height residuals are projected perpendicular to the ground tracks. Track identifications in parentheses refer to Kelly and Gille, [1990], Plate 1. Locations of the relative maximum gradients correspond to maximum within the Gulf Stream, as delineated by warm water shown in the IR image (Plate 1) of Kelly and Gille. The dynamic height residual profiles shown here agree with those of Kelly and Gille within 5%.
Figure 3.2. EM bias shown with the dynamic height residuals of the three Gulf Stream tracks from Figure 3.1. Note the large bias gradient in (b), and the differing gradient directionality of the bias and dynamic height in (c).
Figure 3.3. Significant wave height, radar cross section ($\sigma_0$), and resulting EM bias for the three Gulf Stream tracks shown in Figures 3.1 and 3.2.
3.2 Near Storms

A Tropical Storm

The complex space-time sampling characteristics of Geosat, in conjunction with the relatively short life span of most tropical storms, often precluded coverage of regions of interest during the 1986 and 1987 hurricane seasons. Each ground track is repeated every 17 days, and for any given track the nearest track in the same (ascending/descending) direction to the west is 3 days before the given track, while the nearest track to the east is 3 days later. The possibility of Geosat altogether missing a large storm can be evidenced by the fact that near the equator the crossover points of the ascending and descending tracks are spaced 1.5 days apart resulting in dense temporal sampling for 8.5 days, with no data for the remaining 8.5 days of the repeat cycle.

In mid-August 1987, Geosat passed over Hurricane Arlene relatively closely in both space and time. Arlene was the second strongest storm of the season with 65 knots maximum winds sustained, and a minimum pressure of 987 mb [NOAA Storm Data Summaries, 1987]. On August 13, at 23:30Z, Track 309.4A crossed the path of Arlene approximately 200 km or 24 hours behind the center of the storm, which had 55 knot winds and a pressure of 1000 mb. At that time, a nearby ship, roughly 60 km from the storm's center, and 75 km from Track 309.4A, reported 65 knot winds in heavy weather [NOAA Storm Data Summaries, 1987]. At 13:00Z on August 15 Track 285.0D crossed Arlene's path 150 km or 18 hours behind the storm's center, which had 50 knot winds and a pressure of 995 mb. The path of Arlene, along with the two satellite ground tracks and the position of the ship are shown in Figure 3.4.
The Geosat measured SWH, $\sigma_0$ and resulting EM bias of the two tracks were smoothed using a 35 km running boxcar-mean, and are shown in Figure 3.5. Note the peak values of bias, which occur when the ground track passes closest to the storm, are more than double the ambient bias values in the region. Recalling that these satellite tracks missed the storm’s center by some distance, it is clear that the bias contributions of hurricanes could be considerable, especially since most hurricanes occur in otherwise quiet ocean basins.
Figure 3.4. The path of Hurricane Arlene, along with the location of the two satellite ground tracks which pass nearest the storm. Track 309.4A passed through the area at 23:30 Z on August 13. Track 285.0D passed through at 13:00Z on August 15. Also shown is the location of the ship Nivosa, which reported 65 knot winds at 18:00Z on August 13. [Adapted from NOAA Storm Data Summaries, Aug. 1987.]
Figure 3.5. Significant wave height, radar cross section ($\sigma_0$), and resulting EM bias for the two track shown in Figure 3.4, passing nearest to Hurricane Arlene.
An Extratropical Storm

The North Atlantic has a very active winter storm season at latitudes above 35°, during which the extratropical storms can cause severe weather and sea-state conditions. Synoptic maps of wind speed and significant wave height predictions were obtained from the U.S. Navy Fleet Numerical Oceanographic Center (FNOC) for one such storm during repeat cycle 5. The data from tracks 285.0D (during hour 21 of January 22, 1987), and 301.3D (during hour 19 of January 21, 1987) were smoothed using a 35 km running boxcar-mean, and compared to values obtained from the nearest two 12 hour FNOC synoptic maps. The two satellite tracks are shown overlaid on a FNOC map in Figure 3.6. Figure 3.7 shows the close agreement of the SWH measurements and predictions for both tracks. Note that the data for track 301.3D ends abruptly at 40° latitude where the track encounters the center of the storm, and the remaining GDR data are missing, while track 285.0D misses the center of the storm by just over 100 km.

A comparison of wind speed is more difficult. Many algorithms have been developed to convert the radar cross section (σ₀) to wind speed, but there are large discrepancies between the conversions when σ₀ is less than 10 dB, or alternately, when the wind speed exceeds approximately 20 knots, as shown in Figure 3.8 [from Dobson et al. 1987]. In addition, the range of applicability of the algorithms exclude the high wind speeds (greater than 30 knots) found in large storms. Figure 3.9a shows the σ₀ values obtained from the track data. Figure 3.9b shows the nearest two 12 hour FNOC synoptic map predictions of wind speed for the two tracks. A few σ₀ values were converted to wind speeds using the algorithms shown plotted in Figure 3.8, and the range of resulting wind speed values are shown as bars in Figure 3.9b.
For each of the tracks the EM bias is shown in Figure 3.10. Although the range of direct measurements, on which the bias algorithm is based, did not include these extreme conditions (SWH > 3 m, or $\sigma_0 < 9$ dB), it is important to note both the large bias values (nearly a meter), and gradients in the bias. This extrapolation of the algorithm shows the EM bias may be a significant fraction of the dynamic sea surface height in areas of extreme sea-state conditions.
Figure 3.6. Satellite ground tracks 301.3D and 285.0D shown overlaid on a sample FNOC 12 hour synoptic map. Significant wave height comparisons are shown in Figure 3.7, and wind speed comparisons are shown in Figure 3.9.
Figure 3.7. Comparison of SWH data from two Geosat tracks (shown in Figure 3.6): 301.3D (at 19:00Z on Jan. 21, 1987) and 285.0D (at 22:00Z on Jan. 22, 1987) with the nearest FNOC 12 hour synoptic map predictions during a strong winter storm. Note the abrupt ending of Track 301.3D as it encounters the center of the storm, and the remaining data was either edited or flagged.
Figure 3.8. Comparison of algorithms from various investigators, comparing altimeter-derived radar cross section ($\sigma_0$) with surface wind speed. From Dobson et al., [1987].
Figure 3.9. Comparison between radar cross section ($\sigma_0$), and wind speed for the two tracks shown in Figure 3.6. (a) shows the satellite track data $\sigma_0$, and (b) shows the two nearest 12 hour FNOC synoptic map predictions for winds speed along the two tracks, (with the average of the two 12 hour values drawn for reference). A few points from (a) were converted to a range of wind speed using the various algorithms given in Dobson et al., [1987], shown in Fig. 3.8. The resulting range of wind speeds are shown in (b) as bars.
Figure 3.10. EM bias, during a strong winter storm, along the two tracks shown in Figure 3.6.
3.3 Repeat Cycle Averages

Repeat cycle averages from the first 24 repeat cycles (14 months) of Geosat were plotted on color contour maps to investigate the spatial scale and seasonal variations of the bias predictions (Figures 3.11 through 3.31). Three of the repeat cycles (3, 4 and 6) were not used because all the data had either failed the previously described JPL editing criteria, or the data had been flagged for having an attitude greater than 1°. Each repeat cycle map represents an average over the 17 day repeat cycle, averaged over 1° x 1° bins, and smoothed using a 5 x 5 pixel median filter.

Figures 3.11 through 3.31 show that during winter months extratropical storms can give rise to large regions having very large sea-states, with values of SWH > 7 m, and/or σ₀ < 8 dB, as delineated by white on the contour maps. As previously stated, the Saxon bias algorithm is based on measurements extending only to SWH ≤ 3 m, and σ₀ > 9 dB, and therefore is being extrapolated over large areas of these three figures. Nevertheless, it is significant to note that the storms show an apparent basin-scale influence on the resulting EM bias. The absence of such storms during the summer months, in addition to generally quieter seas, gives significant evidence of a seasonal variation in the bias. As can be seen in Figure 3.32 (from Zlotnicki et al., 1989b), the rms sea level variability, relative to a 1-year mean, is of roughly the same order as the seasonal variability of the EM bias in many regions of the North Atlantic. The significance of this observation is that seasonal variations of EM bias, if not accounted for, could be incorrectly interpreted as seasonal variations in the sea level.

To quantify the seasonal variations observed in the color contour maps, an average EM bias value was computed for three latitude-bands for each repeat cycle. The North Atlantic was divided into three regions consisting of
the Tropical latitudes, from 0° to 20°N, the Sub-Tropics, from 20°N to 40°N, and the Extratropical region, from 40°N to 60°N. For each repeat cycle, all the track data was separated into these regions, and the resulting average values of SWH, σ₀, and bias for each latitudinal band were computed, and are shown in Figure 3.33. Note that the average EM bias values were computed using all data records, and as stated before, the bias algorithm is being extrapolated beyond the range of direct measurements on which it is based. As previously stated, averages for repeat cycles 3, 4 and 6 were not obtained because all the track data from those repeats were either flagged for having an attitude excursion of greater than 1°, or failed some other editing criteria.

As can be seen in Figure 3.33c, there is minimal variation in the bias in the Tropics. The relatively infrequent tropical storms and hurricanes, occurring mainly from August to October, do not contribute noticeably to the overall signal. The Sub-Tropical band has evidence of some seasonal signal, while the Extratropical latitude band shows a significant seasonal variation. Slight differences from year to year would be expected as the number and intensity of storms during a particular time period varies, and indeed the differences between the Sub-Tropical average value of repeat cycle 2, and that of repeat cycle 23, as well as the differences between the Extratropical average of repeat cycle 1 and 22, can be attributed to the differences in the number of storms during those two time periods in 1986 and 1987 [NOAA Storm Data Summaries, 1986-1987].
Figures 3.11 through 3.31. Repeat cycle averages of significant wave height, radar cross section ($\sigma_0$), and resulting EM bias, averaged over $1^\circ$ bins, and 17 days.
Figure 3.32. The root-mean-square of sea level variability, relative to a 1-year mean sea level, averaged over 1° bins and 1 year. From Zlotnicki et al., [1989], Plate 2.
Figure 3.33. Latitude-band averages of significant wave height, radar cross section ($\sigma_0$), and EM bias for each of the first 24 Geosat repeat cycles (except 3, 4, and 6), covering the time period from Nov. 8, 1986 to Dec. 31, 1987.
CHAPTER IV. CONCLUSION

The importance of the predictions for EM bias become evident when considered in relation to the signals of interest. Across the Gulf Stream it was found that the EM bias gradients can contribute an erroneous geostrophic current velocity of approximately 10% of the maximum velocity obtained from the dynamic height gradient. In addition, the bias gradient was found to be opposite to that of the residual dynamic height gradient in some instances. Therefore, the overall errors included in the geostrophic velocity, if the bias is not properly accounted for, could be more than 10%, since in some cases the bias is, in effect, adding to the inferred velocity, and at other times decreasing it.

