

CSDL Informal Technical Note No. 1

WATER LEVEL EVENT ANALYSIS: PROGRAM DOCUMENTATION

Silver Spring, Maryland
June 2002



U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Ocean Service
Coast Survey Development Laboratory

NOTICE

CSDL Informal Technical Notes present work in progress or summaries of results that are not appropriate to be published as either formal NOAA Office of Coast Survey Technical Reports or the less formal Technical Memoranda. Results are intended primarily for use within CSDL. Scientific review of the material is minimal, and CSDL makes no warranty as to its validity or completeness.

CSDL Informal Technical Note No. 1

WATER LEVEL EVENT ANALYSIS: PROGRAM DOCUMENTATION

Philip H. Richardson
Richard A. Schmalz, Jr.

Silver Spring, Maryland
June 2002



U.S. DEPARTMENT OF
COMMERCE

National Oceanic and
Atmospheric Administration

National Ocean Service,
Coast Survey Development
Laboratory

TABLE OF CONTENTS

LIST OF FIGURES	iv
LIST OF TABLES	iv
LIST OF COMPUTER PROGRAM LISTINGS	v
BASE MAP	vi
ABSTRACT	vii
1. INTRODUCTION	1
2. PROGRAM DESCRIPTIONS.	3
2.1. Reform_coops.f	3
2.2. Read_nowforc.f	8
2.3. Match.event.f	18
2.4. W1.sigma.pro	46
2.5. W1.multcur.pro	54
3. APRIL 2000 SAMPLE APPLICATION	64
4. OPERATIONAL USE AND ENHANCEMENTS	84
REFERENCES	85
APPENDIX A: JCL AND CONTROL FILES	86

LIST OF FIGURES

Galveston Bay Base Map	vi
Figure 3.1. Event Analysis Plots at Galveston Pleasure Pier for April 2000: Observed, Forecast, and Adjusted Forecast	78
Figure 3.2. Event Analysis Plots at Morgans Point for April 2000: Observed, Forecast, and Adjusted Forecast	79
Figure 3.3. Event Analysis Plots at Galveston Pleasure Pier for April 2000: Forecast, and Adjusted Forecast	80
Figure 3.4. Event Analysis Plots at Morgans Point for April 2000: Forecast, and Adjusted Forecast	81
Figure 3.5. Event Analysis Plots at Galveston Pleasure Pier for April 2000: Nowcast and Astronomical Tide Prediction	82
Figure 3.6. Event Analysis Plots at Morgans Point for April 2000: Nowcast and Astronomical Tide Prediction	83

