NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.
Radar Altimeter Waveform
Modeled Parameter Recovery

G. S. Hayne

August 1981

NASA
National Aeronautics and Space Administration
Wallops Flight Center
Wallops Island, Virginia 23337
AC-804 824-3411
Radar Altimeter Waveform
Modeled Parameter Recovery

G. S. Hayne

NASA Wallops Flight Center
Wallops Island, Virginia 23337
INTRODUCTION

Recent satellite-borne radar altimeters such as GEOS-3 and SEASAT-1 have included waveform sampling gates providing point samples of the transmitted radar pulse after it has been scattered from the ocean's surface. There were 16 such samplers in GEOS-3 and 63 on SEASAT-1, and Table I lists the time locations of the 63 SEASAT-1 samplers relative to the middle one of the set. The waveform sampler data can be averaged over some specified number of individual radar returns, and the resulting set of average values can be best-fitted in some sense by varying the parameters in a model or theoretical mean return waveform. This report describes the waveform model used, and the details of fitting this model to the waveform data to obtain estimates of the parameters characterizing the modeled waveform.

The particular model radar return waveform used at Wallops Flight Center for SEASAT data analysis is characterized by six parameters: 1) amplitude; 2) time-origin, or trackpoint; 3) ocean surface rms roughness; 4) noise baseline; 5) ocean surface skewness; and 6) attitude angle, or off-nadir angle. Figure 1 sketches the model waveform and indicates qualitatively how the model shape changes as these parameters vary. These will be discussed later, but we can briefly state here the importance of these estimated parameters.

The time-origin parameter is the location of the actual mean return waveform relative to the altimeter's altitude-tracker-positioned sampling gate set, and the time-origin is thus directly interpretable as an altitude correction to be applied to the real-time altitude output. The ocean surface rms roughness provides a revised estimate of the significant waveheight (SWH). The ocean surface skewness parameter provides additional information about the surface elevation probability density function and possibly also about the ocean wave spectrum. The amplitude parameter may be used to revise the altimeter-estimated surface backscattering cross-section $\sigma^0$, and the attitude angle also leads to a correction to $\sigma^0$. The noise baseline parameter is of relatively little direct interest but must be included as one of the fitted waveform parameters because the sampling gates measure radar signal plus noise.
# TABLE I. TIME LOCATION AND INDEXING FOR THE 63 SEASAT WAVEFORM SAMPLERS

<table>
<thead>
<tr>
<th>Index</th>
<th>Time, ns</th>
<th>Previous SEASAT #</th>
<th>Previous Report</th>
<th>Index</th>
<th>Time, ns</th>
<th>Previous SEASAT #</th>
<th>Previous Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-92.1875</td>
<td>-30</td>
<td>1</td>
<td>35</td>
<td>4.6875</td>
<td>+ 2</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>-89.0625</td>
<td>-29</td>
<td>2</td>
<td>36</td>
<td>7.8125</td>
<td>+ 3</td>
<td>33</td>
</tr>
<tr>
<td>3</td>
<td>-85.9375</td>
<td>-28</td>
<td>3</td>
<td>37</td>
<td>10.9375</td>
<td>+ 4</td>
<td>34</td>
</tr>
<tr>
<td>4</td>
<td>-82.8125</td>
<td>-27</td>
<td>4</td>
<td>38</td>
<td>14.0625</td>
<td>+ 5</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>-79.6875</td>
<td>-26</td>
<td>5</td>
<td>39</td>
<td>17.1875</td>
<td>+ 6</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>-76.5625</td>
<td>-25</td>
<td>6</td>
<td>40</td>
<td>20.3125</td>
<td>+ 7</td>
<td>37</td>
</tr>
<tr>
<td>7</td>
<td>-73.4375</td>
<td>-24</td>
<td>7</td>
<td>41</td>
<td>23.4375</td>
<td>+ 8</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>-70.3125</td>
<td>-23</td>
<td>8</td>
<td>42</td>
<td>26.5625</td>
<td>+ 9</td>
<td>39</td>
</tr>
<tr>
<td>9</td>
<td>-67.1875</td>
<td>-22</td>
<td>9</td>
<td>43</td>
<td>29.6875</td>
<td>+10</td>
<td>40</td>
</tr>
<tr>
<td>10</td>
<td>-64.0625</td>
<td>-21</td>
<td>10</td>
<td>44</td>
<td>32.8125</td>
<td>+11</td>
<td>41</td>
</tr>
<tr>
<td>11</td>
<td>-60.9375</td>
<td>-20</td>
<td>11</td>
<td>45</td>
<td>35.9375</td>
<td>+12</td>
<td>42</td>
</tr>
<tr>
<td>12</td>
<td>-57.8125</td>
<td>-19</td>
<td>12</td>
<td>46</td>
<td>39.0625</td>
<td>+13</td>
<td>43</td>
</tr>
<tr>
<td>13</td>
<td>-54.6875</td>
<td>-18</td>
<td>13</td>
<td>47</td>
<td>42.1875</td>
<td>+14</td>
<td>44</td>
</tr>
<tr>
<td>14</td>
<td>-51.5625</td>
<td>-17</td>
<td>14</td>
<td>48</td>
<td>45.3125</td>
<td>+15</td>
<td>45</td>
</tr>
<tr>
<td>15</td>
<td>-48.4375</td>
<td>-16</td>
<td>15</td>
<td>49</td>
<td>48.4375</td>
<td>+16</td>
<td>46</td>
</tr>
<tr>
<td>16</td>
<td>-45.3125</td>
<td>-15</td>
<td>16</td>
<td>50</td>
<td>51.5625</td>
<td>+17</td>
<td>47</td>
</tr>
<tr>
<td>17</td>
<td>-42.1875</td>
<td>-14</td>
<td>17</td>
<td>51</td>
<td>54.6875</td>
<td>+18</td>
<td>48</td>
</tr>
<tr>
<td>18</td>
<td>-39.0625</td>
<td>-13</td>
<td>18</td>
<td>52</td>
<td>57.8125</td>
<td>+19</td>
<td>49</td>
</tr>
<tr>
<td>19</td>
<td>-35.9375</td>
<td>-12</td>
<td>19</td>
<td>53</td>
<td>60.9375</td>
<td>+20</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>-32.8125</td>
<td>-11</td>
<td>20</td>
<td>54</td>
<td>64.0625</td>
<td>+21</td>
<td>51</td>
</tr>
<tr>
<td>21</td>
<td>-29.6875</td>
<td>-10</td>
<td>21</td>
<td>55</td>
<td>67.1875</td>
<td>+22</td>
<td>52</td>
</tr>
<tr>
<td>22</td>
<td>-26.5625</td>
<td>-9</td>
<td>22</td>
<td>56</td>
<td>70.3125</td>
<td>+23</td>
<td>53</td>
</tr>
<tr>
<td>23</td>
<td>-23.4375</td>
<td>-8</td>
<td>23</td>
<td>57</td>
<td>73.4375</td>
<td>+24</td>
<td>54</td>
</tr>
<tr>
<td>24</td>
<td>-20.3125</td>
<td>-7</td>
<td>24</td>
<td>58</td>
<td>76.5625</td>
<td>+25</td>
<td>55</td>
</tr>
<tr>
<td>25</td>
<td>-17.1875</td>
<td>-6</td>
<td>25</td>
<td>59</td>
<td>79.6875</td>
<td>+26</td>
<td>56</td>
</tr>
<tr>
<td>26</td>
<td>-14.0625</td>
<td>-5</td>
<td>26</td>
<td>60</td>
<td>82.8125</td>
<td>+27</td>
<td>57</td>
</tr>
<tr>
<td>27</td>
<td>-10.9375</td>
<td>-4</td>
<td>27</td>
<td>61</td>
<td>85.9375</td>
<td>+28</td>
<td>58</td>
</tr>
<tr>
<td>28</td>
<td>-7.8125</td>
<td>-3</td>
<td>28</td>
<td>62</td>
<td>89.0625</td>
<td>+29</td>
<td>59</td>
</tr>
<tr>
<td>29</td>
<td>-4.6875</td>
<td>-2</td>
<td>29</td>
<td>63</td>
<td>92.1875</td>
<td>+30</td>
<td>60</td>
</tr>
<tr>
<td>30</td>
<td>-3.1250</td>
<td>-1 ½</td>
<td>30</td>
<td>64</td>
<td>94.6875</td>
<td>+31</td>
<td>61</td>
</tr>
<tr>
<td>31</td>
<td>-1.5625</td>
<td>-1</td>
<td>31</td>
<td>65</td>
<td>97.1875</td>
<td>+32</td>
<td>62</td>
</tr>
<tr>
<td>32</td>
<td>0.0000</td>
<td>0</td>
<td>32</td>
<td>66</td>
<td>100.3125</td>
<td>+33</td>
<td>63</td>
</tr>
<tr>
<td>33</td>
<td>1.5625</td>
<td>+ 1</td>
<td>33</td>
<td>67</td>
<td>101.875</td>
<td>+34</td>
<td>64</td>
</tr>
<tr>
<td>34</td>
<td>3.1250</td>
<td>+ 1 ½</td>
<td>34</td>
<td>68</td>
<td>103.4375</td>
<td>+35</td>
<td>65</td>
</tr>
</tbody>
</table>
The SEASAT fitted waveform is characterized by the following 6 parameters:

- Amplitude,
- Time origin,
- Ocean surface rms elevation $\sigma_s$,
- Baseline,
- Ocean surface skewness, and
- Off-nadir angle $\xi$.

Figure 1. SEASAT model waveform.
This report is intended to be relatively complete and self-contained in describing the waveform model and the parameter-estimating procedure. The work depends of course on a number of other reports and papers which will be referred to in the text, but all necessary mathematics should be contained in the report or in the program source listings in the Appendix sections. The actual computer implementation is relatively simple but it is not simple to describe succinctly the number of judgments and decisions built into the work. I have tried to refer to some of these considerations in this report and this is my partial excuse for a more wordy account than that of a typical technical journal article.

There are several separate categories which would be included in a complete account of our SEASAT data analysis, including the following topics:

1) details of waveform data input to the programs, data calibration and correction, data editing or screening, the waveform averaging period, and so forth,
2) the general procedure used in fitting a several-parameter function to the corrected waveform sampler averages,
3) the details of and justification for the model waveform used as the several-parameter function in the fitting procedure,
4) verification of the model and of the fitting programs, and
5) presentation of specific results and ground truth comparisons for the results of the data analysis.

This report is not a complete SEASAT data study and will not present any of the information in 5) above. Item 1) will be ignored except to note that most of our SEASAT work typically used 10,000 pulse, ten second waveform data averages. Item 4) on verification is only briefly discussed later. This report will concentrate on items 2) and 3) in the following two sections, describing first the general iterative non-linear least-squares fitting procedure and then the model waveform used. At the heart of any waveform data fitting procedure is the theoretical model which is being fitted, and this report places its major emphasis on that topic.

As another way of establishing the intended purpose of this report, Figure 2 shows the various procedures which should be included within the general category of waveform processing in planning the data processing for any future satellite-borne radar altimeter. Figure 2 is a rough sketch based on some current Wallops Flight Center studies for the proposed TOPEX radar altimeter program, but the figure is general and not restricted to any one specific program. Outputs from the general waveform processor of Figure 2 include: 1) new quantities not otherwise available (such as ocean surface skewness); 2) quantities which are replacements for, and are more precise than, radar altimeter real-time estimates (such as SWH); and 3) quantities which are corrections to altimeter real-time estimates (such as altitudes). Within the general waveform processor of Figure 2 is a box labeled Waveform Modeled Parameter Recovery, and this present report is concerned only with this one particular box in the general Figure 2.
There are a couple of points of general philosophy to mention here. First, we have strongly preferred fitting the data from all of the SEASAT waveform samplers in the plateau region of the return pulse. Otherwise one would need to decide what samplers to ignore, and this criterion would probably have to be a function of significant waveheight. Second, the model waveform is the convolution of several terms as described later, and the parameters of interest could be obtained by some type of deconvolution process as done by Lipa and Barrick (ref. 1). A direct (numerical) deconvolution in this type of problem, a problem characterized by an inherently noisy set of discrete observations over a finite time span, has the general reputation of significant numerical difficulties and instabilities. Accordingly, at the onset of this work some years ago, we chose to use what could be characterized as "iterative convolution," in which an assumed functional form is fitted through a non-linear least-squares procedure, and this is the method described in the remainder of this report.
Finally, a brief paragraph on the history of the SEASAT analysis and on the relationship of this report to other work may be useful. There has been an active effort in waveform analysis at Wallops Flight Center since before the GEOS-3 satellite was launched. The waveform analysis used here for GEOS-3 was described briefly in an earlier report of mine (ref. 2), and this algorithm is among several which were compared in a paper by Fedor et al. (ref. 3). Our earlier GEOS-3 work and the present SEASAT-1 work are based on the radar-ocean surface interaction summarized by Brown (ref. 4). In a recent paper (ref. 5), I developed a several-term expansion describing the mean return waveform from an idealized SEASAT-like radar altimeter; the paper's results do not apply directly to the actual SEASAT-1 radar altimeter because the actual SEASAT-1 radar pulse (sampled by a calibration mode) is not the simple form assumed in Reference 5. Consequences of the actual SEASAT-1 point target response and of certain calibration problems are described in a report (ref. 6), and preliminary results of our waveform processing of SEASAT-1 data are in References 7 and 8. Details of the actual SEASAT-1 altimeter design and altimeter hardware are available in References 9 and 10, and some of the standard ground-based data processing for SEASAT-1 data is described by Hancock et al. (ref. 11). A later section of this report discusses numerical convolutions performed through use of the Fast Fourier Transform, and Brigham (ref. 12) is one of many sources of information on this subject. We have already mentioned the somewhat different approach to waveform processing by Lipa and Barrick (ref. 1). The principal goal in our work has been the estimation of sea surface skewness from satellite altimeter data, and the work of Huang and colleagues (refs. 13 and 14) provides a relationship between ocean surface spectrum details and skewness and significant waveheight.

**NON-LINEAR LEAST-SQUARES FUNCTION-FITTING ROUTINE**

**Introductory Discussion**

The problem of fitting a function to input experimental data has been divided into two pieces; this section of the report will describe the fitting procedure itself with little concern for what function is being fitted, while a later section of the report will describe the specific function (the model waveform) to be fitted in the radar altimeter waveform problem.