In a typically quiet tropical basin, with generally small EM bias gradients, the influence of a tropical storm was found to more than double the values of bias normally found in the region, as far as 200 km from the center of the hurricane. Strong winter storms in North Atlantic induced EM bias values of up to a meter, along with large bias gradient values. Using the geostrophic approximation, the EM bias gradients could be erroneously interpreted as inducing a geostrophic velocity of roughly 0.2 m/s, which is significant in regions which have no major current systems. In regions affected by strong winter storms it was necessary to extrapolate the algorithm used to predict the bias beyond the range of direct measurements on which it was based. Although the resulting large bias have not been tested by direct measurements, the assumption that the EM bias continues to increase with increasing sea-state is not inconsistent with available data.

Many investigators have developed predictions for EM bias by solving for values of bias as a function of SWH, that when applied to repeated profiles,
improved the agreement of sea surface height. *Cheney et al.* [1989] used this method with Geosat data to obtain a bias prediction of 1% of SWH, although he warned that the procedure he used to remove the radial orbit errors also removed a significant variation of the SWH. *Nerem et al.* [1990] used the same method, but applied improved geoid and orbit models and obtained a bias estimate of $3.6 \pm 1.5\%$ of SWH. Finally, *Ray and Koblinsky* [1990], also using the same method, incorporated a dependence on wind speed into their prediction of bias, although they rejected all data from areas of high variability, and obtained an expression for bias as $B = -0.66 \ H_{1/3} - 0.15 \ H_{1/3} \ U_{10}$, with $H_{1/3}$ (SWH) in m, and $U_{10}$ in m/s. At the location of the 1 m maximum bias in Figure 3.10, *Ray and Koblinsky* [1990] would predict a bias of only 29 cm, and *Nerem et al.* [1990] would predict $25 \pm 10$ cm. The fundamental difficulties in their approach, arising from large radial orbit errors, the presence of other sources of sea level variability, and the need to determine exact sea surface height profiles, may account for the large discrepancy. In addition, by restricting their study to regions with rms sea level variability less than 8 cm, *Ray and Koblinsky* [1990] rejected the areas where these large storms occur (see Figure 3.32).

The EM bias was also found to vary both as a function of latitude and of season. Although the variation in bias was expected, since both SWH and $\sigma_0$ are seasonally dependent, the resulting bias variation is very large at the upper latitudes: nearly a ten-fold increase in the average bias during the winter as compared to the summer. This seasonal dependence of the EM bias is especially important when considering seasonal variations of the mean sea level. Until the bias error is appropriately accounted for, general ocean circulation studies [*e.g.* *Wunsch and Gaposchkin*, 1980] may be erroneously
including the seasonal variation of the EM bias in predictions of seasonal circulation variations.

With these considerations, it is evident that the failure to accurately account for the EM bias error may introduce a significant error in the altimetric height measurement of the sea surface. If indeed the SAXON bias algorithm is valid in the extrapolated regions, the EM bias error can be significantly larger than the values currently used to interpret satellite data. The uncertainty of the extrapolated algorithm, and the possibility of significant EM bias values in the extrapolated regions, emphasizes the need for further studies involving direct EM bias measurements at higher latitudes, in regions with high winds and sea-states.

The study of ocean dynamics using satellite altimetry is dependent on understanding and correcting for the various errors incurred in the measurement system. Predicting the EM bias error is complicated by the fact that waves and currents interact, which leads to modulation of the wave field by currents, which also implies a modulation of the bias. Thus, the bias may correlate with the signal of interest. This dynamic coupling of the smallest scales of the surface roughness with mesoscale features result in wave interactions that can have basin-scale influences in the bias. Understanding these mechanisms is necessary to fully account for the EM bias errors, but, until such knowledge is achieved, the algorithms based on direct measurements serve as a basis for prediction and correction of such errors.
REFERENCES


NOAA Storm data with annual summaries, 28, 12 and 29, 12, National Climatic Data Center, Asheville, NC., 1986 and 1987.
ACKNOWLEDGMENTS

I’d like to thank the following:

My advisor, Ken Melville for all his support and criticisms.
Hans Graber for lending me his Sun Sparkstation, his ideas, and a ride when my car was towed.
Theodric Young for always being there (with his Macintosh expertise).
Carl Wunsch and Charmaine King for lending me the JPL data tapes, and programs to decipher them.
Mike Caruso for his help with the SDPS contouring software.

This work was supported in part by NASA grants NAGW-1272 to the Massachusetts Institute of Technology.
APPENDIX A. DATA PROCESSING PROGRAMS

Programs include:

1. GETDATA: Extracts selected variables from JPL data tapes, given input of desired location (range of latitudes, and longitudes), and beginning and end date. Included with JPL data tapes.

2. BIAS: Edits data for attitude > 1°, and missing values of SWH or σ₀, stores location of flagged data, along with flag type, and computes bias value for each data point.

3. BOXAVE: Applies running box-car mean to segment of data (usually 35 km).

4. BLKAVE: Applies block averaging to data segment (usually 50 km).

5. RMEAN: Computes long-term mean sea level, and removes mean from data of specified repeat cycle to create difference profile (dynamic height residual).

6. RPOLY: Fits and removes second-order polynomial from dynamic height residual profile to remove tilt and bias of orbit error.

7. ZGRID: Performs Laplacian interpolation to grid a random array of points onto a uniform space grid.
The following set of programs and subroutines is a package received from JPL with the data tapes. The package as a whole is referred to as *GETDATA*.

The package includes: maingf.f, extgfc2.c, getbits4.c, output.c, subgf5.f, xdays.c, and secs.c

The following set of programs and subroutines is a package received from JPL with the data tapes. The package as a whole is referred to as *GETDATA*.

The package includes: maingf.f, extgfc2.c, getbits4.c, output.c, subgf5.f, xdays.c, and secs.c

Sample calling routine for program which extracts data from gridded data set.

input files:

gdatdir.inp (a character constant!)
must have the path names to (3) directories, one per record, holding the data files for 86/87, 88, 89, in that order. Example:
/u/geosat/data87
/u/geosat/data88
/u/geosat/data89

files (a variable whose value is input to the pgm):
name of a file, each record of which is the name of a track file. Only those data files listed here will be read. Example:
2385A.int
0010A.int
0025A.int
0040A.int
will cause the files /u/geosat/data87/2385A.int,
/u/geosat/data88/2385A.int, /u/geosat/data89/2385A.int
to be read, and the same for the remaining files.
(the file name is the nominal equator crossing longitude in tens of degrees. A (D) for ascending (descending).

input parameters:

param - character variable holding names of selected parameters
(h1 - height with out tides, h2 - height with tides,
swh - significant wave height, snau - sigma naught,
flag - data flag, 0 - no parameter, just time
latitude and longitude)
rmnl - minimum latitude range
rmxl - maximum latitude range
rmnln - 1st longitude
rmxln - 2nd longitude (East to West selection)
datel - beginning repeat track (if itrkl is greater than itrk2)
date2 - ending repeat track (all tracks will be output)
files - file containing track files to read (unit=11)
edit - edit=0 when no editing is done for out-of-range swh and snau
edit=1 when all records with out-of-range swh and snau are deleted

output parameters:

xlat() - latitude of data points
xlon() - longitude of data points
time() - time of data points
spar() - array which stores selected parameters
C cflag() - array which stores flag data
C eqxl() - longitude of equator crossing (output)
C eqxt() - time of equator crossing (output)
C npts() - number of standardized points selected
C inpts() - number of interpolated points selected (i.e. points not set to
C 32767)
C nrtrk - number of repeat tracks output
C itrkl - repeat track number of 1st repeat track selected from file
C oflag - indicates where flag is output in spar, and if not set to <0
C ier - exit indicator (0 - normal, neg. no. - error in read, (-9) - last
C file in files read)
C
C parameter statement constants:
C
C NSIZE - maximum number of data points in each repeat
C MAXTRK - maximum number of repeats in each repeat track file
C NPARAM - maximum number of parameters to output
C Note: parameters will be output in array in order specified
C parameter (NSIZE=2915, MAXTRK=65, NPARAM=5)
C declare local parameters
C character afile*40
C**************** real rlon(NSIZE)
C declare input parameters
C character param*80, files*40, datel*7, date2*7
C integer edit
C declare output parameters
C****************** character cflag(NSIZE, MAXTRK)*8
C dimension xlat(NSIZE), xlon(NSIZE), spar(NSIZE, MAXTRK, NPARAM)
C dimension eqxl(MAXTRK)
C real*8 time(NSIZE, MAXTRK), eqxt(MAXTRK)
C integer inpts(MAXTRK), npar, nrtrk, np2, unit, oflag
C input variables in common block inpvar
C data ier, unit, edit/0, 21, 1/
C declare other necessary variables
C create input file containing the following information
C line 1: h1, h2, swh, snau, flag (select one to five parameters)
C line 2: rmnlt, rmxlt, rmnln, rmxln
C line 3: datel, date2 (if itrkl.gt. itrk2 all repeat will be read)
C line 4: files
C
C afile = 'file'
C nfile = 0
C read input
C read(5, '(a)') param
C read(5, *) rmnlt, rmxlt, rmnln, rmxln
C read(5, '(a6, 1x, a6)') datel, date2
C read(5, '(a40)') files
C call extraction routine
10 call extgfc2(param, datel, date2, files, rmnlt, rmxlt, rmnln, rmxln,
> npar, nrtrk, np2, inpts, eqxl, xlat, xlon, eqxt, time, spar, itrkl,
> oflag, MAXTRK, NSIZE, NPARAM, edit, ier)
C print*, '1st track select is', itrkl
C if (ier.eq.-9) then
C stop 'All files read.'
C else
C if (ier.gt.0) stop 'ier gt 0'
C ... name output file
C nfile = nfile + 1
write(afile(5:7),'(13)') nfile
   do k=1,7
      if (afile(k:k).eq.' ') afile(k:k)='0'
   end do
   c ... print out if option selected
   call output(afile,xlat,xlon,time,spar,eqx1,eqxt,
      > np2,inpts,nrtrk,itrk1,npar,oflag,NSIZE,MAXTRK,NPARAM)
   c ... go to read next track file
      goto 10
   endif
   end
** Subroutine extgfc2 (A. Hayashi 8/16/88)  