LIST OF TABLES

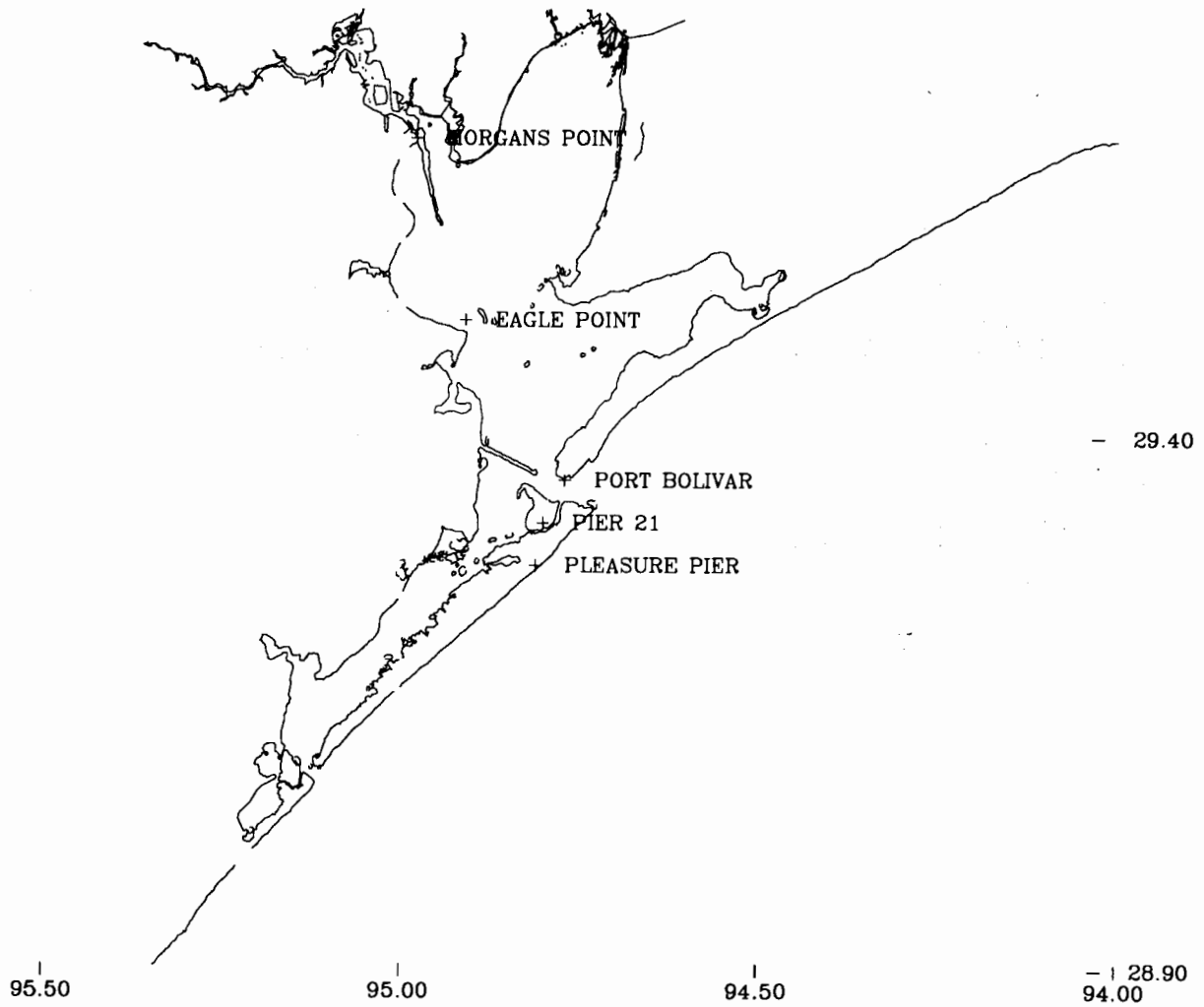
Table 3.1. Job Control, Source File, and Control File Inventory	65
Table 3.2a. Table2.out for Case One, unadjusted	66
Table 3.2b. Table2_high for Case One, unadjusted	68
Table 3.2c. Table2_low for Case One, unadjusted	70
Table 3.3a. Table2.out for Case Two, adjusted	72
Table 3.3b. Table2_high for Case Two, adjusted	74
Table 3.3c. Table2_low for Case Two, adjusted	76

LIST OF COMPUTER PROGRAM LISTINGS

Program Listing 2.1.	Reform_coops.f	4
Program Listing 2.2.	Read_nowforc.f	9
Program Listing 2.3.	Match.event.f	20
Program Listing 2.4.	Wl.sigma.pro	47
Program Listing 2.5.	Wl.multcur.pro	55

GALVESTON BAY BASE MAP

- 29.90



Galveston Bay area base map showing locations mentioned in this report.

ABSTRACT

The National Ocean Service (NOS), as part of its Houston/Galveston Physical Oceanographic Real Time System (PORTS), has developed an experimental nowcast/forecast system to predict water level and currents within Galveston Bay and the Houston Ship Channel. The experimental system provides a daily 36 hour forecast initiated from a continuous 24 hour nowcast as outlined by Schmalz and Richardson (1998). To further assess the water level nowcasts and forecasts, a set of five programs has been developed to study the ability of the nowcast/forecast system to simulate both high and low water events. These five programs are documented and a sample application is provided for the experimental nowcast/forecast system results during April 2000. JCL and control files for the April 2000 sample application are provided in Appendix A.

1. INTRODUCTION

A Physical Oceanographic Real Time System (PORTS) has been installed in Galveston Bay to provide the navigation community with real time water level and current information (Frey, 1991; Bethem and Frey, 1991). The development of the PORTS is in response to the results of a mini-project conducted by NOS in 1988, in which NOS current predictions within the Bay were found to be outside the range of NOS standards (Williams et al., 1990). The present PORTS consists of five tide gauges and two permanent Acoustic Doppler Current Profilers (ADCP). Conductivity/temperature measurement systems have been installed at several tide gauges.