A least-squares fitting procedure is desired but there is a limited number of types of functions for which the normal equations are directly solvable in an ordinary least-squares procedure. The model waveform function described later in this report produces an intractable set of normal equations, and in this situation the approach is to: a) make a
first guess at the values of the parameters in the fitting function, b) do a first-order Taylor series expansion in parameter space around the first guess values, c) do a least-squares solution for the corrections to the first-guess parameter values, and d) use the revised parameter values as a new first guess, looping back to step b) and repeating until some exit criterion is satisfied. This was used earlier in the GEOS-3 data analysis (ref. 2) and the general approach has been described in a number of different sources such as Reference 15. (References 16 and 17 provide some additional discussion of some interest to this general problem.) The following subsection will present this method in more specific detail and a subsequent subsection then discusses the particular subroutine version implemented here.

General Derivation

Suppose that there is a set of experimental data $Y_i$, $i = 1$ to $N$, with each $Y_i$ from a different value of an independent variable $t_i$. Suppose further that a model function $R$ is to be fitted to these data, where $R$ is described by a set of constants $K$ and a set of $m$ parameters $\theta$ to be fitted,

$$\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}.$$  

We will designate the model function value at $t_i$ by $R(t_i; K, \theta)$. The number of points $N$ must be greater than the number of parameters being fitted, $m$. For SEASAT, $R$ will be the model return waveform, the $Y_i$ will be the $N = 63$ waveform sampler values, and number of parameters fitted will be $m = 6$, but the derivation in this section is intended to be general and not specific to SEASAT alone.

Let $\theta_0$ be the set of initial guesses for the $m$ parameters,

$$\theta_0 = \{\theta_1^0, \theta_2^0, \ldots, \theta_m^0\}.$$  

and expand $R$ in a Taylor series about the point $\theta_0$ in the $m$-dimensional fit parameter space. For notation convenience, define

$$R_{t_0} = R(t_i; K, \theta_0).$$

and the Taylor series is given by


\[ R(t_i; K, \Theta) = R_{i0} + (\Theta_i - \Theta_{i0}) \frac{\partial R_{i0}}{\partial \Theta_i} + (\Theta_2 - \Theta_{20}) \frac{\partial R_{i0}}{\partial \Theta_2} + \ldots + (\Theta_m - \Theta_{m0}) \frac{\partial R_{i0}}{\partial \Theta_m} + \text{higher-order terms.} \quad (4) \]

Define \( D_{ij} \) and \( S_j \) by

\[
D_{ij} = \frac{\partial R_{i0}}{\partial \Theta_j} = \frac{\partial}{\partial \Theta_j} R(t_i; K, \Theta) \quad \text{evaluated at } \Theta_0
\]

and

\[
S_j = \Theta_j - \Theta_{j0}.
\]

Ignore all the higher-order terms, and the above Taylor series becomes

\[ R(t_i; K, \Theta) = R_{i0} + \delta_1 D_{i1} + \delta_2 D_{i2} + \ldots + \delta_m D_{im}. \quad (7) \]

This is a linear equation in \( \delta_1, \delta_2, \ldots, \delta_m \) and can be treated by a least-squares method to solve for the \( \delta_1, \ldots, \delta_m \).

Although most of our SEASAT analysis has been for uniformly weighted data we will include the possibility of nonuniform data weighting in the following. Define a set of weights \( w_i \) for the \((Y_i, t_i)\) experimental data points, and then define \( Q \) as the weighted sum of squares of the residuals between the \( Y_i \) and the \( R(t_i; K, \Theta) \),

\[
Q = \sum_{i=1}^{N} w_i [Y_i - R(t_i; K, \Theta)]^2 = \sum w_i [Y_i - R_{i0} - \delta_1 D_{i1} - \delta_2 D_{i2} - \ldots - \delta_m D_{im}]^2. \quad (8)
\]

A minimum in \( Q \) is found by taking the partial derivatives of \( Q \) with respect to \( \delta_1, \delta_2, \ldots, \delta_m \), and setting each of these to zero. This set of \( m \) partial derivatives is
\[
\frac{\partial}{\partial \delta_1} = -2\Sigma_{i=1}^{N} D_{11}(Y_i-R_{i1}) - \delta_{11} - \delta_{12} D_{12} - \cdots - \delta_{1m} D_{1m} = 0
\]
\[
\frac{\partial}{\partial \delta_2} = -2\Sigma_{i=1}^{N} D_{12}(Y_i-R_{i2}) - \delta_{11} - \delta_{12} D_{12} - \cdots - \delta_{2m} D_{2m} = 0
\]
\[
\vdots \quad \vdots \quad \vdots
\]
\[
\frac{\partial}{\partial \delta_m} = -2\Sigma_{i=1}^{N} D_{im}(Y_i-R_{im}) - \delta_{11} - \delta_{12} D_{12} - \cdots - \delta_{mm} D_{mm} = 0
\]

All the indicated sums are from \( i = 1 \) to \( N \). These \( m \) equations can then be written as a matrix equation,

\[
B = A \Delta
\]

in which \( B \) and \( \Delta \) are \( m \) by 1 column matrices and \( A \) is a symmetric \( m \) by \( m \) matrix. These matrices are

\[
\Delta = \begin{bmatrix}
\delta_1 \\
\delta_2 \\
\vdots \\
\delta_m
\end{bmatrix}, \quad
B = \begin{bmatrix}
\Sigma_{i=1}^{N} D_{11}(Y_i-R_{i1}) \\
\Sigma_{i=1}^{N} D_{12}(Y_i-R_{i2}) \\
\vdots \\
\Sigma_{i=1}^{N} D_{im}(Y_i-R_{im})
\end{bmatrix}
\]

and

\[
A = \begin{bmatrix}
\Sigma_{i=1}^{N} D_{11}^2 & \Sigma_{i=1}^{N} D_{11} D_{12} & \cdots & \Sigma_{i=1}^{N} D_{11} D_{1m} \\
\Sigma_{i=1}^{N} D_{11} D_{12} & \Sigma_{i=1}^{N} D_{12}^2 & \cdots & \Sigma_{i=1}^{N} D_{12} D_{1m} \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{i=1}^{N} D_{11} D_{im} & \Sigma_{i=1}^{N} D_{12} D_{im} & \cdots & \Sigma_{i=1}^{N} D_{im}^2
\end{bmatrix}
\]
Then if matrix $A$ has an inverse $A^{-1}$, one solves for $A^{-1}$ and the solution for $\Delta$ is

$$\Delta = A^{-1} B.$$  \hspace{1cm} (13)

Then from the definition of the $\delta_j$ in equation (6), the new parameter estimates $\theta_j$ can be obtained from $\delta_j$ and $\theta_j^0$ by

$$\theta_j = \theta_j^0 + \delta_j, \quad j = 1 \text{ to } m.$$ \hspace{1cm} (14)

These new $\theta_j$ parameters are not necessarily equal to the "true" or final parameter set $\theta$ because of the higher order terms having been ignored in the Taylor series leading to equation (7). The next step is to use the $\theta_j$ of equation (14) as a revised guess, a revised $\theta_0$, and to repeat the entire process just described.

This iterative process is repeated for some preset maximum number of iterations or until some type of convergence criterion is satisfied. Additional error exit provisions must be made for cases in which the matrix $A$ is nearly singular or in which $Q$, the weighted sum of residuals squared, increases over its value in the preceding iteration. The usual convergence criterion is that the fractional change in $Q$ from one iteration to the next should be less than some preset limit. An additional iteration exit should be provided for the absolute value of $Q$ falling below some lower limit even if the $Q$ fractional change criterion has not been met. The particular lower limit specified is based on reasonable judgment in the particular application of the general fitting procedure, and also can be influenced by computer precision limits due to finite computer word lengths.

Additional Discussion of the Wallops Version of a Non-linear Least-Squares Subroutine

The subroutine SWHFIT in Appendix B carries out the general iterative least-squares procedure described above. These several paragraphs will discuss some of the features and considerations in this subroutine.

The function being fitted is provided by a subroutine FILLV which is described further in this report’s section on the model waveform. The various partial derivatives with respect to the fitted parameters, the $D_{ij}$ of equation (5), are supplied by a subroutine FILLD. Notice that by keeping FILLV and FILLD outside the matrix-fitting loop within SWHFIT, we have the ability to modify the model waveform function at will without disturbing the fitting procedure; also the FILLV function is evaluated for all $N$ (usually 63) values at one point in SWHFIT so that the convolution-by-FFT can be used as described later. For our recent work, the derivatives $D_{ij}$ in FILLD are computed by making a step-
change in each of the estimated fit parameters and recalculating the fit function in FILLV, and the array SiPRM in SWHFIT contains the step-sizes currently used.

SWHFIT has been a general research tool for a variety of waveform work and we have found it useful to be able to change the order and number of parameters being fitted without having to change the program source code; an array JORDR allows this as described further by comments in the Appendix B source listing.

It has sometimes been useful to limit the size of the corrections $\Delta$ allowed in equation (14) for the first several iterations to prevent the iterative process from jumping completely outside the region of convergence in the event of a particularly poor first guess for the parameters $\theta_0$. Subroutine SWHFIT allows $1/5$ the correction for the first iteration, $2/5$ for the next, and so forth until the fifth and subsequent iterations when the full correction is made.

Another feature of SWHFIT is its provision for parameter constraints as suggested to us by Dr. William Wells of Washington Analytical Services Center, Inc. (of EG&G, Inc.). These constraints are added to the diagonal elements in the matrix A, with each constraint being approximately the inverse of the appropriate fit parameter's input variance estimate in cases in which some a priori information is available on the variation expected in the parameters to be fitted.

A general simulation program, to be described only briefly later in this report, has been used as an "experimental" means of setting the constraints used for the SEASAT analysis. The ad hoc recipe for limiting $\Delta$ in the first several iterations was the result of earlier data analysis and simulation. The exit criterion, based on the fractional change in the sum of residuals squared, has the value 0.001 in the SEASAT work, and the maximum number of iterations allowed is 30. SWHFIT contains features which are specific to the FILLV function being a radar altimeter mean return waveform. One such feature is a test which bars the possibility of a negative risetime or a negative significant wave-height. Notice also in SWHFIT that when the antenna pointing angle is being fitted, there are limits on the range of values allowed. Also in the case that the sum of residuals squared increases after several iterations, the parameter values returned by SWHFIT will be those for which the sum of residuals squared had its minimum.

For the general iterative non-linear least-squares procedure described, there is no fundamental guarantee that the sum of residuals squared which is reached in a fit is the global minimum and not just a local minimum. We can only point out that for a number of cases tried in the simulation and verification procedure, both with and without noise added to the simulated waveform samples, the correct parameters were recovered in the fitting procedure. Furthermore, the results seemed insensitive to a range of first-guess values of the parameters.
MODEL WAVEFORM FOR RADAR ALTIMETER DATA

In this section I will briefly review the fundamental assumptions underlying the model waveform used for fitting to the radar altimeter waveform sampler data. The three separate terms to be convolved in the model are then separately discussed. One of these terms has, for SEASAT, been impossible to represent by a simple analytical expression with the consequence that at least one numerical convolution step is required in generating a model waveform for any particular set of waveform parameters. To reduce the computation time required, this numerical convolution step can be performed through Fast Fourier Transform (FFT) techniques, and this is further described in a later subsection of this section on the model waveform.

Convolutional Model for Rough Surface Radar Returns

In 1958 Moore and Williams (ref. 18) demonstrated that the average radar power return from a rough surface and for near-normal incidence could be expressed as a convolution of the "transmitted pulse shape" and a term which included the effects of antenna pattern and off-nadir pointing angle, surface properties, and distance. More recently, Brown (ref. 4) discussed this convolutional model and, for the assumptions common to satellite radar altimetry systems, produced a simplified closed-form expression for the average rough surface impulse response function.

My own work is based on the material in Brown's paper (ref. 4) as a starting point and I will assume that the average waveform can be viewed in an "elapsed-time" domain in which the actual hardware waveform sampling gates can be labelled in successive time units. For instance, the 63 SEASAT waveform samplers are spaced over a time span of almost 185 nanoseconds. I will assume that I can use the concept of the "effective" pulse and ignore the actual details of any pulse compression scheme actually used. SEASAT for instance had an effective pulse width of some several nanoseconds, even though the actual signal transmitted from the altimeter was about 3.2 microseconds in duration.

Since the waveform model is viewed in the altimeter's elapsed-time domain, any surface elevations must be converted to elapsed times (or ranging times) through the radar altimeter's two-day ranging time-to-distance factor of (-c/2) where c is the speed of light. The negative sign is included because an increase in surface elevation (the elevation is positive for directions upward out of the sea surface) corresponds to a decrease in the ranging time.

The following description of fundamental assumptions in the convolutional model for rough surface scatter is quoted directly from Brown's paper (ref. 4):
1. The scattering surface may be considered to comprise a sufficiently large number of random independent scattering elements.

2. The surface height statistics are assumed to be constant over the total area illuminated by the radar during construction of the mean return.

3. The scattering is a scalar process with no polarization effects and is frequency independent.

4. The variation of the scattering process with angle of incidence (relative to the normal to the mean surface) is only dependent upon the backscattering cross section per unit scattering area, $o^o$, and the antenna pattern.

5. The total Doppler frequency spread ($4V_r/\lambda$) due to a radial velocity $V_r$ between the radar and any scattering element on the illuminated surface is small relative to the frequency spread of the envelope of the transmitted pulse ($2/T$, where $T$ is the width of the transmitted pulse).

Over the ocean surface, all of the above assumptions are generally satisfied. However, we must always be careful in selecting the averaging time to insure that surface statistical homogeneity is satisfied. For land scatter, the situation is somewhat different, and some of the above assumptions may be violated.

Under these various assumptions above, the model mean return waveform $W(t)$ is given by the convolution of three terms,
Figure 3. General radar altimeter mean return waveform as convolution of 3 terms.

We have assumed in our SEASAT work to date that the tracker jitter is small enough in width compared to the \( q_s(t) \) that we could ignore any part of \( T(t) \) except for the possible track-point shift, \( t_0 \). In this case \( T(t) \) takes the form of a shifted delta function,

\[
T(t) = \delta(t-t_0) .
\]  

We may find on detailed investigation of the tracker jitter about the true waveform position, that the delta function in \( T(t) \) above may need to be replaced by some other form such as a skewed Gaussian whose width and skewness are both functions of the significant waveheight and of the AGC level, but this investigation has not yet been finished.