** Reads gridded data files and outputs them based on time, latitude 
** and longitude selection. Set up to be call by a FORTRAN program. 
** version 4 - indexes corrected (12/15/88) 
** Subroutine call has been changed. Edit option has been added. 
** Opens file to read subsequent years of data in different directories. 
** Create a file called gdatdir.inp in the default directory and place in it 
** the listing of all pathnames of gridded data files. 
** Example: 
** /geosat/gdata87/ 
** /geosat/gdata88/ 
** This will tell the program to read gridded data files in those two 
** directories. Specify track file name without pathnames (i.e. 0003D.int). 
** Correction (5/14/89) lines of code to put string terminators on string 
** arrays passed as arguments moved into first setup if clause.  
*/

#include <stdio.h> 
#include <sys/file.h> 
#define RECORDSIZE 15 
#define NPAR 5 
#define MAXTRK 65 
#define MAXPTS 2915 
#define DIRFIL "gdatdir.inp"

/* declare counter for out-of-range parameter */

int nidx,nti,nhl,nswh,nsnau;

extgfc2_(param,datel,date2,ifile,rmnlt,rmxlt,rmnln,rmxln, 
  npar,strk,wnpts,inpts,eqxl,olat,olon,eqxt,time,spar, 
  rpl,oflag,maxtrak,maxpts,maxp,edit,ier,11,12,13,14)

/************************ declare input variables *******************/
/* param - character variable containing names of parameter to output 
  h1 - height with tidal correction 
  h2 - height w/o tidal correction 
  swh - significant wave height 
  snau - sigma naught 
  0 - outputs only time, latitude and longitude 
  date1 - starting time of data selection 
  date2 - ending time of data selection 
  ifile - file containing list of gridded track files to read 
  rmnlt - minimum latitude range of data selection 
  rmxlt - maximum latitude range of data selection 
  rmnln - starting longitude range of data selection 
  rmxln - ending longitude range of data selection 
  edit - option to edit data 
  0 - if no editing is to be done 
  1 - if records with negative swh & snau are to be deleted 
  Note: 11,12,13,14 are string sizes and need not be specified 
  when calling from a FORTRAN program 
*/

int 11,12,13,14,*maxtrak,*maxpts,*maxp; 
char *param,*datel,*date2,*ifile; 
float *rmnlt,*rmxlt,*rmnln,*rmxln;

73
declare output variables

npar - number of parameters selected for output
strk - number of repeats which were selected for output
wnpts - number of data records in each repeat selected for output
inpts - array containing number interpolated records in each repeat
eqxl - array containing equatorial crossing longitude of each repeat
olat - array containing latitudes of each corresponding data record
olon - array containing longitudes of each corresponding data record
eqxt - array containing equatorial xing time of each repeat
time - array containing time of each data record in each repeat
spar - array containing parameters selected for output
rpl - repeat number of 1st repeat selected for output
oflag - identifies where flag is output, if not output set to <0
maxtrk - maximum dimension size of index corresponding to repeat
maxpts - maximum dimension size of index corresponding to data point
maxp - maximum dimension size of index corresponding to parameter

Note: currently variable array dimensions cannot be declared in this program due to the compiler, so dimensions passed thru argument list (maxtrk,maxpts) must match the ones defined in the subroutine (MAXTRK,MAXPTS). NPAR and maxp need not match since it is not a required dimension. Array dimensions are the inverse of FORTRAN. In C it is row major.

int *ier,*npar,*strk,*wnpts,*rpl,inpts[],*oflag,*edit;
float olat[],olon[],eqxl[];
double eqxt[],time[] [MAXPTS];
float spar[] [MAXTRK][MAXPTS];

define structures

struct record1 { /* reads latitudes and longitudes */
    int ILAT;
    int ILON;
    char JUNK[7];
};

main braces */

static int first=0,opar[NPAR],ndir;
static char sdir[5][40];
static double secs1,secs2;
static FILE *fp,*fpd,*fp3,*fp4;
char tfile[10],ofile[40],vers[16],pfile[50];
int fd,i,j,llbox=0,slen,nfiles=0;
int il,i2,tflag,ctrk,npts,epts,idir;
float lat[MAXPTS],lon[MAXPTS];
double secs();
FILE *fp2;
struct record1 buf;
int ieqxl,ieqxt,itime;
short int hpts,index,ih1,ih2,iswh,isnau;
float flag;
char pad[5];
int k,getbits(),rf;
/* check dimension - currently variable dimension sizes cannot be
** passed thru for MAXPTS and MAXTRK */
if (MAXPTS != *maxpts) {
    printf("extgfc2-e-dimension of data point index must equal %d\n", MAXPTS);
    exit(1);
}
if (MAXTRK != *maxtrk) {
    printf("extgfc2-e-dimension of track index must equal %d\n", MAXTRK);
    exit(1);
}
/* initialize ier */
*ier=0;

/* first call setup */

if (first == 0) {
    /* end all strings passed thru with null string terminator */
    for (i=0; param[i] != ''); i++)
        param[i]="\0";
    date1[6]="\0";
    date2[6]="\0";
    for (i=0; ifile[i] != '; i++)
        ifile[i]="\0";
    /* open files to store out-of-range swh and snau */
    /* if ((fp3=fopen("swh.dat","w")) == NULL){
        printf("error opening swh.dat\n");
        exit(1);
    } if ((fp4=fopen("snau.dat","w")) == NULL){
        printf("error opening snau.dat\n");
        exit(1);}
    /* initialize counters */
    nidx=0;
    nti=0;
    nhl=0;
    nswh=0;
    nsnau=0;
    /* open file containing list of directories containing gridded data files */
    if ((fpd=fopen(DIRFIL,"r")) == NULL) {
        printf("extgfc2-e-error opening file %s\n",DIRFIL);
        exit(1);
    }
    /* read all directory pathname */
    for (ndir=0; fscanf (fpd, "%s", sdir+ndir) != EOF; ndir++)
        printf("extgfc2-i-number of directories given is %d\n", ndir);
    /* open file containing list of track files to read */
    if ((fp=fopen(ifile,"r")) == NULL) {
        printf("extgfc2-e-can't open %s to read\n", ifile);
        exit(1);
    }
    first=1;
    if (*rmnln < 0.) *rmnln=360.- *rmnln;
    if (*rmxln < 0.) *rmxln=360.- *rmxln;
    if ((*rmnlt <= -72.3) && (*rmxlt >= 72.3) && (*rmnln == 0.) && (*rmxln == 360.)) llbox=1;
    printf("extgfc2-i-selected ranges.............:\n");
printf("latitude: %f %f\n",*rmnlt,*rmxlt);
printf("longitude %f %f\n",*rmnln,*rmxln);
/* sets up for parameter selection */
{slen=strlen(param);
strncpy(param+slen,"","");
/* call FORTRAN subroutine parout in file subgf5.f */
parout_(param,npar,opar,80);
if (*npar > *maxp) {
printf("extgfc2-e-number of parameters selected (%d) exceeds dimension\n");
printf("specified in argument list %d\n",*npar,*maxp);
}
/* set oflag variable */
*oflag = -1;
for (i=0; i< *npar; i++)
if (opar[i] == 5) *oflag=i;
printf("extgfc2-i-Number of parameters selected is %d\n",*npar);
/* convert datal and date2 to seconds from reference time */
/* calls secs.c and xdays.c */
secsl=secs(datel);
secs2=secs(date2);
} /* end first call setup */
/* Loop to read all track files */
/* read track file names from file till end-of-file*/
if ((fscanf(fp,"%s",tfile)) != EOF)
{
/* initialize flag for exiting next while loop */
tflag=0;
/* increment track file counter */
nfiles++;
/* set repeat track counter */
*strk=0;
ctrk=0;
/* initialize 1st index for each track file*/
il = -1;
/* add pathname to track file name */
strcpy(pfile,sdir);
strcat(pfile,tfile);
/* open track file */
if ((fd = open(pfile,0)) == -1) {
printf("extgfc2-e-error opening file %s\n",pfile);
exit(1);
}
/** read version and print message*/
vers[16]='\0';
if (!((read(fd,vers,RECORDSIZE)) == RECORDSIZE)
printf("extgfc2-e-error read version of file %s\n",tfile);
/** read lat-lons and set beginning and ending indices between
the specified lat-lon ranges */
for (i=0; i<2915; i++)
{
if ((read(fd,&buf,RECORDSIZE)) != RECORDSIZE) {
    printf("extgfc2-e-read error at read line %d\n",i);
    exit(l);
}    
lat[i]=buf.ILAT*1.e-6;
lon[i]=buf.ILON*1.e-6;

/** set 1st index for selected points */
if (il == -1)
    if ((lat[i] > *rmnlt) && (lat[i] < *rmxlt)) {
        if (*rmnln > *rmxln) {
            if ((lon[i] > *rmnln) || (lon[i] < *rmxln)) il=i;
        } else {
            if ((lon[i] > *rmnln) && (lon[i] < *rmxln)) il=i;
        }
    }