To complement the PORTS, a nowcast/forecast system has been designed based on the National Ocean Service (NOS) Galveston Bay three-dimensional hydrodynamic model (Schmalz, 1996) and the National Weather Service's (NWS) Aviation atmospheric model (Schmalz and Richardson, 1996). To simulate currents within the Houston Ship Channel (HSC), a finer resolution three-dimensional HSC model has been developed. The Galveston Bay model is used to provide Bay-wide water level and near entrance current forecasts as well as to directly provide water levels, density, and turbulence quantities to the HSC model for use in a one-way coupling. The combined model set forms the initial hydrodynamic component of the nowcast/forecast system (Schmalz and Richardson, 1998).

The present nowcast/forecast system is used to provide experimental daily 24 hour nowcasts and 36 hour forecasts of water levels, currents, salinity and temperature at the locations of the PORTS instruments. Forecast inputs have been assessed by Richardson and Schmalz (1999). Here a set of five programs (reform_coops.f, read_nowforc.f, match.event.f, wl.sigma.pro, and wl.multcur.pro) has been developed to evaluate the nowcast/forecast model's ability to forecast water level "event" situations. An "event" occurs when the observed water level rises above a specified high level value, or falls below a specified low level value.

When forecasting water level events, there are three possible outcomes. A "success" occurs when the model successfully predicts an observed event. A "failure" occurs when the model fails to predict an observed event. A "false alarm" occurs when the model predicts an event not seen in the observed data. When the forecast model performs well, the number of successes should be high, and the number of failures and false alarms should be low. Each program is outlined below.

Reform_coops.f was written to reformat either hourly or six minute water level data, from the Center for Operational Oceanographic Products and Services (CO-OPS), into a standard format for analysis. CO-OPS water level data is acquired from their website at <http://co-ops.nos.noaa.gov>.

Read_nowforc.f was developed to read water level results from the NOS Galveston Bay and Houston Ship Channel models. The program reads water level results from the 00z files to produce both nowcast and forecast files for Pleasure Pier, Pier 21, Bolivar Roads, Morgans Point, and Eagle Point. The program can adjust the water level forecasts using a linear regression according to specified values for the bias and gain.

Match.event.f was originally written to perform a one year comparison of event conditions between the East Coast Ocean Forecast System (ECOFS) and the NWS Techniques Development Laboratory (TDL) storm surge model. In the comparisons, the high critical level value was one or two sigma levels above the observed long term mean, and the low critical level value was one or two sigma levels below the observed long term mean. Match.event.f was revised slightly to evaluate Houston/Galveston nowcast/forecast model results. The low water critical level is specified by the negative of a critical value read in from the control file. The high water critical level is determined by adding the critical value to the difference of the MHHW and MLLW for a particular station. Also, the previous table2 output was revised slightly. Table2 output includes the number of successes, failures, and false alarms by station. For each "success", table2 presents comparative model/observed event information including delta (model - observed) start time in hours, delta peak time in hours, and delta peak water level in meters. The revised table presents, in addition to the success number, the event number for that station. Including the event number makes the analysis more straight forward. A more comprehensive description of event characteristics is presented in table1. This table can be used for debugging purposes.

Wl.sigma.pro is written in the IDL programming language, and will plot the observed and model water levels along with two lines which depict the critical high and low water level values. Wl.sigma.pro generates one plot per page. An improved version of the program, wl.multcur.pro, was created to generate plots of observed versus nowcast and observed versus predicted on one page, then observed versus forecast and observed versus adjusted forecast on the second page.

Reform_coops.f, read_nowforc.f, and match.event.f are written in FORTRAN, and are run on the OPSEA system. Wl.sigma.pro and wl.multcur.pro are written in IDL and are run on the OPSEA system.

In Chapter 2, descriptions of each program are provided as well as program listings. In Chapter 3, a sample application is presented for April 2000, which corresponds to the first month of the twelve month period for formal evaluation using NOS (1999) techniques. In Chapter 4, recommendations for the operational use of these assessment programs are presented. In addition, possible future enhancements to each program are discussed. Complete JCL and control file listings are presented in Appendix A.