The final model waveform, from equation (16) above, includes six parameters which are varied in least-squares fitting the input waveform sampler data. These six parameters are listed in the table below. Figure 1 in the Introduction to this report showed qualitatively how the model waveform changes as a result of changes in these six parameters.
In the special case that the $s_r(t)$ and $q_s(t)$ in equation (16) can each be represented by a skewed Gaussian form and that the $T(t)$ has the simple shifted-delta-function form of equation (17), I have described in ref. 5 a several-term analytical expression for $W(t)$; an earlier report of mine (ref. 6) was also based largely on this analytical result. We have more recently decided that, given the SEASAT $s_r(t)$, there is no way to avoid a numerical convolution whenever equation (16) is to be evaluated for any set of values for the parameters in Table II, and we use FFT techniques to speed the numerical convolutions in our computer.

**TABLE II. PARAMETERS VARIED IN FITTING MODEL WAVEFORM TO SAMPLER DATA.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Name or Symbol</th>
<th>Description</th>
<th>Term Within Eq. (16) in Which Parameter Appears</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Amplitude</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>$t_0$</td>
<td>Time-Origin (or track-point)</td>
<td>$T(t)$</td>
<td></td>
</tr>
<tr>
<td>$q_s$</td>
<td>Sea Surface rms Elevation</td>
<td>$q_s(t)$</td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td>Average Noise Baseline</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>$\lambda_s$</td>
<td>Sea Surface Skewness</td>
<td>$q_s(t)$</td>
<td></td>
</tr>
<tr>
<td>$\xi$</td>
<td>Attitude Angle (or off-nadir angle)</td>
<td>$P_{FS}(t)$</td>
<td></td>
</tr>
</tbody>
</table>

"Flat"-Surface Average Impulse Response Function

This function describes the average power return for a delta-pulse scattered off a "flat" ocean; the quotation marks on flat are to remind us that the sea must be rough at the centimeter scale for the incoherent, rough-surface scatter theory to apply, but that any surface roughness effects at tens of centimeters or greater (i.e., at the oceanographer's level of interest) are described elsewhere in the $q_s(t)$ term. From (ref. 4), the flat-surface average impulse response function is given by

$$P_{FS}(t) = A_0 \exp(-\delta t) I_0(t^4) U(t), \quad (18)$$

in which $\delta$ is a constant (not the $\delta$-function),

$$\delta = (4/\gamma) \left( c/h \right) \cos(2\xi) \quad (19)$$
and

$$8 = \left(\frac{4}{Y} \frac{c}{h}\right)^\frac{1}{2} \sin(2\xi), \quad (20)$$

with $U(t)$ being the unit step function, and $I_0(t^2\beta)$ being a modified Bessel function. In equations (19) and (20) $c$ is the speed of light, $h$ is the spacecraft altitude, $\xi$ is the absolute off-nadir pointing angle (or attitude angle), and $Y$ is an antenna beamwidth parameter defined as in Brown's equation (ref. 4) by assuming a Gaussian approximation to the antenna gain for an angle $\theta$ off the axis,

$$G(\theta) = G_0 \exp\left[-\left(\frac{2}{Y}\sin^2\theta\right)\right]. \quad (21)$$

If $\theta_w$ is the usual antenna beamwidth, the angular full-width at 1/2-power points, then

$$\frac{4}{Y} = \frac{\sin 4}{\sin^2(\theta_w/2)}. \quad (22)$$

Notice that the time in equation (18) is zero at the instant of the first non-zero power return to the altimeter.

The amplitude term $A_0$ in equation (18) above contains several other constants:

$$A_0 = \frac{G_0 \lambda_r^2 c \sigma^o(0)}{4(4\pi)^2 L_p h^3} \exp\left(-\frac{4}{Y} \sin^2\xi\right), \quad (23)$$

where

- $\lambda_r$ is the radar wavelength,
- $\sigma^o(0)$ is the ocean surface backscattering cross section at normal incidence,
- $G_0$ is the radar antenna boresight gain, and
- $L_p$ is the two-way propagation loss over and above the free-space loss.

In the SEASAT altimeter, the radar return signals are normalized by the automatic gain control (AGC) circuit and we ignore all the individual terms within $A_0$ in equation (23). Instead $A_0$ is treated as a simple amplitude scaling constant which is set equal to one for convenience since the $A$ constant in equation (18) will provide the necessary amplitude variation. For future work, note that the combination of fitted $A$ and $\xi$ plus the altimeter's AGC data would lead to a revised $\sigma^o$ estimate.

Let's emphasize here that the flat surface impulse response function given by equation (18) above is an average-power quantity over many pulses and has no meaning on an individual radar return waveform. Indeed, as Moore and Williams emphasize (ref. 18), the range from a signal level exceeded 5% of the time to a level exceeded 95% of the time is
about 18 dB for the underlying Rayleigh distribution in amplitudes. Any individual pulse is likely to look very different from the mean described in equation (18).

A sample subroutine GTFSR for evaluating the \( P_{FS}(t) \) of equation (18) is provided in Appendix D. In evaluating the \( I_0 \) in (18), polynomial approximations from Abramowitz and Stegun (ref. 19) are used. In the following \( t = z/3.75 \).

If \(-3.75 \leq z \leq 3.75\), find \( I_0(z) \) from

\[
I_0(z) = 1 + 3.5156229t^2 + 3.0899424t^4 + 1.2067492t^6 + .2659732t^8 + .0360768t^{10} + .0045813t^{12} .
\]  

(24)

If \( 3.75 < z < \infty \), find \( I_0(z) \) from

\[
[z^{1/2}\exp(-z)] I_0(z) = .39894228 + .01328592t^{-1} + .00225319t^{-2} \
- .00157565t^{-3} + .00916281t^{-4} - .02057706t^{-5} \
+ .02635537t^{-6} - .01647633t^{-7} + .00392377t^{-8} .
\]  

(25)

For digital computer use, equations (24) and (25) have to be rewritten in nested form to avoid round-off errors from the higher powers in \( t \), and subroutine GTFSR in Appendix D contains the rewritten \( I_0 \) polynomial approximations.

Finally, the possibility of a "negative angle" has been built into the flat-sea impulse response function \( P_{FS}(t) \) in subroutine GTFSR of Appendix D. The angle \( \xi \) is supposed to be a magnitude only and thus a negative angle has no possible physical meaning. Notice that the fastest plateau decay possible in equation (18) is when \( \xi = 0 \) in which case \( \beta = 0, \delta = (4/\gamma)(c/h) \), and the \( P_{FS}(t) \) becomes

\[
P_{FS}(t) = A_0 U(t) \exp(-\delta t) .
\]

(26)

There are, however, experimental situations in which an even more rapid plateau decay is needed. There are at least two such situations: a) the actual \( \xi \) is very nearly zero and the noise character of the waveform sampler data will lead to plateau decay rates varying both plus and minus around the decay rate in equation (26) above, or b) erroneous waveform sampler gain calibration data are used so that the decay of the plateau is apparently more rapid than allowed in equation (26).

For this reason, a branch is included in subroutine GTSEA which evaluates equation (18) normally for \( \xi \geq 0 \) but which evaluates equation (26) with a changing "effective" \( \delta > 0 \) if \( \xi < 0 \). The magnitude of the change in \( \delta \) is related to the (negative) magnitude of \( \xi \), and the scaling constant on the \( \delta \)-change is chosen so that the average of a set of
simulated ξ = 0 cases is zero, with individual ξ estimates in this simulation work varying positive and negative.

A negative ξ result from fitting real waveform sampler data is a flag or a warning that there could be waveform modeling errors (or calibration errors); if the ξ estimates are only sometimes negative and are consistent with the expected standard deviation of the ξ-estimation process, then the negative signs pose no problems and simply make the averages of sets of ξ-estimates come out closer to zero. The expected standard deviation of the ξ-estimates can be gotten from a simulation program as sketched later in this report.

Radar-Observed Sea Surface Elevation Probability Density Function

The radar-observed surface elevation probability density function q_s(t) describes the vertical extent and distribution of the ocean surface radar scattering process, and the elevation-to-time transformation is made through the (-c/2) conversion factor. The phrase "radar-observed p.d.f." is used as a reminder that the effective p.d.f. may possibly differ from the geometric surface elevation p.d.f. By geometric surface, I mean the sea surface as seen by optical instruments. Our various waveform analysis work (refs. 6, 7, and 8) has assumed the one specific p.d.f. form given in the next paragraph, but the reader should be aware that there are some alternative p.d.f. forms being proposed. Jackson (ref. 20) proposed one modified s_r(t) form based on theoretical work assuming infinitely long-crested waves, while Lipa and Barrick (ref. 1) have proposed a somewhat different form which includes a "height-slope^2 cross-skewness" parameter. My approach has been to proceed with one particular often-assumed surface elevation p.d.f. with the possibility of later substituting a different form at such time as the other research on this problem seems sufficiently convincing. Subroutine GTSEA in Appendix D contains the p.d.f. described in the next paragraphs, and the possible future use of different functional forms can be accomplished simply by suitable rewriting of this one subroutine.

The radar-observed surface elevation p.d.f. q_s(t) used here is assumed to be the skewed Gaussian form given in the time domain by

\[
q_s(t) = \frac{1}{\sqrt{2\pi} \sigma_s} \left[ 1 + \frac{\lambda_s}{6} H_3(t/\sigma_s) \right] \exp\left[ -\frac{1}{2} (t/\sigma_s)^2 \right],
\]

where σ_0 is the surface rms waveheight, λ_s is the surface skewness, and H_3 is a Hermite polynomial. The Hermite polynomials for general argument z needed in this paper's discussion are

\[
H_3(z) = z^3 - 3z.
\]
\[ H_4(z) = z^4 - 6z^2 + 3, \]  
\[ H_6(z) = z^6 - 15z^4 + 45z^2 - 15. \]  

All satellite altimeter measurements of sea surface roughness have assumed that the significant waveheight (SWH) is exactly four times the rms surface elevation, and since \( \sigma_s \) is in time units, the SWH in distance units is

\[ \text{SWH} = 4(c/2)\sigma_s. \]  

The \( \lambda_s \) in equation (27) is the surface skewness. Skewness is a non-dimensional quantity, the ratio of a third central moment to the 3/2 power of the second central moment, so no specific use of the \((c/2)\) distance-time conversation factor is needed. It is important to realize however that the sign of the surface skewness in the spatial domain is opposite to the sign in the (ranging) time domain.

The skewed Gaussian form in equation (27) above was used by Pierson and Mehr (ref. 21) in Skylab discussion and by Walsh (ref. 22) in GEOS-3 data as well as in our SEASAT work (refs. 6, 7, and 8); it is a low order case of a general probability function by Longuet-Higgins (ref. 23) for a random variable that is weakly nonlinear. This form is the result of taking the first two terms only in a general Gram-Charlier series (ref. 24).

Recent work by Huang and Long (ref. 13) proposed that equation (27) above is not adequate for the surface elevation density and that additional terms must be included in the square of the skewness and in the kurtosis, \( \kappa_s \), and that the proper surface elevation p.d.f. to use is

\[ q_s(t) = \left[ 1 + \frac{\lambda_s^2}{6} H_3 \left( \frac{t}{\sigma_s} \right) + \frac{\kappa_s}{24} H_4 \left( \frac{t}{\sigma_s} \right) + \frac{\lambda_s^2}{72} H_6 \left( \frac{t}{\sigma_s} \right) \right] \times \frac{1}{\sqrt{2\pi} \sigma_s} \exp\left[ - \frac{1}{2} \left( \frac{t}{\sigma_s} \right)^2 \right]. \]  

We chose to ignore the \( \lambda_s^2 \) and \( \kappa_s \) terms, feeling that these were probably less important in the ocean than in the laboratory. Also the magnitude was likely to be less than the noise in our case. (Figure 6 of Reference 5 shows the effect of the \( \lambda_s^2 \) term for example.) While our work to date has been based on the simpler \( q_s(t) \) of equation (27), it is a simple matter to change subroutine GTSEA to the equation (32) form if desired.
There are two reasons for strong interest in the sea surface skewness estimates $\lambda_s$. First, this quantity enters some of the theories concerning possible differences between "radar-observed" and "actual" surface elevation p.d.f., the so-called electromagnetic bias in radar altimeters, as in Jackson's paper (ref. 20) for instance. Second, the Huang and Long work (ref. 13) provides the connection between $\sigma_s$ and $\lambda_s$ and two other parameters $\delta$ and $\lambda_0$. The significant slope $\delta$ and the dominant wavelength $\lambda_0$ are the two important parameters in a new unified two-parameter ocean wave spectral model of Huang et al. (ref. 14), and thus there is a possibility of obtaining surface spectrum information from radar altimeter measurements.

Radar Altimeter Point-Target Response Function

The radar altimeter system point-target response function $s_r(t)$ is the system's effective transmitted pulse as sampled by the waveform sampler set. This point target response is primarily the effective transmitted pulse shape but also includes effects of the receiver bandwidths. The word "effective" above is the verbal rug under which is swept the actual details of pulse compression and implementation.

The only information on the SEASAT $s_r(t)$ is provided by the altimeter Calibration Mode I in which a portion of the transmitted pulse is fed back through the receiver to the waveform sampler set. The SEASAT altitude tracking and SWH estimation in the altimeter hardware were based on an assumed pure Gaussian $s_r(t)$ with a full-width at 1/2-power points of 3.125 nanoseconds. MacArthur (ref. 9) describes a ground-based data processing correction to SWH to allow for the \( (\sin x)^2 \) shape which the SEASAT $s_r(t)$ ideally had (as a result of the particular pulse compression technique implemented). I have described in an earlier report (ref. 6) the actual $s_r(t)$ which is not symmetric nor representable by any simple analytical expression. Figure 4 from (ref. 6) sketches this SEASAT $s_r(t)$, based on SEASAT Calibration Mode I Step 9 data, and this figure provides an estimated value every 1.5625 ns. This is the spacing between the five sampling gates at the center of the waveform sampling gate set, while the spacing between all of the rest of the sampling gates is 3.125 ns. We have assumed zero values at every half-gate position between the 3.125 ns-separated samplers, based on the assumed nulls in the ideal \( (\sin x)^2 \) response.

In SEASAT the fundamental 1.5625 ns spacing (of the sampled values on all the terms to be numerically convolved) is fixed by the limits on our knowledge of $s_r(t)$ and on its densest sampling being at the 1.5625 ns separation at the center of the sampling gate set. In subroutine FILLV of Appendix C, the subroutine which evaluates the model waveform, the $s_r(t)$ information is carried in array SYS(514) and the two integers NSYS and NSCTR. NSYS is the point after which all remaining SYS values are zero (clearly NSYS<514), and NSCTR
is the index of the peak or center at the sampled $s_p(t)$. NSCTR is used in a time shifting step which in effect sets the peak element within SYS at the time zero.