/** set 2nd index for selected points */
if (il != -1)
    if ((lat[i] > *rmnlt) && (lat[i] < *rmxlt)) {
        if (*rmnln > *rmxln) {
            if (lon[i] > *rmnln && lon[i] < *rmxln) i2=i;
        } else {
            if ((lon[i] > *rmnln) && (lon[i] < *rmxln)) i2=i;
        }
    }

} /* end of for loop */

/** if data points in specified latitude and longitude */
if (il != -1 && i2 != 0) {

} /* for loop to read all track across directories */
for (idir=0; idir<ndir; idir++) {

/* create track file name */
strcpy(pfile,sdir+idir);
strcat(pfile,tfile);
/* close old file and open new file */
close(fd);
if ((fd=open(pfile,0)) == -1){
    printf("extgfc2-e-error opening file %s\n",pfile);
    exit(l);
}
printf("extgfc2-i-Version %s of track file %s%s
",vers,sdir+idir,tfile);
/* position pointer to first repeat in file */
seek(fd,2916*RECORDSIZE,1);

/* read track header */
while ((read(fd,&hpts,2)) != 0) {

/* increment repeat counter */
ctrk++;

/* read equatorial crossing longitude and time */
if (read(fd,&ieqxl,4) != 4) {
    printf("extgfc2-e-error reading eq crossing longitude\n");
    exit(l);
}
if (read(fd,&ieqxt,4) != 4) {
    printf("extgfc2-e-error reading eq crossing time\n");
    exit(l);
}

/* read padding */
if (read(fd,pad,5) != 5)
{
    printf("extgfc2-e-error reading header padding\n");
    exit(1);
}

/*** check values of ieqxl and ieqxt */
if (ieqxl < 0 || ieqxl*1.e-6 > 360.)
    printf("extgfc2-w-crossing longitude out-of-range %d microdegrees\n", ieqxl);
if (ieqxt < 0)
    printf("extgfc2-w-crossing time negative %f seconds\n", ieqxt*.1);

/*** check times of this track */
if (((ieqxt*.1)-(25.*60.)) < secs1)
    lseek(fd,hpts*RECORDSIZE,1);

/*** all data points in this track before selected time
so move file descriptor to next header and go to next repeat */
else if (((ieqxt*.1)-(25.*60.)) > secs2)
    tflag=1;
else
    printf("track %d
",*strk);
    printf("%d: checking opar %d
",i,opar[0]);

/*** save number of 1st repeat selected for output */
if (*strk == 0) *rpl = ctrk;

/*** increment counter for number of tracks selected */
*strk = *strk+1;

/*** check dimension size **************/
if (*strk > MAXTRK) {
    printf("Numbers of tracks selected %d exceed dimension %d\n", *strk,MAXTRK);
    exit(1);
}

/*** some or all points in this repeat are at selected time
read data and select points of desired lat and lon
convert longitude to float */
for (i=0; i<i2-il+1; i++) {
    time[*strk-1][i]=32767.;
    for (j=0; j< *npar; j++)
        spar[j][*strk-1][i]=32767.;
}

/*** convert equatorial xing time to seconds */
eqxt[*strk-1]=ieqxt*.1;

/*** convert equatorial xing longitude to degrees */
eqxl[*strk-1]=ieqxl*1.e-6;

/*** initialize data point counters */
epts=0;
inpts[*strk-1]=0;

/*** loop thru all longitudes to convert to actual longitude
first repeat of track file */

/*** loop to read data records */
for (i=0; i<hpts; i++)
{
    if ((rdrec(fd,&index,&itime,&ih1,&ih2,&iswh,&isnau,&flag,&rf)) != RECORDSIZE) {
        printf("extgfc2-e-read error at read line %d\n",i);
    }
exit(1); }
/* if edit */
fprintf(fp3,"%f %f %d\n",lat[index-1],lon[index-1],iswh);
if (rf == -2)
fprintf(fp4,"\n",cstrk,i+1);
/* if edit option is in effect save record only if rf>=0 */
if (*edit == 0 || rf > 0)
{
if (index-1 >= il && index-1 <= i2) { inpts[*strk-1]++;
/***************** check special case when track crosses meridian
************ and only top and bottom part of track is needed */
if (*rmxln < *rmnln || lon[index-1] < *rmxln) {
if (itime == 32767)
printf("extgfc2-w-data time of 32767 encountered\n");
else
time[*strk-1][index-1-il]=(itime*1.e-3)+eqxt[*strk-1];
/********************* loop thru to store all selected parameters */
for (j=0; j< *npar; j++)
{
if (opar[j] == 1)
spar[j][*strk-1][index-1-il]=ih1;
else if (opar[j] == 2)
spar[j][*strk-1][index-1-il]=ih2;
else if (opar[j] == 3)
spar[j][*strk-1][index-1-il]=iswh;
else if (opar[j] == 4)
spar[j][*strk-1][index-1-il]=isnau;
else if (opar[j] == 5)
spar[j][*strk-1][index-1-il]=flag;
else
printf("extgfc2-w-Warning undefined parameter chosen\n");
} /* end of for j loop */
} /* end of if (*rmxln < *rmnln || lon[index-1] < *rmxln) */
else
epts++;
/**************** counts number of points in middle of track thrown out */
} /* end of if (index-1 >= il && index-1 <= i2) */
} /* end of if (*edit == 0 || rf != -1) */
} /*end for loop for reading all data records in repeat */
/**************** select lat-lon for output and determine number of points to
************ be output */
if (epts == 0) npts=i2-il+1;
else npts = i2-il-npts;
*wnpts=0;

for (i=il; i<i2; i++)
{
if (*rmxln < *rmnln || lon[i] < *rmxln) {
olat[i-il]=lat[i];
olon[i-il]=lon[i];
*wnpts = *wnpts+1;
} /* if (*rmxln < *rmnln || lon[i] < *rmxln) */
if (*wnpts != (npts))
    printf("extgfc2-w-points written and counted do not correspond %d %d\n",
        *wnpts,npts);

/***************************************************************
    printf("number of pts output %d \n",npts);
/***************************************************************
} /* end of if-elseif-else structure which finds repeats with
correct times */

if (tflag == 1) break; /* goto next track file */
} /* end of while loop for reading all repeats in one track file*/

if (tflag == 1) break; /* exit for loop */
} /* end for(idir=0 */

printf("extgfc2-i-Number of repeats output is %d\n",*strk);
close(fd); /* close current track file */

} /*end if (il != 0 && i2 !=0) */
else /* print message and return to calling routine */
    {printf("extgfc2-i-No data points between specified lat and lon in file %s\n",
        tfile);
close(fd);}
} /* end of file encountered in ifile */

else /* return to calling routine with end of file code */
    { *
        *ier = (-9);
    /* check counter and print out info if gt 0 */
        if (nidx > 0) printf("extgfc2-i-out-of-range index value(s): %d\n",nidx);
        if (nti > 0) printf("extgfc2-i-out-of-range time value(s): %d\n",nti);
        if (nhl > 0) printf("extgfc2-i-out-of-range height value(s): %d\n",nhl);
        if (nswh > 0) printf("extgfc2-i-out-of-range swh value(s): %d\n",nswh);
        if (nsnau > 0) printf("extgfc2-i-out-of-range snau value(s): %d\n",nsnau);
    } /* close else */
} /* end of extgfc2() */

/***************************************************************
    read each data record of gridded data set track file
    checks data ranges and
    returns nbytes the number of bytes read */
rdrec(fd,ptrl,ptr2,ptr3,ptr4,ptr5,ptr6,ptr7,rflag)
int fd,*ptr2,*rflag;
short int *ptr1,*ptr3,*ptr4,*ptr5,*ptr6;
float *ptr7;
{ 
    int nbytes=0;
    *rflag = 0;
    /* read index and check value */
    nbytes=read(fd,ptr1,2);
    if (*ptr1 < 1 || *ptr1 > 2915){
nidx++;
    /* printf("extgfc2-w-index out-of-range: %d",*ptr1);*/
    *rflag = 1;
    /* read time and check value */
    nbytes+=(read(fd,ptr2,4));
    if (*ptr2 < -1530000 || *ptr2 > 1530000){
        nti++;
        /* printf("extgfc2-w-out-of-range time value encountered: %d",*ptr2);*/
        *rflag = 1;
    }
    /* read ht with tidal correction and check value */
    nbytes+=(read(fd,ptr3,2));
    if ((*ptr3 < -14000 || *ptr3 > 10000) && *ptr3 != 32767){
        nhl++;
        /* printf("extgfc2-w-ssh (tide correction) out-of-range: %d",*ptr3);*/
        *rflag = 1;
    }
    /* read ht w/o tidal correction; range not checked */
    nbytes+=(read(fd,ptr4,2));
    /* read swh and check value */
    nbytes+=(read(fd,ptr5,2));
    if ((*ptr5 < 0 || *ptr5 > 2000) && *ptr5 != 32767){
        nswh++;
        /* printf("extgfc2-w-swh out-of-range: %d",*ptr5);*/
        *rflag = -1;
    }
    /* read snau and check values */
    nbytes+=(read(fd,ptr6,2));
    if ((*ptr6 < 0 || *ptr6 > 6400) && *ptr6 != 32767){
        nsnau++;
        /* printf("extgfc2-w-snau out-of-range: %d",*ptr6);*/
        *rflag = -2;
    }
    /* read flag; range not checked */
    nbytes+=(read(fd,ptr7,1));
    return nbytes;
}
getbits4.c Thu Oct 4 11:23:39 1990 1

c************* Getbits ***********
c
Used in program extqfc2.c and in
subroutine output.c.
c
c
getbits_ (x,p,n)
unsigned *p,*n;
unsigned int *x;
{
    return((*x>>(*p+l-*n)) & -0<<n));
}

getbits(x,p,n)
unsigned p,n;
unsigned int *x;
{
    return((*x>>(p+1-n)) & -0<<n));
}
output.c Thu Oct 4 10:49:08 1990 1