2. PROGRAM DESCRIPTIONS

2.1. Program Reform_coops.f

This program reads CO-OPS water level data and reformats the data. The listing for reform_coops.f is given in Program Listing 2.1. The raw water level filename is read in from the control file. Also read in from the control file is time_int, which designates the time interval to be either six minute or hourly. Line number 114 is the read statement. The program reads the station id number, the year, month, day, hour, and minute, then reads the water level. Statement 137 writes the water level value to output. Note that no conversion of the water level datum is made. Here it is assumed that the water level is given with respect to the MLLW datum.

```

1      c PROGRAM NAME : reform_coops.f
2      c
3      c PURPOSE : To reformat either hourly or 6 minute water
4      c             level data, from COOPS, into standard ECFS
5      c             format in order to be plotted or analyzed.
6      c             COOPS wl data is acquired from their website
7      c             at http://co-ops.nos.noaa.gov.
8      c
9      c AUTHOR : Phil Richardson
10     c
11     c LANGUAGE : FORTRAN 77
12     c
13     c SUBROUTINES : Calcjd.f
14     c
15     c LOCATION : On the OPSEA,
16     c             /usr/people/philr/galves/nowforc_eval/observed
17     c
18     c VERSION DATE : June 12, 2000
19
20     *****
21
22     character*3  desgtime
23     character*72 filenm
24     character*75 line(2)
25     character*15 fileout
26
27
28     c Read from Control file :
29     c
30     c     filenm - raw observed water level data
31     c     fileout - output file
32     c     starttime - start time for observed data
33     c     endtime - end time for observed data
34     c     time_int - time interval (hourly or 6 minute)
35
36     read(5,'(a72)')filenm
37     read(5,'(a15)')fileout
38     read(5,*)time_int
39     read(5,31)desgtime
40     if(desgtime.eq.'yes')then
41         read(5,*)starttime
42         read(5,*)endtime
43     endif
44
45
46     31 format(1x,a3)
47
48     *****
49
50     c     Open raw water level data file, open output file
51
52
53     lun = 8
54     n_time = 0
55
56     open(unit=lun,file=filenm,form='formatted',

```

Program Listing 2.1. Reform_coops.f

```

57         *      status='old')
58
59         open(11,file=fileout,form='formatted')
60
61         ****
62
63     c      Read opening three lines from raw data file (the header)
64
65         do l=1,3
66             read(lun,40)line(l)
67             write(6,40)line(l)
68         enddo
69
70     !-----
71
72     c      Position raw data file to correct start time
73
74         if(desgtime.eq.'yes')then
75     55 continue
76         read(lun,45,end=100)istaid,iyr,imon,iday,ihr,min,wl
77         call calcjd(jday,imon,iday,iyr)
78         rjday = float(jday)
79         rtime = rjday + float(ihr)/24.0 + float(min)/1440.0
80         if(rtime.lt.starttime)goto 55
81         write(6,*)rtime,starttime
82         backspace lun
83         endif
84
85
86
87     50 continue
88
89         if(time_int.lt.10.0)then
90             if(n_time.ge.1)then
91                 do i=1,9
92                     read(lun,40,end=100)lines
93                 enddo
94             endif
95         endif
96
97
98
99     40 format(a75)
100
101     !-----
102
103     c      Read statement
104     c      Variables read from raw data file :
105     c
106     c      istaid - station id number
107     c      iyr - year
108     c      imon - month
109     c      iday - day
110     c      ihr - hour
111     c      min - minute
112     c      wl - water level value

```

Program Listing 2.1. Reform_coops.f (continued)

```

113
114         read(lun,45,end=100)istaid,iyr,imon,iday,ihr,min,wl
115
116
117         45 format(i7,1x,i4,1x,i2,1x,i2,1x,i2,1x,i2,f8.3)
118
119         !-----
120
121         if(min.ne.0)then
122             write(6,47)istaid,iyr,imon,iday,ihr,min
123             stop
124         endif
125         write(6,45)istaid,iyr,imon,iday,ihr,min,wl
126
127
128         call calcjd(jday,imon,iday,iyr)
129
130         rjday = float(jday)
131         rtime = rjday + float(ihr)/24.0 + float(min)/1440.0
132         write(6,*)rtime
133         if(desgtime.eq.'yes')then
134             if(rtime.gt.endtime)goto 100
135         endif
136
137         write(11,46)rtime,wl
138
139         n_time = n_time + 1
140         goto 50
141
142     100 continue
143
144
145         46 format(1x,2f9.4)
146         47 format(1x,'Program Terminated',/,1x,i8,5i5)
147
148         *****
149
150         stop
151         end