If one can assume that the $s_p(t)$ has a skewed Gaussian form like equations (27) or (32) of the preceding subsection, then my Reference 5 provides a several-term analytic expression for the model waveform and subroutine FILLV of Appendix C could be replaced by a coded version of that expression. This may be useful for simulations of future altimeters, but I expect that any actual altimeter will have a $s_p(t)$ at least as complicated as the SEASAT case in Figure 4, and that the full convolution procedure of this present report will again be necessary.
Numerical Convolution by the FFT Method

The preceding discussion in this section on model waveforms has emphasized the need to carry out a numerical convolution because of the actual SEASAT $s_s(t)$ form. Moreover, my discussion of $q_s(t)$ has at least implied the desirability of being able to change the functional form used for this surface elevation p.d.f. This in turn means at least two numerical convolutions for a set of model waveform values. Then the step-estimation of derivatives within the fitting process will require four more model waveform sets to be generated if six parameters are fitted (in general, six parameters being fitted would require six more model waveform sets to be generated, but in this problem we know already the derivatives of equation (16) with respect to the parameters $A$ and $b$). Finally the general fitting program will require from five to ten iterations through the waveform-plus-derivative loop. In brief, there are many convolution steps required to estimate waveform parameters for one set of averaged waveform sampler data, and ways to speed this processing must be found.

One way to speed the convolution is to take advantage of the convolution theorem in Fourier transform theory. This theorem asserts that if one has two time functions $p(t)$ and $q(t)$ whose Fourier transforms $P(f)$ and $Q(f)$ exist, the Fourier transform of the convolution $p(t) * q(t)$ is the product $P(f)Q(f)$. Consequently, the convolution $p(t) * q(t)$ can be performed by: a) taking the Fourier transforms of $p(t)$ and $q(t)$ to obtain $P(f)$ and $Q(f)$; b) forming the (complex) product $[P(f)Q(f)]$; and c) taking the inverse Fourier transform of $[P(f)Q(f)]$ to obtain the desired answer. This procedure also can be used for two sampled time functions using the discrete Fourier transform (DFT) and the inverse DFT, providing that the two functions can be set to zero outside a region of interest. The computational efficiencies of the Fast Fourier Transform (FFT) result in the above-sketched convolution through transforms technique being faster in many cases than the simpler straightforward numerical convolution, and we have found by trying both methods that there is a speed advantage to convolution by transforms within the model waveform computation.

There is a large body of literature concerning Fourier transforms, DFTs, FFTs, convolution, and so forth; the textbook by Brigham (ref. 12) is a source which is quite readable and accessible, and upon which the discussion in these paragraphs is based. The particular FFT and inverse FFT programs used here for model waveform computations were the subroutines FFA and FFS and their associated subroutines from a published package (ref. 25) of digital signal processing programs. FFA is a radix 8-4-2 FFT routine for a real data sequence which performs as many base 8 iterations as possible and then performs one base 4 or base 2 iteration if necessary, and FFS is the radix 8-4-2 inverse counterpart of FFA. The inclusion of the radix 8 steps in FFA and FFS leads to longer Fortran source code but shorter computation times than for other simpler FFT routines, and additional timing discussion is provided by Reference 25.
The FFT-convolution will require that the total array length $N_T$, used in each term to be convolved, be the same and in general there is the FFT requirement that $N_T$ be some power of 2, $N_T = 2^M$. The smaller the value of $M$, the faster the computation, but $M$ must be large enough to avoid overlap effects as discussed in ref. 12. Suppose again that two (discrete) sequences $p(t)$ and $q(t)$ are to be convolved and that $p(t)$ has $N_p$ non-zero sampled values while $q(t)$ has $N_q$ such values. There will be $(N_T-N_p)$ and $(N_T-N_q)$ successive zero values in the arrays for $p(t)$ and $q(t)$ respectively. Then the requirement that $N_T \geq N_p + N_q - 1$ will avoid the overlap problem. If there are three sequences $p(t)$, $q(t)$, and $s(t)$ with numbers $N_p$, $N_q$ and $N_s$, the requirement on $N_T$ becomes $N_T \geq N_p + N_q + N_s - 2$. Obviously we are assuming the same sample time intervals $\Delta$ for $p(t)$, $q(t)$, and $s(t)$ and for the final answer.

In the specific SEASAT waveform problem, the sampling time $\Delta_S$ is 1.5625 ns, the spacing of the gates in the center of the set of waveform samplers. The total width spanned by the sampling gates is around $120 \Delta_S$ and the flat sea response function $PFS(t)$ must be allowed at least this width of non-zero values so $N_p = 120$. To allow for up to 20 m SWH implies that upwards of 200 $\Delta_S$ is needed to represent $q_s(t)$ out to the $\pm 4\Delta_S$ level, so $N_q = 200$. Then the system point-target response function has a width upwards of 100 $\Delta_S$, so $N_p = 100$. This leads to a total $N_T = 120 + 200 + 100 = 400$, and the next highest $2^M$ is at $M = 9$ for $N_T = 512$. The actual widths used vary somewhat from this illustrative example, but $N_T = 512$ is the total array length used.

The model waveform evaluation subroutine FILLV in Appendix C uses 512-point arrays for the $PFS(t)$, $q_s(t)$ and $s_p(t)$ and 512-point FFT evaluation (the actual array dimensions are 514 because of the details of the FFA and FFS subroutines used). The time-shift term $T(t)$ of equation (17) is not written as a separate term in the convolution, but is accomplished through the Fourier transform time-shift theorem (ref. 12); a shift in the time domain leads to a phase shift in the frequency domain, so that for example if $p(t)$ and $P(f)$ are Fourier transform pairs, then the Fourier transform of $p(t-to)$ is $P(f) \exp(-j 2\pi ft_o)$. A similar property applies to the discrete Fourier transform, and FILLV takes advantage of this and includes the phase shift in the complex product in the transform domain before doing the inverse FFT to bring the model waveform back to the time domain. Details of this are obvious in the FILLV source code in Appendix C.

Finally, the convolution via FFT method described above was tried initially as an experiment, to see if the computation speed was improved over the straightforward convolution of two time functions; there was an improvement of nearly an order of magnitude, so the FFT method was kept in our waveform fitting. Adding one more convolution for a total of three terms convolved then became simple and cost less than twice the time for two terms only to be convolved via FFTs, and this allowed us to separate the problem so that $PFS(t)$, $q_s(t)$, and $s_p(t)$ were entered by different subroutines. This modular structure will easily permit future possible changes in the functional form used for $q_s(t)$. Also as
we note later, most of the time in our waveform fitting computations is taken by the
computer doing FFTs, and this makes attractive the process of using one of the available
array processors in any future computer system which spends an appreciable part of its
time doing waveform fitting for modeled parameter recovery.

ADDITIONAL PROCESSING CONSIDERATIONS

Use of Calibration Data

If the set of SEASAT waveform sampler measurements is \( Y_i, 1 \leq i \leq 63 \), it has been
assumed that there is a set of gain corrections \( g_i \) and offset (or shift) corrections \( s_i \)
such that a corrected set of sampler values \( Y_i' \) is obtained from

\[
Y_i' = g_i (y_i - s_i), \quad 1 \leq i \leq 63
\]  

(33)

The offsets \( s_i \) were to be obtained from the altimeter's standby mode, under the assumption
that they were generated somewhere in the sampler-to-telemetry system interface. The
gains \( g_i \) were to be obtained from the altimeter's calibration mode II in which a uniform
signal level is presented to the input of the waveform sampler set so that the output dif-
ferences can be interpreted as gain variations from which a set of gain correction factors
can be obtained. We have already described (ref. 6) some of the problems with the gains
\( g_i \) derived from Calibrate Mode II of the altimeter and, more recently, we found that there
was an apparent gain change correlated with whether the altimeter's AGC word is above or
below 32 dB. The SEASAT altimeter's Calibration Mode I supplies the sampled point-target
response function \( s_r(t) \) but even here the sampler gain calibration problems cause some
concern.

There is work yet to finish on the SEASAT sampler gain problems, and no final values
for \( g_i \) and \( s_i \) in equation (33) can be given here. For purposes of this report, it will be
simply assumed that any necessary waveform sampler calibrations have already been applied
to the averaged sample data prior to the radar altimeter waveform modeled parameter
recovery procedure.

Separate Subroutines for Terms Within Waveform Convolution

I want to emphasize here the desirability of keeping separate waveform terms in
separate subroutines because of the freedom and flexibility in then changing any one of
these terms. Once there is a numerical convolution step required in a model waveform (and I cannot imagine that this will change in future altimeter systems), then the use of an FFT-convolution process means that adding an additional convolution will not even double the processing time. For example, Reference 5 gives the answer for the convolution of the flat-sea impulse response function $P_{FS}(t)$ and the surface elevation density function $q_s(t)$ for the particular forms of $P_{FS}(t)$ and $q_s(t)$ which I am now using. Yet I prefer to do this particular convolution numerically too (through FFT again) in order to preserve for future work the ability to try different functional forms for $q_s(t)$, forms which would invalidate the results from Reference 5.

Table III below lists typical computation time taken by the general six-parameter estimating iterative least-squares waveform fitting program on the Data General ECLIPSE S-200 computer system in Building E-106 at Wallops Flight Center for three different methods of generating the model waveform and its derivatives.

### TABLE III. COMPARISON OF TYPICAL SEASAT WAVEFORM FIT TIMES ON S-200 COMPUTER

<table>
<thead>
<tr>
<th>Method</th>
<th>Waveform Generation</th>
<th>Derivative Generation</th>
<th>Waveform Fit Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Analytical Numerical&quot;</td>
<td>$[P_{FS}(t)q_s(t)]$Analytical; then Simple Numerical Convolution with $s_r(t)$</td>
<td>Step-Change in Each Parameter, Reevaluate, for $\Delta$ Waveform $\Delta$ Parameter</td>
<td>2-3 minutes</td>
</tr>
<tr>
<td>&quot;Full Numerical by FFT&quot;</td>
<td>FFT for $P_{FS}(t), q_s(t)$ $s_r(t)$, Then Complex Product and Inverse FFT</td>
<td>Step-Change ...</td>
<td>15-25 seconds</td>
</tr>
</tbody>
</table>

In planning future altimeter data processing systems, the 20 seconds per fit of the "Full Numerical by FFT" entry in the table above would seem to pose a problem since the waveform data will probably be averaged and waveform fits done for typically every 2.5 seconds of data. The factor of 8 between processing time and real-time would create an
intolerable data processing bottleneck, but the existence of add-on hardware array processors (and their associated software) suggests a way out of the problem. In Table III, the additional time of the "Full Numerical by FFT" compared to the "Full Analytical" is primarily spent doing FFTs in the Fortran programs. Let's assume that all the rest of the waveform fitting processing time could be gotten below one second. This is probably a reasonable goal given the relative clumsiness of the Fortran source code written for the waveform fitting programs which were developed as research tools and not as production-oriented programs. Then let's estimate how many FFT and inverse FFT operations there are in one typical waveform fit, and estimate the operating time required by one representative array processor (ref. 26) which can be interfaced to Data General computers such as the ECLIPSE S-200 on which Table III is based.

To evaluate a model waveform, subroutine FILLV in Appendix C does two FFTs (on \(P_{FS}(t)\) and \(q_s(t)\), with transform of \(s_r(t)\) already having been performed once in the program initialization), then three complex multiplications (multiplication of the transforms of \(P_{FS}(t)\), \(q_s(t)\), and \(s_r(t)\) and of the \(T(t)\)-generated phase shift), and then one inverse FFT. Thus in effect three FFTs and three complex multiplications of 512-point array quantities are required to generate one set of model waveform samples. The MSP-2 array processor from Computer Design and Applications, Inc. (ref. 26) does a 512-point real FFT in 3.26 milliseconds and a 512-point complex product in 1.28 ms so that the total time per model waveform is about \(3(3.26) + 3(1.28)\) ms or about 13.6 ms. Then the derivatives for the six parameters fitted will require generation of four more model waveforms for step changes in different parameters in subroutine FILLD in Appendix C, so that one pass through the iteration loop in the least-squares fitting program uses a total of five waveform-generating steps or \(5(13.6) = 68\) ms. Then the average number of iterations in the waveform fitting process is generally eight or less so that the total waveform generating time in one complete waveform fit is around \(8(68) = 550\) milliseconds. If the array sizes discussed were to increase to 1024 instead of 512 this time would become about 1.16 seconds, slightly more than twice 550 milliseconds, based again on the MSP-2 processor literature (ref. 26).

Thus the entire waveform fitting procedure for one set of waveform sampler averages could probably be carried out in under two seconds of computer time on an ECLIPSE S-200 computer with an array processor. I use the Computer Design and Applications MSP-2 array processor only as an example to show that a 2.5 second average of waveform data could be processed in less than 2.5 seconds of computation. Other array processors are available for minicomputer or large mainframe computer systems at varying prices, speeds, and features. At least one unit available for ECLIPSE computers runs more than twice as fast as the MSP-2 processor in the example above. No array set-up or transfer times or other system overhead times have been included in the above examples, and the possible use of an array processor deserves a more detailed study for future altimeter waveform proc-
essing. Finally, it should be noted that the computation times required by many of the current 32-bit "super-minicomputers" is considerably less than the computation time of our ECLIPSE S-200 system; it might be possible that a newer computer would operate fast enough that no array processor is needed, but this again is a further issue for detailed study.

Simulation Program for Assessing Operation of Fitting Program

The model waveform generating procedure has been discussed earlier. Verification of correct operation of this program is fairly simple; one generates a waveform for a given parameter set and checks that the result matches what the theory predicts. One way to test the SEASAT waveform program FILLV of Appendix C is to write a small test program to produce a skewed Gaussian characterized by a rms \( \sigma_s \) and \( \lambda_s \) and to then take 1.5625 ns-separated samples of this function to form a point-target function \( s_r(t) \) in array SYS; then the waveform generated by FILLV for these SYS must agree closely with values calculated from the method described by Reference 5. This has been done for the SEASAT model waveform used here and the verification was satisfactory.

Another level of checking is needed as well to verify that the waveform least-squares fitting program recovers the correct parameters from the input waveform samples. For this check, a simulation program was developed which generates model waveform samples for one particular set of waveform parameters and then uses those sample values as input to the waveform fitting program; a comparison of the parameters recovered and the input parameters will then reveal any errors or biases induced by the fitting program. This simulation program is also extremely useful in determining the fundamental limits to parameter recoverability, limits due to the fundamental noise-like character of the individual radar return waveform.