/********** Subroutine output (A. Hayashi 8/16/88) **********/

/* Outputs to file selected data from extgfc() */

#include <stdio.h>
#define MAXPTS 2915
#define MAXTRK 65
#define NPAR 5

output_(ofile, olat, olon, time, spar, eqxl, eqxt, npts, inpts, nrtrk, itrkl, npar, oflag, maxpts, maxtrk, maxp, ll)

/* ofile - name of file to output data to (any length character variable)
   npar - number of parameters selected for output
   nrtrk - number of repeats which were selected for output
   npts - number of data records in each repeat selected for output
   inpts - array containing number interpolated records in each repeat
   eqxl - array containing equatorial crossing longitude of each repeat
   olat - array containing latitudes of each corresponding data record
  olon - array containing longitudes of each corresponding data record
   eqxt - array containing equatorial xing time of each repeat
   time - array containing time of each data record in each repeat
   spar - array containing parameters selected for output
   itrkl - repeat number of 1st repeat selected for output
   maxtrk - maximum dimension size of index corresponding to repeat
   maxpts - maximum dimension size of index corresponding to data point
   maxp - maximum dimension size of index corresponding to parameter

   Note: currently variable array dimensions cannot be declared in
   this program due to the compiler, so dimensions passed thru
   argument list (maxtrk,maxpts) must match the ones defined in
   the subroutine (MAXTRK,MAXPTS). NPAR and maxp need not match
   since it is not a required dimension. C arrays are row major.
*/

/* declare arguments */
char *ofile;
float olat[],olon[],spar[][MAXTRK][MAXPTS],eqxl[];
double time[][MAXPTS],eqxt[];
int *npts,inpts[],*nrtrk,*itrkl,*npar,*maxpts,*maxtrk,*maxp,*oflag,11;

{ /* declare local variables */

   int i,j,k,p,getbits();
   FILE *fp,*fopen();

/* check dimension - currently variable dimension sizes cannot be
 ** passed thru for MAXPTS and MAXTRK */

   if (*maxpts != MAXPTS) {
       printf("output-e-data point dimension must equal 2915\n");
       exit(1);  }

   if (*maxtrk != MAXTRK) {
       printf("output-e-repeat track dimension must equal 65\n");
       exit(1);  }

/* end all strings passed thru with null string terminator */
for (i=0; ofile[i] != '\0'; i++)
    ofile[i]='\0';

/* open output file */
if ((fp=fopen(ofile,"a")) == NULL) {
    printf("output-e-Can't open file %s\n",ofile);
    exit(1);
}

printf("output-i-Opening file %s for output\n",ofile);
/* write out data */
for (i=0; i< *nrtrk; i++)
{
    fprintf(fp,"%d %d %f %13.1f\n",*itrkl+i,*npts,inpts[i],eqxl[i],eqxt[i]);
    for (j=0; j< *npts; j++)
    {
        fprintf(fp,"%13.3f %10.6f %10.6f",time(i)[j],olat[j],olon[j]);
        for (k=0; k< *npar; k++)
        {
            if (k != *oflag && k != -1)
                fprintf(fp," %6.0f",spar[k][i][j]);
            else if (k == *oflag)
            {
                fprintf(fp,"'");
                if (time(i)[j] != 32767.)
                    for (p=24; p<32; p++) fprintf(fp, "%d", getbits(spar[k][i]+j,p,1));
                else
                    fprintf(fp,"00000000");
            }
        }
    }
    fprintf(fp,"\n");
}
fclose(fp);
subgf5.f

Thu Oct  4 11:16:16 1990 1

c********** Subroutine parout()********** **************
c Called by extqfc2.f. Determines which parameters
c are requested as output

subroutine parout(param,npar,opar)
character param*80,cpar*10
integer opar(5)

c ... search thru param and separate request
write(6,*)
i=1
j=0
npar=0
1 if (param(i:i).ne.',' .and. param(i:i).ne.' ') then
  j=j+1
  cpar(j:j)=param(i:i)
i=i+1
goto 1
else
  c... one request separated, increment counter and set opar
  if (cpar(1:j).eq.'hl') then
    npar=npar+1
    opar(npar)=1
    goto 2
  endif
  if (cpar(1:j).eq.'h2') then
    npar=npar+1
    opar(npar)=2
    goto 2
  endif
  if (cpar(1:j).eq.'swh') then
    npar=npar+1
    opar(npar)=3
    goto 2
  endif
  if (cpar(1:j).eq.'snau') then
    npar=npar+1
    opar(npar)=4
    goto 2
  endif
  if (cpar(1:j).eq.'flag') then
    npar=npar+1
    opar(npar)=5
    goto 2
  endif
  if (cpar(1:j).eq.'O') then
    npar=0
    write(6,*)'no parameters chosen'
goto 99
  endif
  write(6,*)'Unidentifiable parameter',cpar
  stop 'Program terminated.'
  endif
2 write(6,100) cpar(1:j)
100 format(2x,'Parameter',a6,' selected')
3 if (param(i:i).eq.' ' .or. npar.eq.5) then
  return
else

85
subgf5.f

Thu Oct 4 11:16:16 1990

i=i+1
j=0
goto 1
endif
99
return
end

c function subroutine seconds() which converts year,month,day to
seconds for data selection
real*8 function seconds(d)
character d*6
real*8 year,month,day,xdays
read(d(1:2),'(f2.0)') year
read(d(3:4),'(f2.0)') month
read(d(5:6),'(f2.0)') day
if (year.le.85.) then
  write(6,*)'extgf6-i-GEOSAT data starts from Nov. 8 1986'
  year=85.
endif
  c ... seconds in a year
  seconds=(year-85.)*365.*24.*3600.
  c ... leap year adjustment
  if (year.ge.89) seconds=seconds+(24.*3600.)
  c ... add seconds in the previous months
  seconds=seconds+(xdays(month-1,year)*24.*3600.)
  c ... add seconds in the days
  seconds=seconds+((day-1)*24.*3600.)
return
end

c function subroutine which determines how many days there are
in the year up to month specified
real*8 function xdays(month,year)
real*8 month
if (month.eq.0.) xdays=0
if (month.eq.1.) xdays=31.
if (month.eq.2.) xdays=59.
if (month.eq.3.) xdays=90.
if (month.eq.4.) xdays=120.
if (month.eq.5.) xdays=151.
if (month.eq.6.) xdays=181.
if (month.eq.7.) xdays=212.
if (month.eq.8.) xdays=243.
if (month.eq.9.) xdays=273.
if (month.eq.10.) xdays=304.
if (month.eq.11.) xdays=334.
if (month.eq.12.) xdays=365.
  c ... leap year correction
  if (year.ge.88. .and. month.gt.2) xdays=xdays+1
return
end
/*************** subroutine xdays() ***************
   Determines how many days there are in the year up to month specified */

double xdays(month,year)
int month,year;
{
   double days;
   if (month == 0.) days=0.;
   if (month == 1.) days=31.;
   if (month == 2.) days=59.;
   if (month == 3.) days=90.;
   if (month == 4.) days=120.;
   if (month == 5.) days=151.;
   if (month == 6.) days=181.;
   if (month == 7.) days=212.;
   if (month == 8.) days=243.;
   if (month == 9.) days=273.;
   if (month == 10.) days=304.;
   if (month == 11.) days=334.;
   if (month == 12.) days=365;
   /* leap year correction */
   if (year >= 88. && month >= 2.) days=days+1;
   return(days);
}
subroutine seconds()

Converts year, month, day to seconds
from reference for data selection /\n
double secs(char d)
{
    int i,n;
    int year,month,day;
    double seconds, xdays();
    /* convert characters to numbers */
    year=0;
    for (i=0; i<=1; i++)
        year = 10 * year + d[i] - '0';
    month=0;
    for (i=2; i<=3; i++)
        month = 10 * month + d[i] - '0';
    day=0;
    for (i=4; i<=5; i++)
        day = 10 * day + d[i] - '0';
    /* print information message if required */
    if (year < 85.) {
        printf("extgfc-i-GEOSAT data starts from Nov. 8, 1986\n");
        year=85.;
    }
    /* seconds in years */
    /* leap year adjustment */
    if (year >= 89.) seconds=seconds+ (24.*3600.);
    /* add seconds in the previous months */
    seconds=seconds+(xdays(month-1,year)*24.*3600.);
    /* add seconds in the days */
    seconds=seconds+((day-1)*24.*3600.);
    return(seconds);
}
c Program Bias. Compute the EM bias value by applying
the SAXON algorithm. Corrects for deviations of orbit
c from exact repeat path (+/- 1 km), and finds actual
c longitude of each point by finding difference between
c crossing lon of 1st repeat and nth repeat.
c********************************************************
ccharacter flag*8,infile*8,repeat*2
integer*2 i,npts,inpnts,rtk,nfiles,junk
real*8 time,ect
real*4 bias,lat,lon,ecl,xlst,corr,swh,snau
data c1/-0.0163/,c2/-0.002790/,c3/-2.15/
open(2, file='/home/anne/repeat.number')
do 200 m=1,24
   read(2,'(a)')repeat
   c Input file in form of number of files,
c name of first file, equatorial crossing
c longitude of first files, name of second
c file, etc.
   open(22, file='/home/anne/input')
   read(22,*nfiles
   do 90 j-l,nfiles
      read(22,' (a)')infile
      read(22,*xilst
c Directory Getdata contains the files
   c generated as output from the GETDATA program.
   open(10, file='/home/anne/Getdata/'//infile)
   open(20, file='/home/anne/Bias/Rpt//'//repeat//'//A0.'//infile)
   open(30, file='/home/anne/Flagtypes/Rpt//'//repeat//'//A0.'//infile)
   open(33, file='/home/anne/Flagpoints/Rpt//'//repeat//'//A0.'//infile)
c Suffix ".A0" refers to ascending tracks with crossing
   c longitudes between 0 and 99 degrees.
c (Possibilities: A0, A1, A2, A3, D0, D1, D2, D3)
11   read(10,*end=90)rtk,npts,inpnts,ecl,ect
   if (rtk.eq.m) then
      go to 12
   else
      do 4 k=1,npts
         read(10,*end=11)time, lat, lon, hl, swh, snau, flag
         continue
      go to 11
   endif
   c write header
12   write(20,21)rtk,npts,inpnts,ecl,ect
21   format(i3,i5,i5,f12.6,f12.1)
c correct for repeat track longitude
   if (rtk.eq.1) then
      corr=0
   else
CORR=ECL-XLST
ENDF