```

Program Listing 2.1. Reform_coops.f (continued)


```

1      SUBROUTINE CALCJD (JDY,ICM,ICD,IYR)
2
3
4      C      This subroutine will convert the calender month (ICM) and
5      C      calender day (ICD) to the corresponding Julian day (JDY).
6      C
7      C      Input arguments -
8      C
9      C          ICM - Calender Month
10     C          ICD - Calender Day
11     C          IYR - Year
12     C
13     C      Output Argument -
14     C
15     C          JDY - Julian Day
16
17     DIMENSION JDAY(12), JDAYL(12)
18     DATA JDAY /0,31,59,90,120,151,181,212,243,273,304,334/
19     DATA JDAYL /0,31,60,91,121,152,182,213,244,274,305,335/
20
21     IF(MOD(IYR,4) .EQ. 0 .AND. MOD(IYR,100) .NE. 0 .OR.
22     *   MOD(IYR,400) .EQ. 0) THEN
23         RJD = FLOAT(JDAYL(ICM)) + FLOAT(ICD)
24     ELSE
25         RJD = FLOAT(JDAY(ICM)) + FLOAT(ICD)
26     END IF
27
28     JDY = INT(RJD)
29     RETURN
30     END

```

Program Listing 2.1. Reform_coops.f (continued)

2.2 Program Read_nowforc.f

The computer listing for Read_nowforc.f is given in Program Listing 2.2. For each of the five stations included in our analysis, the program first reads nowcast and forecast output filenames, then reads bias and gain (by station), from the control file. Line 86 begins the loop in which the daily 00z files are read. Nfiledays is the number of daily files to be read (30 for April). The Galveston Bay model (GBM) 00z files are stored in filename_gbm(nf), while the Houston Ship Channel Model (HSCM) 00z files are stored in filename_hsc(nf). Nf is equal to one through nfiledays. Also read from the control file is nhr_skip. Nhr_skip designates how many hours of forecast water level data to skip. If we want the first 24 hours of forecast data, nhr_skip is set equal to 0. If we want forecast hours 13 through 36, nhr_skip is set equal to 12.

Read_nowforc.f accounts for days in which a 00z file is missing in either the GBM or the HSCM. Read from the control file are nmiss_gbm and nmiss_hsc, the number of missing 00z files for each. The program then reads the particular days that 00z files are missing for both GBM and HSCM. For periods of time of missing data, the program substitutes a null value.

The 500 loop, which begins on line 148, loops through the days of the month from nf equals one through nfiledays. Both the GBM and the HSCM 00z files are opened. The program first reads through 24 hours of nowcast data. Lines 186 through 191 read the nowcast water level values from the GBM data file for Pleasure Pier, Pier 21, and Bolivar Roads. An offset, which is particular to each station, is added to each water level value in the 105 loop. This offset places the water level on the MLLW datum and represents the difference between MTL and MLLW. Lines 233 through 236 read water level values from the HSCM data file for Morgans Point and Eagle Point. The appropriate offset is applied in the 106 loop.

After the designated number of forecast hours have been skipped (nhr_skip), the reading of 24 hours of forecast results is done in the 200 loop. The 200 loop begins on line 309, where nhr ranges from 1 through 24. As with the nowcast results, water level values are read for Pleasure Pier, Pier 21, and Bolivar Roads from the GBM output file, and for Morgans Point and Eagle Point from the HSCM output file. The calls to subroutine modelfit, which occur in the 205 loop for GBM data and in the 206 loop for HSCM results, adjust the water level data according to the specified values of the bias and gain. After the adjustment, the same offsets added to the nowcast values are added to the forecast values.