Figure 5 sketches the simulation program used here to investigate waveform parameter recoverability. The average of \( N \) individual returns, for any particular specified time-point on the waveform, is expected to have a Gaussian distribution because of the central limit theorem. The individual return itself is described by an exponential distribution whose mean and standard deviation are equal, so the standard deviation of the \( N \)-average return is expected to have a standard deviation equal to the mean divided by \( \sqrt{N} \), and this noise modeling is built into the "noise loop" shown in Figure 5. The results of considerable experimenting with this simulation for the SEASAT work showed that the waveform fitting program operated with no appreciable biases and appeared tolerant of a reasonable range of first-guess parameter values. No further details will be presented here as this is the subject for a future separate report. The important point for this report is that this sort of simulation program must be produced and exercised for each different radar altimeter's particular conditions, and that work on waveform processing software is not
complete until the simulation program has exercised the waveform modeled parameter recovery programs.

**START**

Set: NR = # individual returns to be averaged; number of waveform parameters to be fitted for; range of waveform samplers to use in fits; NWF = number of noise waveforms to select; other initial conditions; and the initial, final, and delta values for the problem parameters such as SWH, pointing angle, and skewness.

Select next set of problem parameters such as SWH, $\xi$, and

Compute the 63 expected sampler values $Y_{e_i}$, $i=1...63$, for these problem parameters.

Zero the noise loop summing locations, set NFIT=0.

Do linefit, print recovered waveform parameters for the zero-noise limit case $Y_{e_i}$.

$\text{NFIT} \rightarrow \text{NFIT} + 1$

Select a set of random numbers $R_i$, $i=1...63$, with a unit mean and standard deviation of $1/\sqrt{\text{NR}}$.

Form the average-of-$\text{NR}$ return samples $Y_i$ by $Y_i = R_i Y_{e_i}$, $i=1...63$.

Do linefit, store recovered waveform parameter estimates.

Yes

$\text{NFIT} < \text{NWF}$?

Form mean and standard deviation for each of waveform fit parameters, and print out these results

Yes

More input problem parameter cases to run?

STOP

Figure 5. Simulation program for assessing waveform parameter recoverability.
REFERENCES


APPENDIX A

GENERAL DISCUSSION RELATED TO WAVEFORM FITTING SUBROUTINES

The following four appendixes in this report provide source listings for various subroutines in the waveform fitting procedures and for SEASAT-1 data. The main program, which is not provided here, handles obvious details of data input, averaging, and calibration. There are also some initial set-up details handled in the main program, setting the number of parameters to be fitted, lower and upper sampling gate limits if fewer than 63 waveform sampler results are to be used, setting the array for the system point-target response, and so on. The main program also sets the values of array SYS which carries the sampled system point-target response function $s_r(t)$, and sets the values of an array GUESS which carries first parameter guess for each parameter being determined in the iterative fit within SWHFIT. The array also carries the fixed values of the parameters not being fitted if fewer than six parameters are being determined.

Table A-1 lists subroutines used in the waveform fitting and indicates where their source listings are to be found. Table A-2 shows which of the subroutines are called by others; obviously subroutine SWHFIT is called from the main program. In the Data General FORTRAN used, "X" in column one of a source line means the line is to be ignored unless the /X switch is used in the FORTRAN compile step. Various debug statements in the source listings, with X in column one, call subroutine RLOUT and CPOUT which are not supplied. RLOUT and CPOUT are trivial array formatting and printout routines for real arrays and complex arrays respectively.

The calling list for SWHFIT includes the 63 waveform sampler corrected data averages in YIN(63), and sampling gate numbers NLO and NUP. Data for gates from NLO to NUP will be fitted, and normally NLO=1 and NUP=63. The calling list array WTY(63) carries relative weighting factors and, for the generally-used uniform data weighting all the elements of WTY are set to unity. Notice the array STPRM(7) which is set to the step sizes to be taken in producing the parameter derivative estimates by FILLD. Actually the first, fourth, and seventh values of STPRM have no meaning since the amplitude (first) and baseline (fourth) derivatives are found without making a step-change in parameters one and four, and the seventh value of STPRM was intended for the kurtosis which has not (to date) been written into the function in GTSEA. Notice also in the SWHFIT comments the description of JORDR(7), an integer array determining the order in which the parameters are to be found.

Part of the data communication in these subroutines is by calling lists and part by named COMMON areas. Table A-3 lists the contents of named COMMON area SSM4N, and Table
A-4 lists the contents of named COMMON area SYSTM. Two additional named common areas CON and CON1 are used within the FFT package only.

The individual subroutines have a number of additional comments provided in their source listings. It is hoped that these comments, the Tables A-1 through A-4, the remarks above, and the main body of this report should provide the reader an adequate acquaintance with the Wallops waveform modeled parameter recovery procedures.

### TABLE A-1. SUBROUTINES USED IN WAVEFORM MODELED PARAMETER RECOVERY PROGRAMS FOR SEASAT-1 DATA

<table>
<thead>
<tr>
<th>Routine Name</th>
<th>Where Provided</th>
<th>Purpose and Descriptive Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWHFIT</td>
<td>Appendix B</td>
<td>Function subroutine to fit up-to-6-parameter model waveform to corrected averaged waveform sampler data supplied by main calling program.</td>
</tr>
<tr>
<td>SYMINV</td>
<td>Appendix B</td>
<td>Symmetric matrix inversion subroutine used by SWHFIT.</td>
</tr>
<tr>
<td>FILLV</td>
<td>Appendix C</td>
<td>For given parameter set, generates model waveform at the 63 waveform sampler locations. Waveform is generated through FFT-numerical convolution of three separate terms.</td>
</tr>
<tr>
<td>FILLD</td>
<td>Appendix C</td>
<td>Uses FILLV, changes each parameter by specified step to form estimates of derivatives of model waveform with respect to each of the waveform fit parameters.</td>
</tr>
<tr>
<td>GTFSR</td>
<td>Appendix D</td>
<td>Produces flat-sea impulse response function, one of the terms convolved in FILLV.</td>
</tr>
<tr>
<td>GTSEA</td>
<td>Appendix D</td>
<td>Produces radar-observed surface elevation p.d.f., one of the terms convolved in FILLV.</td>
</tr>
<tr>
<td>FFA</td>
<td>Appendix E</td>
<td>Does Fourier transforms in radix 2-4-8 FFT package used.</td>
</tr>
<tr>
<td>FFS</td>
<td>Appendix E</td>
<td>Does inverse Fourier transforms in radix 2-4-8 FFT package used.</td>
</tr>
<tr>
<td>ORD1, ORD2,</td>
<td>Not Provided</td>
<td>Additional subroutines in radix 2-4-8 FFT package not provided in this report but available from Reference [25] or as described in the comments in FFA and FFS in Appendix E.</td>
</tr>
<tr>
<td>R2TR, R45SYN,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4TR, RBSYN,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4TR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4TR, R5TR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4OUT, CPOUT</td>
<td>Not provided</td>
<td>Routines for printing out real and complex arrays, respectively, and used for debugging only.</td>
</tr>
</tbody>
</table>
### TABLE A-2. RELATIONSHIPS AMONG WAVEFORM FITTING SUBROUTINES

Subroutines Called from Routines in Column at Left Side of this Table

<table>
<thead>
<tr>
<th>Waveform Fitting Subroutines</th>
<th>SYMINV</th>
<th>FILLV</th>
<th>FILLD</th>
<th>GTFSR</th>
<th>GTSEA</th>
<th>FFA</th>
<th>FFS</th>
<th>ORD1</th>
<th>ORD2</th>
<th>R2TR</th>
<th>R4SYN</th>
<th>R4TR</th>
<th>R8SYN</th>
<th>R8TR</th>
<th>RLOUT</th>
<th>CPOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWHFIT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILLV</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILLD</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFA</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes: ✓ = Always Needed.

* = Debug (optional) Routines RLOUT, or CPOUT to Print Values in Real or Complex Arrays Respectively.
TABLE A-3. DESCRIPTION OF VARIABLES IN NAMED COMMON AREA /SSM4N/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(7)</td>
<td>The seven final values of the up-to-seven parameters fitted to input waveform data. NA = number fitted. (The seventh is kurtosis, not yet written into GTSEA, so A(7) = 0 always.)</td>
</tr>
<tr>
<td>AEDIT(2,7)</td>
<td>Provides for possible lower, upper edit limits for the A(7).</td>
</tr>
<tr>
<td>CALM</td>
<td>&quot;Calm-Sea Constant&quot; or effective 1-a point-target response width used in finding SWH with &quot;analytical&quot; fitting. When numerical convolution (via FFT) is used, main program should set CALM = 0.</td>
</tr>
<tr>
<td>CNSTR(7)</td>
<td>Constraints on the fitted A(7) which in effect limit possible size of parameter changes in the fitting iterations. CNSTR values are set by a priori information, once, at start of main program.</td>
</tr>
<tr>
<td>CORRL(21)</td>
<td>Provides information on correlation of fitted parameters - probably of doubtful use in this non-linear problem.</td>
</tr>
<tr>
<td>GUESS(7)</td>
<td>Carries first-guess for waveform parameters to start iterative fit procedure, and carries fixed values for parameters not being fitted.</td>
</tr>
<tr>
<td>ITER</td>
<td>Indicates # iterations needed to fit data upon return from SWHFIT.</td>
</tr>
<tr>
<td>NA</td>
<td>Number of the A(7) being fitted, with maximum NA = 7 (if kurtosis is added, for example - current maximum NA is 6).</td>
</tr>
<tr>
<td>JORDR(7)</td>
<td>Set by main program, indicates order of parameters being fitted, and provides ability to fit different parameter combinations without rewriting source code.</td>
</tr>
<tr>
<td>SERSQ</td>
<td>Sum of squares of fit residuals, upon return from SWHFIT.</td>
</tr>
<tr>
<td>T(63)</td>
<td>Carries the 63 independent parameter (sampler time) values. In SEASAT, T(1) = -92.1875 and T(63) = 92.1875.</td>
</tr>
<tr>
<td>XCNST(7)</td>
<td>Provided for transporting different constants, but not used for anything significant at this time.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>NSYS</td>
<td>Value of index I after which all SYS(I) are zero.</td>
</tr>
<tr>
<td>NSCTR</td>
<td>Center-of-response index, the index within SYS to which is attached the time zero label.</td>
</tr>
<tr>
<td>SYS(514)</td>
<td>Contains the sampled point-target response in the elements SYS(1) through SYS(NSYS), with all remaining elements through SYS(514) zero before the first call to FILLV. After that, SYS contains the 512-point FFT transform of the input s_r(t) and must not be changed.</td>
</tr>
<tr>
<td>SYSID(21)</td>
<td>A comment line carrying information on how SYS was determined. SYSID is only of interest to the main program.</td>
</tr>
</tbody>
</table>
APPENDIX B.

SOURCE LISTINGS FOR SUBROUTINES SWHFIT AND SYMINV

SUBROUTINE SWHFIT3 IS EXACTLY SAME AS SWHFIT2 EXCEPT THAT
A NEW COMMON AREA /SYSMT/ HAS BEEN ADDED AND THE COMMUNICATION
WITH FILLV & FILLD CHANGED SLIGHTLY. THIS IS SO THAT THE
SYSTEM POINT-TARGET RESPONSE DATA FROM FILLV IS AVAILABLE
BACK AT THE MAIN CALLING PROGRAM.

SMALL CHANGE ON 10/30/80 CHANGED WHAT HAPPENED FOR A(3)
NEAR ZERO.

CHANGE ON 07/16/80 ADDED ARGUMENT XDBG TO ROUTINE CALLING
LIST; A VALUE OF 1 FOR XDBG SIGNALS PRINTOUT OF EACH
INDIVIDUAL ITERATION IN THE FIT.

THIS VERSION FOR THE 512-POINT TRANSFORM, AND COMMON AREA
SSMAN.

This is Revision of SWHSSN (REV.07/77/79) Now Incorporating
ffT-Convolution Method of Generating Seasat-1 Function Values
And Derivatives. The Subroutines Needed by SWHFIT Include:
FILLV3,FILLD3,FAST,FSST,FR2TR,FR4TR,FR4SYN,FRD1,
FORD2,GTSEA2,GTFSR2, and SYMINV.
NLO & NUP Are Now in Calling List of SWHFIT; These Define
Lower and the Upper Limits to the Number of Points Being
Fitted. It Is Required That NLO.GE.1, NUP.LE.63, and that
NLO.LT.(NUP-NLO).

AS IN SWHSSN, THE INTEGER ARRAY JORDR(7) DESCRIBES THE ORDER
IN WHICH PARAMETERS ARE TO BE FITTED. THE FIRST NA (NA.LE.7)
ELEMENTS OF JORDR() ARE TO BE INDIVIDUAL (AND DIFFERENT) VALUES
OF J, WHERE J HAS THE FOLLOWING MEANING:
J = 1 DESIGNATES AMPLITUDE
2 TIME ORIGIN
3 RISE TIME
4 EASELINE
5 SKEWNESS
6 OFF-HORIZON ANGLE
7 KURTOSIS.

A 7-PARAMETER NON-LINEAR LEAST-SQUARES FITTING FUNCTION SUB-
ROUTINE SPECIALIZED TO CASE OF SEASAT SIGNIFICANT WAVEHEIGHT
DETERMINATION. SUBROUTINE SWHFIT INCLUDES POSSIBILITY OF unequal
HEIGHTS FOR THE 63 INPUT DATA POINTS. WEIGHTING BY INVERSE
OF VARIANCE ESTIMATES, FOR EXAMPLE, PRODUCES IN EFFECT A
MAXIMUM-LIKELIHOOD FIT. THE INDICATOR IIR IS 1 ON THE RETURN
FROM A SUCCESSFUL FIT; OTHERWISE IIR CAN INDICATE THE TYPE OF
ERROR AS INDICATED IN COMMENTS BELOW.

THERE'S ALSO PROVISION FOR CALCULATING CORRELATION COEFFICIENTS,
AND ABILITY TO ADD A PRIORI FIT CONSTRAINTS, RESULTING FROM
CONVERSATIONS WITH BILL WELLS OF WASSI, INC.
SUBROUTINE SWFMAT USES SYMMETRIC MATRIX INVERSION ROUTINE SYMV.
TO INVERT SYMMETRIC MATRIX XMAT IN PROBLEM. FUNCTIONAL FORM TO
BE FITTED IS SUPPLIED BY SUBROUTINE FILLY AND THE NEEDED
DERIVATIVES BY SUBROUTINE MILLD.