13 DO 80 I=1,NPTS
READ(10,*,END=81)TIME,LAT,LON,SWH,SNAU,FLAG
C SKIP FLAGGED DATA AND COUNT NUMBER OF FLAGS
IF(SNAU.EQ.0.0)THEN
WRITE(6,'(A)')'DIVISION BY 0.0 (SIGNAUGHT) IN FILE',INFILE
ENDIF
IF(FLAG(7:7).EQ.'1')THEN
ATTCNT=ATTCNT+1
WRITE(33,34)LAT,LON,FLAG
GO TO 80
ENDIF
IF(TIME.EQ.32767)THEN
TIMECNT=TIMECNT+1
GO TO 80
ENDIF
IF(HL.EQ.32767)THEN
HCNT=HCNT+1
GO TO 80
ENDIF
IF(SWH.EQ.32767)THEN
SWHCNT=SWHCNT+1
GO TO 80
ENDIF
IF(SNAU.EQ.32767)THEN
SNAUCNT=SNAUCNT+1
GO TO 80
ENDIF
C COUNT BUT KEEP OTHER FLAGGED DATA
IF(FLAG(1:1).EQ.'1')THEN
WRITE(33,34)LAT,LON,FLAG
GO TO 40
ENDIF
IF(FLAG(2:2).EQ.'1')THEN
WRITE(33,34)LAT,LON,FLAG
GO TO 40
ENDIF
IF(FLAG(3:3).EQ.'1')THEN
WRITE(33,34)LAT,LON,FLAG
GO TO 40
ENDIF
IF(FLAG(4:4).EQ.'1')THEN
WRITE(33,34)LAT,LON,FLAG
GO TO 40
ENDIF
IF(FLAG(5:5).EQ.'1')THEN
WRITE(33,34)LAT,LON,FLAG
GO TO 40
ENDIF
IF(FLAG(6:6).EQ.'1')THEN
WRITE(33,34)LAT,LON,FLAG
GO TO 40
ENDIF
IF(FLAG(7:7).EQ.'1')THEN
WRITE(33,34)LAT,LON,FLAG
GO TO 40
ENDIF
IF(FLAG(8:8).EQ.'1')THEN
WRITE(33,34)LAT,LON,FLAG
GO TO 40
ENDIF
endif
goto 41

40 other=other+1

c change swh to meters and snau to dB (from cm and .01dB)
41 swh=swh/100
    snau=snau/100
    bias=c1*swh+c2*(swh**2)+(c3*swh)/(snau*snau)
lon=lon+ corr
    write(20,50)time,lat,lon,swh,snau,bias
50 format(f12.3,f12.6,f12.6,f10.4,f10.4,f12.4)
80 continue
81 close(20)

c write out flag counts
81 write(30,'(a)')'Input file name',infile
    write(30,*)'Points processed = ',npts
    write(30,*)'Points output = '
    *npts-(attcnt+swhcnt+snaucnt+timecnt+hcnt)
    write(30,*)'Attitude flags=',attcnt
    write(30,*)'hl flags = ',hcnt
    write(30,*)'swh flags = ',swhcnt
    write(30,*)'snau flags = ',snaucnt
    write(30,*)'time flags = ',timecnt
    write(30,*)'other flags (of pts kept) = ',other
34 format(f12.6,f12.6,2x,a)
    attcnt=0
    swhcnt=0
    hcnt=0
    snaucnt=0
    timecnt=0
    other=0
90 continue
    close(22)
200 continue
100 stop
end
c ************* Program BoxAve ***********************
c Used to smooth data using 5 point running boxcar mean .

real swh(300),smswh(300),bias(300),smbias(300)
real snau(300),smsnau(300),lat(300),lon(300)

c load file to be smoothed
j=0
do 100 i=1,300
read(5,*,end=101) junk, lat(i), lon(i), swh(i), snau(i), bias(i)
j=j+1
write(6,*) i, j
100 continue
101 do 200 i=1,j
if(i.eq.1) go to 148
if(i.eq.j) go to 149
if(i.eq.2) go to 150
if(i.eq.(j-1)) go to 150
smswh(i)=(swh(i-2)+swh(i-1)+swh(i)+swh(i+1)+swh(i+2))/5.
smsnau(i)=(snau(i-2)+snau(i-1)+snau(i)+snau(i+1)+snau(i+2))/5.
smbias(i)=(bias(i-2)+bias(i-1)+bias(i)+bias(i+1)+bias(i+2))/5.
go to 200
148 smswh(i)=(swh(i)+swh(i+1))/2.
smsnau(i)=(snau(i)+snau(i+1))/2.
smbias(i)=(bias(i)+bias(i+1))/2.
go to 200
149 smswh(i)=(swh(i)+swh(i-1))/2.
smsnau(i)=(snau(i)+snau(i-1))/2.
smbias(i)=(bias(i)+bias(i-1))/2.
go to 200
150 smswh(i)=(swh(i-1)+swh(i)+swh(i+1))/3.
smsnau(i)=(snau(i-1)+snau(i)+snau(i+1))/3.
smbias(i)=(bias(i-1)+bias(i)+bias(i+1))/3.
go to 200
200 continue

do 300 i=1,300
write(6,*,end=99) lat(i), lon(i), smswh(i), smsnau(i), smbias(i)
300 continue
99 stop
end
Program BlkAve - Block average along track data every 7 seconds (= 50 km) and saves standard deviations.

Program is set up to block average the output of the Bias program, and uses the same input file as the one used in the Bias program.

```fortran
character infile*7, repeat*2
integer*2 npts, inpnts, rtk, nfiles, count, junk
real*8 time, ect, end, start
real*4 bias, lat, lon, ecl, avelat, avelon, aveswh, avesnau
real*4 varwh, varbias, varvsnau, x
real*4 stdwh, stdbias, stdvsnau, dbstdvsnau
real*4 avebias, snau, swh, vsnau, dbavesnau
real*4 ibias(8), iswh(8), ivsnau(8)

open(2, file='/home/anne/repeat.number')
do 200 m=1, 24
   read(2, '(a)') repeat
   c Use same input file as the one used in the bias program, just skip the lines containing equator crossing info.
   open(5, file='/home/anne/input.AO')
   c Suffix "AO" refers to all ascending tracks with equatorial crossings between 0 and 99.
   read(5, *) nfiles
   do 90 j=1, nfiles
      read(5, '(a)') infile
      read(5, '*') junk
      open(10, file='/home/anne/Bias/Rpt'//repeat//'/bias.'//infile)
      open(20, file='/home/anne/Ave/Rpt'//repeat//'.AO')
      open(60, file='/home/anne/Std/Rpt'//repeat//'.AO')
      c read header
      read(10, *, end=90) rtk, npts, inpnts, ecl, ect
      read(10, *, end=90) time, lat, lon, swh, snau, bias
      if (lon.gt.180.) lon=lon-360.
      c change snau in dB to V2 so that averaging is linear
      vsnau=10**(snau/10)
      4   start=time
      end=start+7.096
      avelat=0
      avelon=0
      aveswh=0
      avesnau=0
      avebias=0
      varwh=0
      varvsnau=0
      varbias=0
      count=0
      count=count+1
      ibias(count)=bias
      iswh(count)=swh
      ivsnau(count)=vsnau
      avelat=avelat+lat
```
avelon=avelon+lon
aveswh=aveswh+swh
avesnau=avesnau+vsnau
avebias=avebias+bias
read(10,*,end=99)time,lat,lon,swh,snau,bias
if(lon.gt.180.)lon=lon-360.
snau=10**(snau/10)
if(time.1t.end)then
   goto 3
else
   avelat=avelat/count
   avelon=avelon/count
   aveswh=aveswh/count
   avesnau=avesnau/count
   avebias=avebias/count
   c find standard deviations
   do 47 n=1,count
      varsw=varsw+(iswh(n)-aveswh)**2
      varvsnau=varvsnau+(ivsnau(n)-avesnau)**2
      varbias=varbias+(ibias(n)-avebias)**2
   continue
   x=float(count)
   stdswh=(1/x)*(varsw)**0.5
   stdvsnau=(1/x)*(varvsnau)**0.5
   stdbias=(1/x)*(varbias)**0.5
   c convert avesnau back to dBs
   c but leave std in terms of volts (otherwise get negative large number)
   dbavesnau=10*(log10(avesnau))
   if(avelon.lt.0.)avelon=avelon+360.
   write(20,50)avelat,avelon,aveswh,dbavesnau,avebias
   write(60,50)avelat,avelon,stdswh,stdvsnau,stdbias
   format(f12.6,f12.6,f9.4,f9.4,f9.5)
   goto 4
99 goto
   avelat=avelat/count
   avelon=avelon/count
   aveswh=aveswh/count
   avesnau=avesnau/count
   avebias=avebias/count
   c find standard deviations
   do 66 n=1,count
      varsw=varsw+(iswh(n)-aveswh)**2
      varvsnau=varvsnau+(ivsnau(n)-avesnau)**2
      varbias=varbias+(ibias(n)-avebias)**2
   continue
   x=float(count)
   stdswh=(1/x)*(varsw)**0.5
   stdvsnau=(1/x)*(varvsnau)**0.5
   stdbias=(1/x)*(varbias)**0.5
   dbstdswh=10*(log10(stdswh))
   if(avelon.lt.0.)avelon=avelon+360.
   write(20,50)avelat,avelon,aveswh,dbavesnau,avebias
   write(60,50)avelat,avelon,stdswh,stdvsnau,stdbias
90 goto
200 continue
100 stop
end
Program Rmean - Computes long term (24 repeats) average height profile of a track, then removes that mean from the height profile of the chosen repeat cycle. The difference is the residual height or dynamic height profile.

character infile*7, repeat*2
integer*2 npts,inpnts,rtk,bad(2000),nfiles,size,q,sizefile,xcnt
real*8 time,ect
real*4 sizeave(51),xlat,xlon,xdh,x,xavelat,xavedh,xavelon