At the end of the 500 loop, the GBM 00z file and the HSCM 00z file are closed for that day.

```

1      c PROGRAM NAME : read_nowforc.f
2      c
3      c PURPOSE : To read water level data from 00z (GBM and HSC)
4      c             files to produce nowcast and forecast water
5      c             level files for Pleasure Pier, Pier 21,
6      c             Bolivar Roads, Morgan's Point, and Eagle Point.
7      c             The revised version of the program will adjust
8      c             the water level values based upon the values
9      c             read in for the bias and gain.
10     c
11     c LOCATION : on OPSEA,
12     c             /usr/people/philr/galves/nowforc_eval/
13     c
14     c SUBROUTINES : modelfit
15     c
16     c AUTHOR : Philip Richardson
17     c
18     c VERSION DATE : July 6, 2000
19
20     *****
21
22     parameter(numdays=31,nstations=5)
23
24     character*70 line
25     character*55 filenm_gbm(numdays),filenm_hsc(numdays)
26     character*22 header
27     character*15 fileout(nstations)
28     character*16 fileforc(nstations)
29
30     dimension lunout(nstations),lnforc(nstations),
31     *          ndaymiss_gbm(numdays),ndaymiss_hsc(numdays),
32     *          wlevel(nstations),offset(nstations),
33     *          wlevel_prm(nstations),bias(nstations),
34     *          gain(nstations)
35
36     *****
37
38     c Read from control file :
39
40     c     startday - Julian date at beginning of month
41     c     lunout - logical unit numbers for nowcast output
42     c             files by station
43     c     lnforc - logical unit numbers for forecast output
44     c             files by station
45     c     nhr_skip - number of hours of forecast data to skip
46     c     bias - water level offset
47     c     gain - slope, factor to multiply wl by
48     c     nfiledays - number of daily 00z files to read/open
49     c     nmiss_gbm - number of missing daily files (GBM)
50     c     nmiss_hsc - number of missing daily files (HSC)
51     c     filenm_gbm - GBM (Galveston Bay Model) wl data
52     c     filenm_hsc - HSC (Houston Ship Channel) wl data
53
54
55     read(5,*)startday
56

```

Program Listing 2.2. Read_nowforc.f

```

57     do ns=1,nstations
58         read(5,*)lunout(ns)
59         read(5,92)fileout(ns)
60         read(5,*)lnforc(ns)
61         read(5,93)fileforc(ns)
62         read(5,*)bias(ns),gain(ns)
63     enddo
64
65     read(5,*)nhr_skip
66     read(5,1001)header
67
68     read(5,*)nfiledays
69     read(5,*)nmiss_gbm
70     write(6,*)nmiss_gbm
71     do nm=1,nmiss_gbm
72         read(5,*)ndaymiss_gbm(nm)
73         write(6,*)ndaymiss_gbm(nm)
74     enddo
75     read(5,*)nmiss_hsc
76     write(6,*)nmiss_hsc
77     do nm=1,nmiss_hsc
78         read(5,*)ndaymiss_hsc(nm)
79         write(6,*)ndaymiss_hsc(nm)
80     enddo
81
82
83     nmgbm = 1
84     nmhsc = 1
85
86     do nf=1,nfiledays
87         if(nf.ne.ndaymiss_gbm(nmgbm))then
88             read(5,91)filenm_gbm(nf)
89             write(6,91)filenm_gbm(nf)
90         else
91             nmgbm = nmgbm + 1
92         endif
93         if(nf.ne.ndaymiss_hsc(nmhsc))then
94             read(5,91)filenm_hsc(nf)
95             write(6,91)filenm_hsc(nf)
96         else
97             nmhsc = nmhsc + 1
98         endif
99     enddo
100
101
102     91 format(a55)
103     92 format(a15)
104     93 format(a16)
105
106     *****
107
108     c Open nowcast and forecast output files
109
110
111     do ns=1,nstations
112         open(lunout(ns),file=fileout(ns),form='formatted')