FUNCTION SWFMAT(NLO,NUP,VIN,WVT,IXDBG,IER)

COMPILER STATIC

REAL STPM(7)/1.0, .25, .20, 1.0, .05, .05/

DIMENSION VIN(63),WVT(63),XMAT(7),BCOLM(7),PVECT(7),
1 WT(63),PSYM(7),QSYM(7),MSYM(7),MDST(7),AMP(7),AMP(7),
2 VAL(63),CVTV(63,7)

COMMON /SVSTM/ SYS(514),NSYS,NSCTR

COMMON /SSEMIV/ A(7),XCNST(7),T(63),NA,ITER,SERSQ,CALM,corr(21),
1 GUESS(7),CNSTR(7),JORDR(7),AEDIT(7,7)

X WRITE FREE (12) ' <15 > AT SWFMAT INPUT, WTV VALUES ARE:'
X CALL RLOUT(12,WVT,63)
C... SET INITIAL VALUES, LIMITS

IF (NLO.LT.1) NLO=1
IF (NUP.GT.63) NUP=63

ERLIM=0.001
LIMIT=50
ITER=0
SWFMAT=-3333.
IER=-10
WT=0.
SWMIN=1.E66
SERSQ=SCHIN

DO 1=1,NLO

DO 10 I=1,NUP

WT(I)=WT(I)+WTV(I)

V=VIN(I)

IF ((VI.LT.-25.),OR.(VI.GT.500.)) RETURN

DO 5 CONTINUE

DO 10 I=NLO,NUP

WT(I)=WTV(I)/WT

C... SET INITIAL FIT PARAMETER ESTIMATES. SET ALL TERMS FROM GUESS(7)

C... WHICH IS ASSUMED TO SET VALUES NOT BEING FITTED AS WELL AS

C... THE INITIAL GUESSES FOR THOSE BEING FITTED.
DO 9 I=1,7
A(I)=GUESS(I)

C--- AT THIS POINT IN PROGRAM, SET THE INVERSE-VARIANCE CONSTRAINTS
C--- TO BE ADDED TO THE ON-DIAGONAL TERMS BELOW AT "DO 52" STATEMENT.
C--- PUT TEST TO AVOID AN INPUT STANDARD DEVIATION ESTIMATE LESS THAN
C--- ABOUT 0.033 .

IKJ=0
DO 12 II=1,NA
I=JCRDR(II)
IF (I.EQ.6) IKJ=II
C... CHECKS WHETHER POINTING MCLE IS ONE OF PARAMETERS BEING FITTED,
C... SINCE THERE MAY BE CORRECTION SUBSEQUENTLY TO AVOID A NEGATIVE ANGLE.
VI=CNSTR(I)**2
IF (VI.LT.0.001) VI=0.001
CNST(I)=1./VI
AKEEP(I)=A(I)

C--- FILLV GETS THE 63 SAMPLED VALUES FROM THE FFT-CONVOLUTION
C--- PROGRAM. THESE ARE IN ASCENDING ORDER IN INDEPENDENT VARIABLE.
CALL FILLV(VAL)
EOLD=0.
IER=1
DO 13 I=HLO,NUP
VI=VIN(I)-VAL(I)
EOLD=EOLD + VI*VI*WT(I)
13
C

WRITE FREE (12) '<15> IN SWMRT, THE VIN VALUES ARE :
CALL RLOUT(12,VIN,63)
WRITE FREE (12) '<15> AND FIRST-CALL VAL FROM FILLV ARE :
CALL RLOUT(12,VAL,63)

SOMIN=EOLD
SOMAX=EOLD
IF (IXDBG.NE.1) GO TO 11
C
WRITE FREE (12) '<15> SWMRT; NA =',NA,' ITERATION RESULTS ARE:
WRITE (12,14) ITER,EOLD,A
14 FORMAT (' ITER =',I3,',', SERIES =',G14.6,', A(.) = :',(' ',7F11.5))
C
C... STATEMENT 11 IS RETURN POINT FOR NEXT ITERATION IN FITTING .
CONTINUE
C
C... ZERO UPPER PART, SYMMETRIC MATRIX .
DO 20 I=1,NA
DO 19 J=1,1
XMAT(J,I)=0.
19
20 ECOLN(I)=0.
t=1.
C---PUTTING A COMMENT IN FOLLOWING CARD WILL REMOVE EFFECTS OF
C--- THIS ATTEMPT (2/17/78) AT REDUCING SIZE OF FIRST FEW "STEPS"
C--- IN PARAMETER SPACE . .
IF (ITER.LT.4) XFRCT=(1.+FLOAT(ITER))/6.

C ELIM=ELIMIT*XFRCT

C--- SUBROUTINE FILLD SETS UP THE (63,7) DERIVATIVE ARRAY BY MAKING
C--- STEPS IN THE VALUES OF THE PARAMETERS A(7); THE STEP SIZES
C--- TAKEN ARE CARRIED BY STP(7), AND THE ORDER OF THE DERIVATIVES
C--- IN DRV(63,7) IS SET BY JORDR. FILLD MUST HAVE BEEN CALLED BEFORE
C--- FILLD; THE HLD VALUE IN VAL(63) FROM THE CALL TO FILLD IS USED IN
C--- FILLING DRV(63,7). INCIDENTALLY, NUMERICAL DERIVATIVE IS NOT
C--- DONE IN CASE OF THE AMPLITUDE AND BASELINE, SO THE VALUES SET IN
C--- STP(1) AND STP(4) ARE IRRELEVANT.

CALL FILLD(VAL,DRV,STP)

C--- "DO 30" LOOP FILLS UPPER HALF OF THE SYMMETRIC MATRIX, ALSO THE
C--- COLUMN VECTOR . .
DO 30 JP=NLO,NUP
  WT=WT(JP)
  DV=VIN(JP)-VAL(JP)
DO 35 CONTINUE

DO 30 JA=1,NA
  PVECT(JA)=DV
  DF=DFAL(JA)
DO 35 CONTINUE

C--- NOW ADD ON-DIAGONAL CONSTRAINT ELEMENTS TO THE SYMMETRIC MATRIX;
C--- THESE ARE FROM THE A PRIORI INFORMATION ON VARIATION EXPECTED IN
C--- THE PARAMETERS TO BE FITTED (REF. CONVERSATION WITH BILL WELLS OF
C--- MASCY, ABOUT 65/25/75 ). THE CONSTRAINT IS THE INVERSE OF THE INPUT
C--- VARIANCE ESTIMATE. MINIMUM ALLOWED VARIANCE OF 0.001 .

DO 774 JA,FAL
  XMAT(JA,J)=XMAT(JA,J)+DF
  CALL SYSTMAT(XMAT,FAL,FAL)
  IF (FAI.LT.0) WRITE FREE (12) 'BEFORE MATRIX INVERSION, INPUT MATRIX ':
DO 774

DO 52 I=1,NA
  JJ=JORDR(I)
  XMAT(I,I)=XMAT(I,I)+CONST(JJ)

C--- CALL HERE THE SYMMETRIC MATRIX INVERSION ROUTINE FROM J. MCMILLAN

C--- WRITE FREE (12) 'BEFORE MATRIX INVERSION, OUTPUT MATRIX '
DO 780 J=1,NA
  XMAT(J,J)=XMAT(J,J)+CONST(J)
  CALL SYSTMAT(XMAT,FAIL,FAL)
  IF (FAIL.NE.0) WRITE FREE (12) 'MATRIX INVERSION FAILED . .'

C--- WRITE FREE (12) 'AFTER MATRIX INVERSION, OUTPUT MATRIX '
DO 780
WRITE (12,780) (XMAT(J,I), J=1,I)

WRITE FRE (12) 'SWMV IFAIL SHOULD = 0; IT IS ',IFAIL

IF (IFAIL.NE.0) GO TO 10000

DO 66 I=1,NA
  II=JORDR(I)
  ACLM=ACL/8.
  IF (0.2 J=1,NA
  JLT.I) GO TO 60
  A(I,J) = XMAT(I,J)*BCOLM(J)
  GO TO 62
0253 65 ACLM=ACL+XMAT(I,J)*BCOLM(J)
0254 62 CONTINUE

IF (II.NE.6) GO TO 64

C--- THE II=6 PARAMETER 'S POINTING ANGLE; FOLLOWING TREATMENT IS AD
C--- HOC AND SPECIFIC TO SEASAT-1 CASE (07/25/80).

IF (ABS(A(6)).LT.0.025) ACLM=ACLM/5.
A(6) = A(6) + XFRCT*ACLM
GO TO 65

A(II) = A(II) + XFRCT*ACLM
CONTINUE

IF (A(3).LT.1.E-06) A(3) = 1.E-06

CALL FILLV(VA!)

SEPSO=0.
DO 175 I=NLG,NUP
  VI=Vin(I) - VAL(I)
  SERSO=SEPSO+VI*VI*WT(I)
  ITER=ITER+1
175 CONTINUE

IF (IXD3G.EQ.1) WRITE (12,14) ITER,SEPSO,A

C--- PUT FOLLOWING STEP IN TO AVOID NEGATIVE RISE TIME.

IF (A(3).LT.1.E-06) A(3) = 1.E-06

CALL FILLV(VAL)

C--- RECALCULATE VALUES OF THE SAMPLED WAVEFORM FUNCTION FOR THE NEW,
C--- UPDATED ESTIMATES OF THE PARAMETERS A(7).

SEPSO=0.
DO 175 I=NLG,NUP
  VI=Vin(I) - VAL(I)
  SERSO=SEPSO+VI*VI*WT(I)
  ITER=ITER+1
175 CONTINUE

IF (IXD3G.EQ.1) WRITE (12,14) ITER,SEPSO,A

C--- CHECK THAT WE KEEP COEFFICIENTS PRODUCING MINIMUM SUM ERRORS**2

IF (SEPSO.GE.SGMIN) GO TO 195

DO 189 I=1,NA
  DO 179 J=1,1
  XKEEP(J,I)=XMAT(J,I)
  II=JORDR(I)
  AKEEP(II)=A(II)
  SGMIN=SEPSO
179 CONTINUE
189 CONTINUE

B5
C.... TEST; AVOID TRYING FOR ABSURDLY SMALL RESIDUALS ABOUT FIT . . .

0300 IF (SERSQ.LT.4.4E-65) GO TO 3020
0301 C
0302 IF (ITER.GE.LIMIT) GO TO 2000
0303 IF (SERSQ.GT.EOLD) GO TO 5000
0304 C
0305 IF ( ((EOLD-SERSQ)/EOLD).LE.ELIM ) GO TO 3000
0306 190 EOLD=SERSQ
0307 GO TO 10
0308 C
0309 C
0310 C--- FOR THE SEASAT WAVEFORM CASE, CHECK IF ERRORS**2 INCREASED; DON'T
0311 C--- MAKE ERROR EXIT IF (FRACTIONAL) INCREASE IS LESS THAN 10 TIMES
0312 C--- LIMIT. ALSO CHECK ABS. MAGNITUDE OF SERSQ; VALUE LESS THAN .0099
0313 C--- IS A INDIVIDUAL SEH GATE STANDARD DEVIATION OF ABOUT .015 M,
0314 C--- WHICH WE ASSUME TO BE AN ADEQUATE LOWER LIMIT TO THE SEASAT
0315 C--- SITUATION WHEN USING ONLY LAST 45 GATES.
0316 5000 IF (SERSQ.LE.0.0099) GO TO 3000
0317 IF (ABS((SERSQ-EOLD)/EOLD).LE.10.`ERLIM) GO TO 190
0318 C
0319 IF EIR=-1
0320 C--- SAME ERRORS**2 INCREASED
0321 C
0322 GO TO 2500
0323 C
0324 C
0325 1000 IER=-2
0326 C--- MATRIX INVERSION FAILURE (SINGULAR MATRIX . . .)
0327 C
0328 GO TO 2500
0329 2500 IER=-3
0330 C--- ITERATION COUNT EXCEEDED
0331 C
0332 C
0333 C
0334 C... IF ITER .GT. 2, FIGURE THAT SOME SORT OF SOLUTION EXISTS, SO SET
0335 C,. IER=1 AND RETRIEVE THE MINIMUM-VALUE-PRODUCING SET OF A(.). AND THE
0336 C,. RESULTING XMAT(...). VALUES IF NECESSARY . .
0337 2550 IF (ITER.LE.2) GO TO 4500
0338 C
0339 IF EIR=1
0340 3000 DO 2510 J=1,NA
0341 DO 2505 J=1,KA
0342 2505 XMAT(J,1)=XKEEP(J,1)
0343 II=JORDR(J)
0344 2510 A(J)=XKEEP(II)
0345 C
0346 C
0347 3000 CONTINUE
0348 C
0349 C--- USE VALUES FROM XMAT (AT LAST ITERATION & AFTER INVERSION) TO
0350 C--- FIND CORRELATIONS WHICH WILL THEN BE SET INTO ARRAY CORRL(21) IN
0351 C--- ORDER : 2.1 .3.1 .3.2 .4.1 .4.2 .4.3. ETC. . NOTE THAT FIRST THE
0352 C--- SQUARE ROOTS OF DIAGONAL ELEMENTS WILL BE TAKEN, FOR CONVENIENCE.
0353 C--- ALSO NOTE THAT ORDER IN THIS CORRELATION ARRAY IS IN TERMS OF
0354 C--- THE ORDER IN WHICH THE PARAMETERS WERE FITTED, NOT THE ORDER IN
0355 C--- WHICH THEY ARE IN A(.) .
0356 C
DO 3001 I=1,NA
3001 CORR(I)=0.
C
IJI=0
DO 3005 J=2,NA
J=M-J-1
DO 3005 I=1,JM
IJ=IJ+1
CORR(IJ)=XMAT(I,J)/(XMAT(I,I)*XMAT(J,J))
CONTINUE
C
C--- 3010 REACHED WHEN LINEFIT CONVERGED, PRODUCED PARAMETER ESTIMATES.
CONTINUE
C
C... CHECK LINEFIT PARAMETERS AGAINST EDIT LIMITS, SIGNAL BY IER.GT.1
C
DO 3012 II=1,NA
I=JORDR(II)
VI=A(I)
3012 IF ((VI.LT.AEDIT(1,I)).OR.(VI.GT.AEDIT(2,I))) GO TO 4000
GO TO 4000
C
IER=1+I
C
C--- COMPUTE SWH ESTIMATE, WITH POSSIBILITY OF NEGATIVE SWH...
DIFQ=A(I)**2-CALM*CALM
SWHIT=9.6*SIGN(SQRT(ABS(DIFQ)),DIFQ)
C
RETURN
END
SUBROUTINE SYMINV (A,N,IFAIL,NROW,P,Q,M)

C...PROGRAMMED BY:  JIM MCMILLAN - WASC - REVISED 03/08/78

C...PURPOSE:  TO COMPUTE THE INVERSE OF A SYMMETRIC MATRIX

DIMENSION A(NROW,1),P(1),Q(1),M(1)

C...BEGIN CALCULATION.