First, find long term average profile.

open(4,file='~/home/anne/input')

Input file form: number of files, name of first file, number of points in first file, equatorial crossing longitude of first file, same info for second file, etc.

read(4,*nfiles
do 200 q=1,nfiles
  sizeave(q)=0
  read(4, '(a)')infile
  read(4,*size
  read(4,*xlst
  open(5, file='~/home/anne/Getdata/'//infile)
  open(6, file='~/home/anne/H/ave.'//infile)
  do 5 i=1,size
    bad(i)=0
    sumlon(i)=0.
    sumh(i)=0.
  5 continue
  do 10 i=1,24
    read(4,*nfiles
  do 20 k=1,size
    c load data
    read(5,*rtk,npts,inpnts,ecl,ect
do 20 k=1,size
    c Include average data point value only
    c if at least 8 values were used to compute the average.
    do 30 k=1,size
if(bad(k).gt.16)then
  n=n+1
  go to 30
endif
aveh(k)=sumh(k)/(24-bad(k))
c***note h is left in terms of cm *****
avelon(k)=sumlon(k)/(24-bad(k))
write(6,50)lat(k),avelon(k),aveh(k)
50  format(f12.6,f13.6,f14.4)
30 continue
continue
sizeave(q)=size-n
n=0
200 continue
Close(4)
c**Removal of computed long term average from chosen repeat.
open(4, file='home/anne/input')
read(4,*nfiles
do 400 q=1,nfiles
  read(4,'(a)')infiltr
  read(4,*sizefile
  read(4,*xlst
open(5, file='home/anne/Getdata/'//infiltr)
open(10, file='home/anne/average.'//infiltr)
if(sizeave(q).eq.0)goto 400
do 103 i=1,sizeave(q)
c load average profile
  read(10,*rlat(i),avelon(i),aveh(i)
103 continue
do 105 i=1,sizefile
  bad(i)=0
105 continue
c specified repeat cycle = 13
  do 110 i=1,13
    read(5,*rtk,npts,inpnts,ecl
open(22, file='repeat.list')
read(22,'(a)')repeat
  if(i.ne.13)then
    do 107 k=1,sizefile
  endif
  open(20, file='home/anne/H/Rpt//repeat//'//infiltr)
  j=0
  if(rtk.eq.1)then
    corr=0
  else
    corr=ecl-xlst
  endif
  do 120 k=1,sizefile
    l=k-j
    read(5,*time,lat(k),lon(k),h(k)
    if(rlat(l).ne.lat(k))then
      bad(k)=bad(k)+1
      j=j+1
      go to 120
    endif
    if(h(k).eq.32767.)then

bad(k)=bad(k)+1
endif
don(k)=lon(k)+corr
rmh(k)=h(k)-aveh(1)
write(20,21)time,lat(k),lon(k),rmh(k)
21    format(f12.3,f14.6,f14.6,f12.2)
120  continue
    close(20)
110  continue
j=0
    close(22)
400 continue
    close(4)
99    stop
end
c **********Program Rpoly**********
c Fits 2nd degree polynomial to dynamic wave height data (after mean has been removed).
c Uses IMSL subroutine with weighting (wt = 1/variance).
c Fit is of form y=a0+(a1)x+(a2)x**2
c Then removes polynomial fit from each data point.

character infile*7
real*8 ht(531),d(531),c(5),s(5),a(2),b(2),rsq
real*8 p(2124),t(12),wt(531),junk,inwt(531)
real a0,a1,a2,h,newh,pfit,lat,lon
integer id,ier,md,n,nfiles

open(5,file='3064.rm',status='unknown')
do 10 i=1,531
   read(5,*)lat,lon,ht(i),inwt(i)
d(i)=i
   wt(i)=l./ (inwt(i)) **0.5
10 continue
close (5)
c call subr RLOTW to fit data with polyn with weighting
******call rlfotw(x,y,n,rsq,md,w,id,p,c,s,a,b,ier)
c x=array, distance along track
c y=ht(i), n= #data pts, rsq=100.(not important)
c md=max degree=2, w=weight array=var(i)
c id=output degree, p=work array(4*n)
c c=output array(md+3)=(5), s=output array(md+3)=(5)
c a=output array(md)=(2), b= output array(md)=(2)
c ier=error message
********
n=531
md=2
rsq=100.
call rlfotw(d,ht,n,rsq,md,wt,id,p,c,s,a,b,ier)
c call sub RLDOPM to decode the polyn coeffs from above
******call rldopm(c,id,a,b,t)
c c=input above c, outputs the true coeffs, id=same=2,
c a=input above a, b=b above
ct=doble precision work array (4*(id+1))=(12)
c ******
call rldopm(c,id,a,b,t)
a0=c(1)
a1=c(2)
a2=c(3)
c Remove polynomial fit from each height difference file.
c Functional dependence on distance along track, taken
c as point number.
c Input file of form: number of files,
c list of names of files.
open(10, file='input')
read(10,*) nfiles
    do 200 n=1, nfiles
read(10, '(a)') infile

open(5, file='/home/anne/gulf/'// infile)
open(6, file=infile//'.rp')
    do 100 i=1, 600
read(5, *, end=101) lat, lon, h
    pfit = a0 + (a1*i) + (a2)*(i**2)
newh = h - pfit
write(6,*) i, lat, lon, newh
100 continue
101 close(6)
200 continue
stop
end
Program Zgrid

Calls zgrid subroutine to grid Geosat data.
Important to dimension all arrays to exact size
array to contain gridded data lon: 260-360, lat: 0-65

real z2(52,34), latp2(7417), lonp2(7417), zp2(7417), junk
real latpl(7072), lonpl(7072), zl(101,66), zpl(7072)
real land
land=.9E35
call abrupt_underflow

2x2 degree gridding
open (6, file='/home/anne/Grids/Rpt1.swu')
do 180 n=1,4
  open(5, file='/home/anne/FinalAve/Rpt1.dat')
do 10 i=-1,7417
    read(5, *, end=ll) latp2(i), lonp2(i), zp2(i)
c snau
    read(5, *, end=ll) latp2(i), lonp2(i), junk, zp2(i)
c bias
    read(5, *, end=ll) latp2(i), lonp2(i), junk, zp2(i)
10 continue
close(5)
do 20 j=1,34
  do 20 i=1,52
    z2(i,j)=0.
20 continue
c call zgrid2(z,nx,ny,xl,yl,dx,dy,xp,yp,zp,n,cayin,nrng)
if (n.eq.1) call zgrid2(z2,52,34,259.5,-0.5,2.0,2.0,lonp2,latp2,zp2,7417,0,2)
if (n.eq.2) call zgrid2(z2,52,34,259.5,0.5,2.0,2.0,lonp2,latp2,zp2,7417,0,2)
if (n.eq.3) call zgrid2(z2,52,34,260.5,-0.5,2.0,2.0,lonp2,latp2,zp2,7417,0,2)
if (n.eq.4) call zgrid2(z2,52,34,260.5,0.5,2.0,2.0,lonp2,latp2,zp2,7417,0,2)

c output of 2x2 is input of 1x1 gridin
do 100 j=1,34
  do 70 i=1,52
    if (z2(i,j).gt.1000.) go to 69
    if (n.eq.1) then
      lonpl((j-1)*52+i)=259.5+(i-1)*2
      latpl((j-1)*52+i)=-0.5+(j-1)*2
      zpl((j-1)*52+i)=z2(i,j)
    elseif (n.eq.2) then
      lonpl(3536+(j-1)*52+i)=259.5+(i-1)*2
      latpl(3536+(j-1)*52+i)=-0.5+(j-1)*2
      zpl(3536+(j-1)*52+i)=z2(i,j)
    elseif (n.eq.3) then
      lonpl(5304+(j-1)*52+i)=260.5+(i-1)*2
      latpl(5304+(j-1)*52+i)=-0.5+(j-1)*2
      zpl(5304+(j-1)*52+i)=z2(i,j)
    elseif (n.eq.4) then
      lonpl(1768+(j-1)*52+i)=260.5+(i-1)*2
      latpl(1768+(j-1)*52+i)=-0.5+(j-1)*2
      zpl(1768+(j-1)*52+i)=z2(i,j)
  end if
100 continue

Zgrid.f
endif
continue
continue
continue

c ************* 1x1 degree gridding

c set undefined areas (land) to .9E35, using land mask map,
c and the rest of zl(i,j) to 0
open(10,file=':/home/anne/Map/atlantic.asc')
do 200 j=1,66
do 200 i=1,101
read(10,*x,y,zl(i,j))
if(zl(i,j).eq.1.)then
zl(i,j)=land
endif
200 continue
c call zgridl(zl,nx,ny,x1,yl,dx,dy,xp,yp,zpl,n,cayin,nrng)
call zgridl(zl,101,66,260.0,0.0,1.0,1.0,lonpl,latpl,zpl,7072,0,10)
c output in raster format
do 250 j=1,66
 k=67-j
do 270 i=1,101
 c change land from .9E35 to zero
  if(zl(i,k).eq.0.9E35)then
   zl(i,k)=0.0
  elseif(zl(i,k).gt.1000.)then
   zl(i,k)=0.0
  endif
270 continue
 c for sigma naught put bounds of 7.0<sigma<19.0 from Chelton
 c elseif(zl(i,k).lt.7.0)then
  c  zl(i,k)=7.0
 c elseif(zl(i,k).gt.19.0)then
  c  zl(i,k)=19.0
 c for bias put maximum bound of 1 m on bias
 c elseif(zl(i,k).lt.-1.0)then
  c  zl(i,k)=-1.0
 endif
270 continue
 c change bias to positive cm
  write(6,280)(-100.*zl(i,k),i=1,101)
  write(6,280)(zl(i,k),i=1,101)
280 format(101f12.5)
250 continue