```

Program Listing 2.2. Read_nowforc.f (continued)

```

113         open(lnforc(ns),file=fileforc(ns),form='formatted')
114         write(lnforc(ns),1001)header
115         write(lnforc(ns),94)bias(ns),gain(ns)
116     enddo
117
118
119     94 format(1x,'bias = ',f6.3,', gain = ',f5.2)
120
121     *****
122
123     c Read nowcast wl data. 500 loop is 1 through nfiledays
124     c (number of days in the month).
125     c
126     c
127     c Variables :
128     c
129     c     wlnull - null water level value for missing nowcast
130     c             data
131     c     rtime - calculated time (by day), not including
132     c             hour
133     c     wltime - calculated time (Julian date) including
134     c             hour
135     c     gbm_time - time read from Bay model wl data file
136     c     wl_pleas - water level value at Pleasure Pier
137     c     wl_pie21 - water level value at Pier 21
138     c     wl_bolvr - water level value at Bolivar Roads
139
140
141     wlnull = 99.99
142     nmgbm = 1
143     nmhsc = 1
144
145     rday = startday - 2.0 + 0.7083
146
147
148     do 500 nf=1,nfiledays
149
150         rtime = rday + float(nf-1)
151     c     Open daily wl data files, both gbm and HSC. Read
152     c     header information.
153
154         lungbm = 8
155         lunhsc = 9
156         if(nf.ne.ndaymiss_gbm(nmgbm))then
157             open(lungbm,file=filenm_gbm(nf),
158                 * form='formatted',status='old')
159             do l=1,6
160                 read(lungbm,101)line
161                 write(6,101)line
162             enddo
163         endif
164         if(nf.ne.ndaymiss_hsc(nmhsc))then
165             open(lunhsc,file=filenm_hsc(nf),
166                 * form='formatted',status='old')
167             do l=1,7
168                 read(lunhsc,101)line

```

Program Listing 2.2. Read_nowforc.f (continued)

```

169         write(6,101)line
170     enddo
171 endif
172
173
174 c     Read first 24 hours of wl data for nowcast file.
175 c     The 100 loop is the hour loop for reading nowcast
176 c     wl data.
177
178     do 100 nhr=1,24
179         wlttime = rtime + float(nhr)/24.0
180 c     Read time (Julian date) from both model output
181 c     files.
182         if(nf.ne.ndaymiss_gbm(nmgbm))then
183             read(lungbm,*)gbm_time
184             write(6,115)gbm_time
185
186             read(lungbm,113)wl_pleas
187             write(6,115)wl_pleas
188             read(lungbm,112)wl_pie21
189             write(6,115)wl_pie21
190             read(lungbm,113)wl_bolvr
191             write(6,115)wl_bolvr
192
193             wlevel(1) = wl_pleas
194             wlevel(2) = wl_pie21
195             wlevel(3) = wl_bolvr
196             offset(1) = 0.362
197             offset(2) = 0.253
198             offset(3) = 0.253
199
200             do 105 ns=1,3
201                 wlevel(ns) = wlevel(ns) + offset(ns)
202                 write(lunout(ns),121)gbm_time,wlevel(ns)
203             105 continue
204
205
206             do nt=1,9
207                 read(lungbm,*)gbm_time
208                 write(6,115)gbm_time
209                 do lwl=1,3
210                     read(lungbm,101)line
211                     write(6,101)line
212                 enddo
213             enddo
214         else
215             write(lunout(1),121)wlttime,wlnull
216             write(lunout(2),121)wlttime,wlnull
217             write(lunout(3),121)wlttime,wlnull
218         endif
219
220
221 c Read from HSC file
222 c
223 c Variables :
224 c