DO 1 I=1,N
 1 M(I) = 1

C...BEGIN CALCULATION.

DO 10 I = 1,N
 10 CONTINUE

C...SEARCH FOR PIVOT.

BIG = 0.0

DO 4 J=1,N
 4 CONTINUE

C...SEARCH FOR PIVOT.

IF (TEST-BIG) 4,4,2

4 IF (M(J)) 15,4,3

3 BIG = TEST

K = J

4 CONTINUE

C...PREPARATION FOR ELIMINATION STEP NO. 1.

M(K) = 0

G(K) = 1.0 / A(K,K)

P(K) = 1.0

A(K,K) = 0.0

K1 = K + 1

KM1 = K - 1

IF (K1) 15,8,5

DO 7 J=1,KM1
 7 CONTINUE

C...PREPARATION FOR ELIMINATION STEP NO. 1.

P(J) = A(J,K)

Q(J) = A(J,K) * Q(K)

IF (M(J)) 15,7,6

6 Q(J) = -Q(J)

7 A(J,K) = 0.0

8 IF (K1) 9,13,15

DO 12 J=K1,N
 12 CONTINUE

C...PREPARATION FOR ELIMINATION STEP NO. 1.

P(J) = A(K,J)

IF (M(J)) 15,10,11

10 P(J) = -P(J)

11 Q(J) = -A(K,J) * Q(K)

12 A(K,J) = 0.0

IF (K) 15,7,6

C...PREPARATION FOR ELIMINATION STEP NO. 1.

13 DO 16 J=1,N
 16 CONTINUE

C...ERROR EXIT.

C...ERROR EXIT.

C...ERROR EXIT.

C...ERROR EXIT.

C...ERROR EXIT.

RETURN

RETURN

END
APPENDIX C.

SOURCE LISTINGS FOR SUBROUTINES FILLV AND FILLD

THIS VERSION USES THE RADIX 8-4-2 FFT ROUTINES; THE VERSION IN FILE FILLV2.FR USES THE RADIX 4-2 ROUTINES. FILLV3 USES THE VALUES OF SYS(514) WHICH ARE PASSED BY A NEW LABELLED COMMON ARRAY /SYS(514/.

SUBROUTINE EVALUATES SEASAT-1 WAVEFORMS USING FFT TECHNIQUES TO PERFORM CONVOLUTION OF: 1) SYSTEM POINT-TARGET RESPONSE, 2) SEA-SURFACE ELEVATION DISTRIBUTION, AND 3) FLAT-SEA RESPONSE. THIS ROUTINE IS SET UP FOR A 512-POINT TRANSFORM, AND USES FFT ROUTINES FFA &FFS TOGETHER WITH REQUIRED SUBROUTINES FOR FFA AND FFS (THESE ARE R2TR,R4TR,R8TR,R4SYN,R8SYN, ORDI, AND ORD2).

SUBROUTINE FILLV(VAL)

INTEGER IIST/0/,NNP/512/,NP2/514/,NC2/257/

REAL VAL(63).S:A(514),FSR(514),RCS(514)

COMPLEX CTSVS(257),CTSEA(257),CTFSR(257),CTRES(257),CPHAS

EQUIVALENCE (S:"),CTSVS(1),(SEA(1),CTSEA(1)),(FSR(1),CTFSR(1)),

1 (RES(1),CTRES(1)),(A(1),AMPLI),(A(2),TIMO),(A(3),SIGMA),

2 (A(4),BSLIN),(A(5),XLMDA),(A(6),XIDEG),(A(7),XKURT)

COMMON /SSMN/ A(7),XONST(T(63),NA,ITER,SEISO,CLAM,CORRL(21),
1 GUESS7),CNSTR(T),JONDR(T),AEDIT(2,7)

COMMON /SYS(514)/ SYS(514),NSYS,NSCTR

C... NSYS=X VALUES AT THE POINT AFTER WHICH THE REST OF THE 514 VALUES EQUAL ZERO, AND NSCTR=INDEX OF PEAK OF THE POINT-TARGET RESPONSE.

C... WARNNING ! ! !

C... SYS( ) CONTAINS THE TRANSFORM AFTER THE FIRST CALL TO FILLV3, SO DO NOT CHANGE SYS( ) IN THE EXTERNAL PROGRAM.

C... ALSO, IF THE INFORMATION IS TO BE PRINTED OUT, DO THIS BEFORE THE FIRST FILLV3 CALL.

C... NNP IS # POINTS IN TRANSFORM (MUST BE POWER OF 2), NP2 IS 2 MORE THAN NNP, AND NC2 IS # COMPLEX TRANSFORM VALUES WHICH WILL RESULT FROM THIS.

CALL DATE(JDATE,1)

JDATE(3) = JDATE(3) - 1900

CALL TIME(JDATE,1)
0039 X WRITE (12,133) JTIME,JOATE
0060 X 133 FORMAT(‘ FILLV3 DEBUG AT ‘,I2,2(‘:‘,I2),’ ON ‘,I2,2(‘/‘,I2))
0061 C
0062 IF (11ST.NE.0) GO TO 250
0063 I1ST=1
0064 SMSYS=0.
0065 DO 150 I=1,NSYS
0066 150 SMSYS = SMSYS + SYS(I)
0067 J = NSYS + 1
0068 DO 155 I=J,NSYS
0069 155 SYS(I) = 0.
0070 C
0071 C..... IF DESIRED, CAN REPLACE THE BUILT-IN SAMPLED SYSTEM RESPONSE
0072 C FUNCTION HERE BY ANOTHER ROUTINE GTSYS BY EXECUTING
0073 C THE FOLLOWING STATEMENT (I.E., REMOVE "C" IN COL.1).
0074 C CALL GTSYS(NNP,SYS,SMSYS)
0075 C
0076 C--- NOW DO THE TRANSFORM OF SYSTEM IMPULSE RESPONSE
0077 X WRITE FREE (12) ’<15> INPUT REAL SYS(.):’
0078 X WRITE FREE (12) ‘ SMSYS = ’,SMSYS
0079 X CALL RLOUT(12(SYS,NNP)
0080 X CALL FFA(SYS,NNP)
0081 X WRITE FREE (12) ’<15> COMPLEX, TRANSFORMED SYS(.):’
0082 X CALL CPOUT(12,SYS,NC2)
0083 C
0084 C------ NOW SET UP SEA SURFACE ELEVATION DISTRIBUTION.
0085 250 CALL GTSEA(NNP,SEA,SMSEA)
0086 C --- NOW DO TRANSFORM OF SEA SURFACE ELEVATION DISTRIBUTION.
0087 X WRITE FREE (12) ’<15> INPUT REAL SEA-SURFACE SEA(.):’
0088 X WRITE FREE (12) ‘ SMSEA = ’,SMSEA
0089 X CALL RLOUT(12,SEA,NNP)
0090 X CALL FFA(SEA,NNP)
0091 X WRITE FREE (12) ’<15> COMPLEX, TRANSFORMED SEA(.):’
0092 X CALL CPOUT(12,SEA,NC2)
0093 C
0094 C------ SET UP THE FLAT-SEA RESPONSE FUNCTION.
0095 X CALL GTFSR(NNP,FSR,SNFSR)
0096 C --- TRANSFORM THE FLAT-SEA RESPONSE FUNCTION
0097 X WRITE FREE (12) ’<15> FLAT-SEA IMPULSE RESPONSE FSR(.):’
0098 X WRITE FREE (12) ‘ SNFSR = ’,SNFSR
0099 X CALL RLOUT(12,FSR,NNP)
0100 X CALL FFA(FSR,NNP)
0101 X WRITE FREE (12) ’<15> COMPLEX, TRANSFORMED FSR(.):’
0102 X CALL CPOUT(12,FSR,NC2)
0103 C
0104 C------ FORM AMPLITUDE NORMALIZATION ANORM, THEN SET UP PHASE MULTI-
0105 C PLIER DELTA FACTOR DFHI (XNCTR IS FOR POSSIBLE LATER SHIFT OF
0106 C ENTIRE RESULT IF DESIRED. AS NOW SET, FIRST 119 VALUES OF RCX
0107 C ARE THOSE SPANNING THE DESIRED 63 SAMPLER SET OF SEASAT-1 AND
0108 C THE CENTER IS AT THE 60TH GATE IF XNCTR IS 0.)
0109 X XNCTR = 0.
0110 PHI = 0.
0111 DFHI = -( (XNCTR=24.5-FLOAT(NSCTR)) + TIX3/1.5625)
0112 1 6.2931850/FLOAT(NNP)
0113 ANORM = 1./ (SMSYS*SMSEA)
0114 CTRES(I) = CMPLX(ANORM,0.)*CTS(1)*CTSEA(I)*CTFSR(I)
0115 DO 500 I=2,NC2
0116 PHI = PHI + DFHI
0117 CPHAS = CMPLX(ANORM*COS(PHI),ANORM*SIN(PHI))
0118 500 CTRES(I) = CPHAS*CTSV(I)*CTSEA(I)*CTFSR(I)
DO 119 C
0120 C-------- ABOVE WAS A COMPLEX MULTIPLICATION (IN TRANSFORM DOMAIN)
0121 C
0122 C--- DO THE INVERSE TRANSFORM TO GET FINAL CONVOLUTION RESULT
0123 C
0124 X WRITE FREE (12) ' (15) COMPLEX, TRANSFORM RESULT RES(.):'
0125 X WRITE FREE (12) ' ANORM = ',ANORM
0126 X CALL CPOUT(12,RES,NC2)
0127 X CALL FFS(RES,N2P)
0128 X WRITE FREE (12) ' (15) FINAL REAL RESULT RES(.):'
0129 X CALL RLOUT(12,RES,N2P)
0130 C
0131 IF (ABS(BSLIN).LE.(AMPLI*0.1E-05)) GO TO 550
0132 DO 530 I=1,N2P
0133 530 RES(I) = RES(I) • BSLIN
0134 550 CONTINUE
0135 C
0136 C TRANSFER DATA TO FINAL OUTPUT ARRAY VAL(63) AND WRITE OUT.
0137 C
0138 C DO 610 I=1,29
0139 DO 610 J=I+1
0140 610 VAL(I+34)=RES(J+61)
0141 610 VAL(I)=RES(J-1)
0142 C
0143 DO 620 I=30,34
0144 620 VAL(I)=RES(I-28)
0145 C
0146 620 VAL(I)=RES(I+28)
0147 C
0148 C
0149 C--- NOW RETURN TO CALLING PROGRAM
0150 C
0151 RETURN
0152 END
A subroutine to fill an array DRV(63,7) containing 63 sample values of the up to 7 derivatives for Seasat waveform case; the order of the derivative terms in DRV is set by JORDR(J).

FILLD is a companion to FILLV; it requires that FILLV has been called already. See further comments in the fitting routine which calls FILLD and FILLV. These routines are part of the new programs using the FFT-convolution technique within waveform generation. Debug printout to (assumed already opened) device 12 is provided on compile switch.

FILLD2 requires subroutines FILLV, GTSYS, GTSEA2, GTFSR2, FR2TR, FR4T, FR4SYN, FORD1, and FORD2.

The derivatives are numerically found by stepping A(7), the parameter values by an amount set by STRM (except that first and fourth terms, amplitude & baseline, are found differently).

Subroutine FILLD(VAL,DRV,STPRM)

Compiler static

COMMON /SYS/ SYS(514),NSYS,NSCTR

COMMON /SSYSN/ A(7),XSNST(7),T(63),NA,ITER,SEISS,CALM,CONNR(21),
I GUESS(7),CNSTR(7), JORDR(7), AEDIT(2,7)

EQUIVALENCE (AMPLI,A(1)),(BSLIN,A(4))

REAL VAL(2),DRV(63,7),STPRM(7)

REAL XTRM(63)

WRITE FREE (12) ' DEBUG IN FILLD; CALL FILLV & PRINT RESULTS ;'

CALL FILLV(VAL)

CALL RLOUT(12,VAL,63)

C--- LOOP TO 400 FOR THE NA DERIVATIVES NEEDED.

DO 400 K=1,NA

J=JORDR(K)
C--- DO NUMERICAL STEP PROCEDURE AT 300 FOR EVERYTHING EXCEPT AMPLI AND BSLIN; FOR THESE WE KNOW EASIER WAY TO GET ANSWER.

DO 100 J & K = ',J,K

GO TO (105,300,200,300,300,200,300), J

C--- 100 IS STATEMENT REACHED WHEN AMPLI DERIV IS NEEDED.

100 WRITE FREE (12) ' FILLD DEBUG: AT COMPUTED GO TO, J & K = ',J,K

GO TO (100,300,200,300,300,200,300), J

C

C--- 200 IS STATEMENT REACHED WHEN BSLIN DERIV IS NEEDED.

200 WRITE FREE (12) ' FILLD DEBUG: AT STATEMENT # 200'

IF (TMP.LT.1.E-05) TMP=1.E-05

DO 110 I=1,63

110 DRV(I,K)=(VAL(I)-BSLIN)/TMP

GO TO 400

C

C--- 300 IS STATEMENT FOR NUMERICAL STEP ESTIMATIONS OF THE DERIVATIVE.

300 WRITE FREE (12) ' FILLD DEBUG: AT STATEMENT # 300'

ATMP=AMP

STEP=STPM(J)

A(J)=A(J)+STEP

C--- ABOVE STATEMENTS SWAPPED IN NEW VALUE FOR THE DESIRED A(J).

C--- NOW STORE ELEMENTS OF VAL(.) TEMPORARILY IN DRV(..J)

X WRITE FREE (12) ' FILLD DEBUG: J & A(J) = ',J,A(J)

DO 310 I=1,63

310 DRV(I,K)=VAL(I)

C--- GET NEW WAVEFORM WITH NEW (TEMPORARY, STEPPED) PARAMETER VALUES.