stop
end
SUBROUTINE ZGRID1(Z,NX,NY,X1,Y1,DX,DY,XP,YP,ZP,N,CAYIN,NRNG)
C
C SETS UP SQUARE GRID FOR CONTOURING, GIVEN ARBITRARILY PLACED I
C DATA POINTS. LAPLACE INTERPOLATION IS USED.
C THE METHOD USED HERE WAS LIFTED DIRECTLY FROM NOTES LEFT BY I
C MR IAN CRAIN FORMERLY WITH THE COMP.SC ENCE IV.
C INFO ON RELAXATION SOLN OF LAPLACE EQN SUPPLIED BY DR T MURTY.
C FORTRAN II OCEANOGRAPHY/EMR DEC/68 JDT
C
C Z = 2-D ARRAY OF HGTS TO BE SET UP. POINTS OUTSIDE REGION TO I
C BE CONTOURED SHOULD BE INITIALIZED TO 10**35. YOU MIGHT SET I
C THE REST OF Z TO 0.0
C NX,NY = MAX SUBSCRIPTS OF Z IN X AND Y DIRECTIONS.
C X1,Y1 = COORDINATES OF Z(1,1).
C DX,DY = X AND Y INCREMENTS.
C XP,YP,ZP = ARRAYS GIVING POSITION AND HGT OF EACH DATA POINT.
C N = SIZE OF ARRAYS XP,YP AND ZP.
C
C MODIFICATION FEB,69 TO GET SMOOTHER RESULTS A PORTION OF THE I
C BEAM EQN WAS ADDED TO THE LAPLACE EQN IVING
C DELTA2X(Z)+DELTA2Y(Z) - K(DELTA4X(Z)+DELTA4Y(Z)) = 0.
C CAYIN = K = AMOUNT OF SPLINE EQN (BETWEEN 0 AND INF.)
C K=0 GIVES PURE LAPLACE SOLUTION. K=INF. GIVES PURE SPLINE SOL.
C NRNG ALL GRID POINTS MORE THAN NRNG GRID SPACES FROM THE I
C CLOSEST DATA POINT ARE SET UNDEFINED (10**35).
C
C
C DIMENSION Z(101,66)
DIMENSION IMNEW(110)
DIMENSION XP(7072),YP(7072),ZP(7072)

WRITE(6,10)

10 FORMAT(/'***SUBROUTINE ZGRID*** ',/PI=3.1415926/)
BIG=.9E35
CAY=CAYIN

GET ZBASE WHICH WILL MAKE ALL ZP VALUES POSITIVE BY AT LEAST
.25(ZMAX-ZMIN) AND FILL IN GRID WITH ZEROS.

ZMIN=ZP(1)
ZMAX=ZP(1)
DO 20 K=2,N
IF(ZP(K)-ZMAX)14,14,12
12 ZMAX=ZP(K)
14 IF(ZP(K)-ZMIN)16,20,20
16 ZMIN=ZP(K)
20 CONTINUE
ZRANGE=ZMAX-ZMIN
ZBASE=ZMIN+.25*ZRANGE
ZUL=ZBASE+ZMAX
ZUL20=ZUL*20.
ZUL400=ZUL*400.
DO 40 I=1,NX
DO 40 J=1,NY
   IF(Z(I,J) -BIG) 30,40,40
30  Z(I,J)=0.
40 CONTINUE
C
C   AFFIX EACH POINT ZP TO NEAREST GRID PT. TAKE AVG IF MORE THAN
C   ONE NEAR PT. ADD ZBASE PLUS 10*ZRANGE AND MAKE NEGATIVE.
C   INITIALLY SET EACH UNSET GRID PT TO VALUE OF NEAREST KNOWN PT
C
DO 110 K=1,N
   I=(XP(K)-X1)/DX+1.5
   IF(I*(NX+1-I)) 70,70,60
60   J=(YP(K)-Y1)/DY+1.5
   IF(J*(NY+1-J)) 70,70,90
C70  WRITE(6,80) K,XP(K),YP(K),ZP(K)
C edit out write line so rename line 70 at "goto 110"
80  FORMAT(' POINT OUT OF RANGE K,X,Y,Z =', I5,3f10.4)
70  GO TO 110
90  IF(Z(I,J)-ZUL400)100,110,110
100  Z(I,J)=Z(I,J)+ZP(K)+ZBASE+ZUL20
110 CONTINUE
C
NPG=0
DO 150 I=1,NX
   DO 150 J=1,NY
      IF(Z(I,J)-BIG) 130,150,150
130  NIJ=Z(I,J)/ZUL20
      IF(NIJ)145,145,140
140  Z(I,J)=Z(I,J)/NIJ+ZUL20-10.*ZRANGE
      GO TO 150
145  Z(I,J)=1.E35
      NPG=NPG+1
150 CONTINUE
C
DO 199 ITER=1,NRNG
   NNEW=0
   DO 197 I=1,NX
      DO 197 J=1,NY
         IF(Z(I,J)+BIG)152,192,192
152  IF(J-1)162,162,153
153  IF(JMNEW)154,154,162
154  ZIJN=ABS(Z(I,J-1))
         IF(ZIJN-BIG)195,162,162
162  IF(I-1)172,172,163
163  IF(IMNEW(J))164,164,172
164  ZIJN=ABS(Z(I-1,J))
         IF(ZIJN-BIG)195,172,172
172  IF(J-NY)173,182,182
173  ZIJN=ABS(Z(I,J+1))
         IF(ZIJN-BIG)195,182,182
182  IF(I-NX)183,192,192
183  ZIJN=ABS(Z(I+1,J))
         IF(ZIJN-BIG)195,192,192
192  IMNEW(J)=0
      JMNEW=0
      GO TO 197
195  IMNEW(J)=1
      JMNEW=1

103
Z(I,J)=ZIJN
NNEW=NNEW+1
197 CONTINUE
   IF(NNEW) 200,200,199
199 CONTINUE
200 CONTINUE
   DO 202 I=1,NX
   DO 202 J=1, NY
   ABZ=ABS(Z(I,J))
   IF(ABZ-BIG)202,201,201
201 Z(I,J)=ABZ
202 CONTINUE

C
C IMPROVE THE NON-DATA POINTS BY APPLYING POINT OVER-RELAXATION
C USING THE LAPLACE-SPLINE EQUATION (CARRES METHOD IS USED)
C
DZRMS=ZRANGE
RELAX=1.0
EPS=.01
ITMAX=100
DO 2100 ITER=1,ITMAX
   DZRMS=0.
   DZMAX=0.
   DO 2000 I=1,NX
   DO 2000 J=1, NY
   Z00=Z(I,J)
   IF(Z00-BIG)205,2000,2000
205 IF(Z00)2000,208,208
208 WGT=0.
   ZSUM=0.
500 IM=0
   IF(I-1)570,570,510
510 ZIM=ABS(Z(I-1,J))
   IF(ZIM-BIG)530,570,570
530 IM=1
   WGT=WGT+1.
   ZSUM=ZSUM+ZIM
   IF(I-2)570,570,540
540 ZIMM=ABS(Z(I-2,J))
   IF(ZIMM-BIG)560,700,700
560 WGT=WGT+CAY
   ZSUM=ZSUM-CAY*(ZIMM-2.*ZIM)
570 IF(NX-I)700,700,580
580 ZIP=ABS(Z(I+1,J))
   IF(ZIP-BIG)600,700,700
600 WGT=WGT+1.
   ZSUM=ZSUM+ZIP
   IF(IM)620,620,610
610 WGT=WGT+4.*CAY
   ZSUM=ZSUM+2.*CAY*(ZIM+ZIP)
620 IF(NX-I)700,700,630
630 ZIPP=ABS(Z(I+2,J))
   IF(ZIPP-BIG)650,700,700
650 WGT=WGT+CAY
   ZSUM=ZSUM-CAY*(ZIPP-2.*ZIP)
700 CONTINUE
1500 JM=0
1510 ZJM=ABS(Z(I,J-1))
   IF(ZJM-BIG)1530,1570,1570
1530 JM=1
   WGT=WGT+1.
   ZSUM=ZSUM+ZJM
   IF(J-2)1570,1570,1540
1540 ZJMM=ABS(Z(I,J-2))
   IF(ZJMM-BIG)1560,1570,1570
1560 WGT=WGT+CAV
   ZSUM=ZSUM-CAV*(ZJMM-2.*ZJM)
1570 IF(NY-J)1700,1700,1580
1580 ZJP=ABS(Z(I,J+1))
   IF(ZJP-BIG)1600,1700,1700
1600 WGT=WGT+1.
   ZSUM=ZSUM+ZJP
   IF(JM)1620,1620,1610
1610 WGT=WGT+4.*CAV
   ZSUM=ZSUM+2.*CAV*(ZJM+ZJP)
1620 CONTINUE
C
   DZ=ZSUM/WGT-Z000
   DZRMS=DZRMS+DZ*DZ
   DZMAX=AMAX1(ABS(DZ),DZMAX)
   Z(I,J)=Z000+DZ*RELAX
2000 CONTINUE
C
   DZRMS=SQRT(DZRMS/NPG)
   RTRMS=SRMS/DZRMS
   DZMAXF=DZMAX/ZRANGE
C
   WRITE(6,2050) ITER,RELAX,RTRMS,DZMAXF
2050 FORMAT(I5,4H W=F9.6, 7H ROOT=F9.6, 15H DZMAX/ZRANGE= F9.7 )
   IF(ITER-20*(ITER/20))2100,2060,2100
2060 WC=RTRMS+1.
   IF(RELAX-1.-RTRMS)2065,2080,2080
2065 IF(RTRMS-.999) 2070,2100,2100
2070 TPY=(RTRMS+RELAX-1.)/RELAX
   RTJSQ=TPY*TPY/RTRMS
   DEN=1.+SQRT(1.-RTJSQ)
   WC=2./DEN
2080 CONTINUE
   RELAX=WC-.25*(2.-WC)
   IF(DZMAXF/(1.-RTRMS)-EPS)2120,2120,2100
2100 CONTINUE
2120 CONTINUE
C
   DO 2500 I=1,NX
   DO 2500 J=1,NY
   IF(Z(I,J)-BIG)2400,2500,2500
2400 Z(I,J)=ABS(Z(I,J))--ZBASE-10.*ZRANGE
2500 CONTINUE
RETURN
END