CALL FILLV(VG,L)

AND FOR THESE VALUES, FUNCTION IS :

CALL RLOJ(T,VAL,53)

C--- NOW FORM DERIV. ESTIMATES & SWAP VALUES BACK TO ORIGINAL...

DO 320 I=1,63

320 VAL(I)=TMP

A(J)=ATMP

C

C

C

X WRITE FREE (12) ' <LF> STEP-DERIV. VALUES ARE :

CALL RLOJ(T,XTMP,63)
WRITE FREE (12) 'RESTORED FUNCTION VALUES :
CALL RLOUT(12,VAL,63)
--- BOTTOM OF MAIN DERIVATIVE LOOP
CONTINUE
RETURN
END
APPENDIX D.
SOURCE LISTINGS FOR SUBROUTINES GTFSK AND GTSEA

0231 C   DP:GHAYNE:GTFSR2.FR   REVISED 11/13/80, 1505 HRS
0232 C
0233 C   G.S.HAYNE   APPL.SCI.ASSOC'S   07/16/80
0234 C
0235 C   THIS VERSION ADMITS A NEGATIVE ANGLE, BUT FAKE THE RESULT
0236 C   TO GIVE A FASTER-THAN-ZERO-DEGREES PLATEAU DECAY.   07/16/80
0237 C
0238 C----- SUBROUTINE GTFSR2 FILLS THE FLAT-SURFACE RESPONSE, USING IO
0239 C   TERM ONLY FROM EXPANSION IN GARY BROWN'S PAPER. A POWER SERIES
0240 C   FROM ABRAMOWITZ & STEGUN IS USED TO EVALUATE IO. THIS VERSION OF
0241 C   GTFSR USES 230 NON-ZERO VALUES OF THE FLAT-SURFACE RESPONSE, AND
0242 C   ASSUMES THAT NNP > 230. GTFSR2 IS DERIVED FROM GTFSR BUT IS
0243 C   FOR THE NEW FFT-BASED CONVOLUTION USING ONLY 512-POINT
0244 C   FFT.
0245 C
0246 SUBROUTINE GTFSR(NNP,FSR,SMFSR)
0247     REAL FSR(1)
0248     COMMON /SS44N/ AMPLI,TIME,SGMA,BSLIN,XMMA,XIDEG,XKURT
0249     NFSR = 230
0250     SMFSR = 0.
0251     DT = 1.5623
0252     T = -DT/2.
0253 C
0254 C... TEST FOR (IMPOSSIBLE) NEGATIVE ANGLE; IF PRESENT, CHOOSE BRANCH WHICH
0255 C... EFFECTIVELY INCREASES THE DLTA AT < ZERO DEGREES POINTING (CHANGES
0256 C... MADE ON 07/16/80)
0257 C
0258 C IF (XIDEC.GT.0.) GO TO 20
0259 C
0260 C... THE FOLLOWING STATEMENT CAUSES DLTA TO INCREASE BY A FACTOR OF
0261 C.. TWO FOR 1 DEGREE (FICTICIOUS) NEGATIVE SEASAT ANGLE...
0262 C
0263   DLTA = 2.56496E-03*(1.-XIDEG)
0264   DO 16 J=1,NFSR
0265       T = T + DT
0266       Z = AMPLI*EXP(-DLTA*T)
0267   15  SMFSR = SMFSR + Z
0268   GO TO 16
0269 C
0270   20  X2RAD = XIDEG/28.64789
0271   21  BETA = 4.35331*SN(X2RAD)
0272   22  DLTA = 2.66496E-03*CUS(X2RAD)
0273   23  DO 30 J=1,NFSR
0274       T = T + DT
0275       Z = BETA*SQRT(T)
0276   24  IF (Z.GT.3.75) GO TO 23
0277   25  Z = Z/14.6298
0279       +Z*(0.2659732+Z*(0.0360766+Z*(0.0045813))))))
0280   27  GO TO 27
0281   28  A = EXP(Z)/SQRT(Z)
0282   29  Z = 3.75/Z
0283   30  A = A*Z*0.3998423-Z*(0.03988024+Z*(0.00362018
0284       +Z*(0.0163801-Z*(0.0103155-Z*(0.02282967-Z*(0.0289512
0285       +Z*(0.01787654-Z*(0.0042059))))))))))
0286   31  Z = AMPLI*EXP(-DLTA*T)*A
FSR(J) = Z
30 SMFSR = SMFSR + Z
C
35 CONTINUE
C
C-------- FILL REST OF THE ARRAY WITH ZEROES
K = NFSR + 1
NP2 = NP2 + 2
DO 40 J = K, NP2
40 FSR(J) = 0.
IIST = 1
RETURN
C
END
SUBROUTINE GTSEA2(FILL ARRAY WITH A SKewed GAUSSIAN SURFACE

ELEVATION DISTRIBUTION, CENTERED ON THE SAMPLE READER 96.

ZER0ES ENTERED IN ALL OTHER ELEMENTS THAN IN INTERVAL 1 - 171.

ASSUMES NNP > 171. THIS ROUTINE SAME AS GTSEA.FR EXCEPT FOR DIFFERENT
COMMON NAME AND FOR THE DISTRIBUTION CENTERED AT 171. IT IS INTENDED
FOR USE IN THE 812-POINT FFT PROCESSES.

SUBROUTINE GTSEA(NNP,SEA,SMSEA)

REAL SEA(2)

COMMON /SSM4N/ AMPLI,TIMP,SIGMA,BSLIN,XMLDA,XIDGE,XKURT

------ CONVERT SEA SIGMA TO # GATE INTERVALS

XNGTS = SIGMA/1.5625

ZERO WIDTH IS NOT ALLOWED

IF (XNGTS.LT.0.001) XNGTS = 0.001

WGTS = 0.

ESTABLISH CENTER AT 86TH GATE

NCTR = 86

SEA(NCTR) = 1.

SMSEA = 1.

K = NCTR - 1

XG = XMLDA/6.

INDX = 0.

------ FILL NON-ZERO ELEMENTS OF THE ARRAY

DO 20 J=1,K

INDX = XNGTS*WGTS

Z = -WGTS*WGTS/2.

IF (Z.LT.-6.G.) GO TO 10

A = EXP(Z)

Z = X6*WGTS*(WGTS*UGTS-3.)

------ REVERSED +,- SIGNS IN A1,A2 BELOW AFTER FINDING SIGN ERROR ON

10/15/80

------ 18/15/80

------ NOW FILL ZERO ELEMENTS OF THE ARRAY

DO 30 J=K,NP2

SEA(J) = 0.

END
APPENDIX E.

SOURCE LISTINGS FOR SUBROUTINES FFA AND FFS

0001 C DD0:CHAYNE:FFA.FR REVISED 07/03/80, APPROX. 1055 HOURS

0002 C

0003 C G.S.HAVNE APPL.SCI.AS30C'S. 01/01/80

0004 C

0005 C THE FOLLOWING IS ONE OF THE SET OF ROUTINES FOR FAST FOURIER

0006 C TRANSFORM OF REAL DATA SEQUENCE AS DESCRIBED IN SECTION 1.2 OF

0007 C "PROGRAMS FOR DIGITAL SIGNAL PROCESSING," ED. BY DIGITAL SIGNAL

0008 C PROCESSING COMMITTEE OF THE IEEE ASSP, PUBLISHED BY IEEE PRESS,

0009 C NY,1979. THESE ROUTINES ARE COLLECTIVELY THE "FFA-FFS PACKAGE"

0010 C WHICH INCLUDES: FFA, FFS, R2TR, R4TR, R8TR, R4SYN, R8SYN, ORD1,

0011 C AND ORD2.

0012 C

0013 C --------------------------------------------------

0014 C SUBROUTINE: FFA

0015 C FAST FOURIER ANALYSIS SUBROUTINE

0016 C --------------------------------------------------

0017 C SUBROUTINE FFA(B, NFFT)

0018 C

0019 C THIS SUBROUTINE REPLACES THE REAL VECTOR B(K), (K=1,2,...,N),

0020 C WITH ITS FINITE DISCRETE FOURIER TRANSFORM. THE DC TERM IS

0021 C RETURNED IN LOCATION B(1) WITH B(2) SET TO 0. THEREAFTER, THE

0022 C JTH HARMONIC IS RETURNED AS A COMPLEX NUMBER STORED AS

0023 C B(2*J+1) + I B(2*J+2). NOTE THAT THE N/2 HARMONIC IS RETURNED

0024 C IN B(N+1) WITH B(N+2) SET TO 0. HENCE, B MUST BE DIMENSIONED

0025 C TO SIZE N+2.

0026 C SUBROUTINE IS CALLED AS FFA (B,N) WHERE N=2**M AND B IS AN

0027 C M-TERM REAL ARRAY. A REAL-VALUED, RADIX 8 ALGORITHM IS USED

0028 C WITH IN-PLACE REORDERING AND THE TRIG FUNCTIONS ARE COMPUTED AS

0029 C NEEDED.

0030 C

0031 C DIMENSION B(2)

0032 C COMMON /COM/ PII, P7, P7TWO, C22, S22, PI2

0033 C

0034 C IW IS A MACHINE-DEPENDENT WRITE-DEVICE NUMBER

0035 C

0036 C 1W=12

0037 C

0038 C PII = 4.*ATAN(1.)

0039 C PI7 = PII/7.

0040 C P7 = 1./SIN(2.)

0041 C P7TWO = 2.*P7

0042 C C22 = COS(PII)

0043 C S22 = SIN(PII)

0044 C PI2 = 2.*PII

0045 C N = 1;

0046 C DO 10 I=1,16

0047 C M = I

0048 C N = N*2

0049 C IF (N.EQ.NFFT) GO TO 20

0050 10 CONTINUE

0051 20 CONTINUE

0052 WRITE (1W,9999)

0053 FORMAT (' NFFT NOT A POWER OF 2 FOR FFA')

0054 STOP

0055 9999 FORMAT (1X)

0056 NSPOM = M/3

0057 C

0058 C DO A RADIX 2 OR RADIX 4 ITERATION FIRST IF CHE IS REQUIRED
C

0050 C IF (M-NBPOW*3-1) 59, 49, 39
0051 59 NN = 4
0052 INT = N/NN
0053 CALL R4TR(INT, B(1), B(INT+1), B(2*INT+1), B(3*INT+1))
0054 GO TO 69
0055 49 NN = 2
0056 INT = N/NN
0057 CALL R2TR(INT, B(1), B(INT+1))
0058 GO TO 69
0059 39 NN = 1
0060 C
0061 C PERFORM RADIX 8 ITERATIONS
0062 C
0063 69 IF (NBPOW) 99, 99, 79
0064 79 DO 89 IT=1,NBPOW
0065 89 NN = NN*8
0066 INT = N/NN
0067 CALL R8TR(INT,NN,B(1),B(INT+1),B(2*INT+1),B(3*INT+1),
0068 1 B(4*INT+1),B(5*INT+1),B(6*INT+1),B(7*INT+1),B(1),
0069 2 B(INT+1),B(2*INT+1),B(3*INT+1),B(4*INT+1),B(5*INT+1),
0070 3 B(6*INT+1),B(7*INT+1))
0071 99 CONTINUE
0072 C
0073 C PERFORM IN-PLACE REORDERING
0074 C
0075 99 CALL ORD1(M, B)
0076 CALL ORD2(M, B)
0077 T = B(2)
0078 B(2) = Ø.
0079 B(NFFT+1) = T
0080 B(NFFT+2) = Ø.
0081 DO 100 I=4,NFFT,2
0082 B(I) = -3(I)
0083 100 CONTINUE
0084 RETURN
0085 END

ORIGINAL PAGE IS OF POOR QUALITY
THE FOLLOWING IS ONE OF THE SET OF ROUTINES FOR FAST FOURIER
TRANSFORM OF REAL DATA SEQUENCE AS DESCRIBED IN SECTION 1.2 OF
"PROGRAMS FOR DIGITAL SIGNAL PROCESSING," ED, BY DIGITAL SIGNAL
PROCESSING COMMITTEE OF THE IEEE ASSP, PUBLISHED BY IEEE PRESS,
NY, 1979. THESE ROUTINES COLLECTIVELY ARE "FFA-FFS PACKAGE"
WHICH INCLUDES: FFA, FFS, R2TR, R4TR, R8TR, R4SYN, R8SYN, ORD1,
AND ORD2.
SUBROUTINE FFS(B, NFFT)

DIMENSION B(2)
COMMON /CON1/ PI1, PI3, P7TWO, C22, S22, PI2

IV IS A MACHINE-DEPENDENT WRITE-DEVICE NUMBER

12 = 12

PI1 = 4.*ATAN(1.)
PI3 = PI1/6.
P7 = 1./SIN(PI3)
P7TWO = 2.*P7
C22 = COS(PI2)
S22 = SIN(PI2)

DO 10 I=1,16

N = N+2

M = 1

IF (N.EQ.NFFT) GO TO 20

DO 10 CONTINUE

WRITE (1W,9999)

FORMAT ("NFFT NOT A POWER OF 2 FOR FFS")

STOP

CONTINUE

B(2) = B(NFFT+1)

DO 20 I=1,NFFT

B(I) = B(I)/FLOAT(NFFT)

CONTINUE
CC55 DO 49 I=4,NFFT,2
CC56 B(I) = -B(I)
CC57 CONTINUE
CC58 K8POW = N/3
CC59 C REORDER THE INPUT FOURIER COEFFICIENTS
CC60 C CALL ORD2(N, B)
CC61 CALL ORD1(M, B)
CC62 C IF (N8POW.EQ.0) GO TO 60
CC63 C PERFORM THE RADIX 8 ITERATIONS
CC64 NN = N
CC65 DO 50 IT=1,NSPOW
CC66 INT = N/NN
CC67 CALL RBSYN(INT,NN,B,B(INT+1),B(2*INT+1),B(3*INT+1),
CC68 B(4*INT+1),B(5*INT+1),B(6*INT+1),B(7*INT+1),B(1),
CC69 B(2*INT+1),B(3*INT+1),B(4*INT+1),B(5*INT+1),
CC70 B(6*INT+1),B(7*INT+1))
CC71 NN = NN/8
CC72 CONTINUE
CC73 C DO A RADIX 2 OR RADIX 4 ITERATION IF ONE IS REQUIRED
CC74 INT = N/8
CC75 IF (M-N8POW*3-1) 90, 80, 70
CC76 CALL RBSYN(INT,B(INT+1),B(2*INT+1),B(3*INT+1))
CC77 GO TO 90
CC78 INT = N/2
CC79 CALL R2TR(INT, B(INT), B(INT+1))
CC80 RETURN
CC81 END