```

Program Listing 2.2. Read_nowforc.f (continued)

```

225 c hsc_time - time read from Houston ship channel
226 c wl data file
227 c wl_morgn - water level value at Morgan's Point
228 c wl_eagle - water level value at Eagle Point
229
230 if(nf.ne.ndaymiss_hsc(nmhsc))then
231 read(lunhsc,*)hsc_time
232 write(6,115)hsc_time
233 read(lunhsc,114)wl_morgn
234 write(6,115)wl_morgn
235 read(lunhsc,112)wl_eagle
236 write(6,115)wl_eagle
237 do l=1,3
238 read(lunhsc,101)line
239 enddo
240
241 wlevel(4) = wl_morgn
242 wlevel(5) = wl_eagle
243 offset(4) = 0.198
244 offset(5) = 0.174
245
246 do 106 ns=4,5
247 wlevel(ns) = wlevel(ns) + offset(ns)
248 write(lunout(ns),121)hsc_time,wlevel(ns)
249 106 continue
250
251
252 do nt=1,9
253 read(lunhsc,*)hsc_time
254 write(6,115)hsc_time
255 do lwl=1,5
256 read(lunhsc,101)line
257 write(6,101)line
258 enddo
259 enddo
260 else
261 write(lunout(4),121)wltime,wlnull
262 write(lunout(5),121)wltime,wlnull
263 endif
264
265 100 continue
266
267
268 c Skip over first nhr_skip hours (12) to prepare for
269 c reading forecast data
270
271 if(nf.ne.ndaymiss_gbm(nmgbm))then
272 do 110 nhr=1,nhr_skip
273 do nt=1,10
274 read(lungbm,*)gbm_time
275 do lwl=1,3
276 read(lungbm,101)line
277 enddo
278 enddo
279 110 continue
280 endif

```

Program Listing 2.2. Read_nowforc.f (continued)

```

281
282
283         if(nf.ne.ndaymiss_hsc(nmhsc))then
284         do 120 nhr=1,nhr_skip
285             do nt=1,10
286                 read(lunhsc,*)hsc_time
287                 do lwl=1,5
288                     read(lunhsc,101)line
289                 enddo
290             enddo
291         120     continue
292         endif
293
294         !-----
295
296         c Read forecast water level data.  Read GBM data first.
297         c Call to subroutine modelfit adjusts water level values
298         c based upon values of bias and gain.
299         c
300         c Variables :
301         c
302         c         wlevel - array which stores wl values for all
303         c                 five stations
304         c         wlevel_prm - adjusted wl values
305
306
307         rtime = rtime + 1.0 + float(nhr_skip)/24.0
308
309         do 200 nhr=1,24
310             wlttime = rtime + float(nhr)/24.0
311             if(nf.ne.ndaymiss_gbm(nmgbm))then
312                 read(lungbm,*)gbm_time
313                 write(6,115)gbm_time
314                 read(lungbm,113)wl_pleas
315                 write(6,115)wl_pleas
316                 read(lungbm,112)wl_pie21
317                 write(6,115)wl_pie21
318                 read(lungbm,113)wl_bolvr
319                 write(6,115)wl_bolvr
320
321                 wlevel(1) = wl_pleas
322                 wlevel(2) = wl_pie21
323                 wlevel(3) = wl_bolvr
324
325                 do 205 ns=1,3
326                     call modelfit(wlevel(ns),bias(ns),gain(ns),
327             *                 wlevel_prm(ns))
328                     wlevel_prm(ns) = wlevel_prm(ns) + offset(ns)
329                     write(lnforc(ns),121)gbm_time,wlevel_prm(ns)
330         205     continue
331
332
333             do nt=1,9
334                 read(lungbm,*)gbm_time
335                 write(6,115)gbm_time
336                 do l=1,3

```

Program Listing 2.2. Read_nowforc.f (continued)


```

337             read(lungbm,101)line
338             write(6,101)line
339         enddo
340     enddo
341     else
342         write(lnforc(1),121)wlttime,wlnull
343         write(lnforc(2),121)wlttime,wlnull
344         write(lnforc(3),121)wlttime,wlnull
345         if(nhr.eq.24)then
346             nmgbm = nmgbm + 1
347         endif
348     endif
349
350
351
352 c Read HSC water level data
353
354     if(nf.ne.ndaymiss_hsc(nmhsc))then
355         read(lunhsc,*)hsc_time
356         write(6,115)hsc_time
357
358         read(lunhsc,114)wl_morgn
359         write(6,115)wl_morgn
360         read(lunhsc,112)wl_eagle
361         write(6,115)wl_eagle
362
363         wlevel(4) = wl_morgn
364         wlevel(5) = wl_eagle
365
366         do 206 ns=4,5
367             call modelfit(wlevel(ns),bias(ns),gain(ns),
368                 *           wlevel_prm(ns))
369             wlevel_prm(ns) = wlevel_prm(ns) + offset(ns)
370             write(lnforc(ns),121)hsc_time,wlevel_prm(ns)
371         206 continue
372
373
374         do l=1,3
375             read(lunhsc,101)line
376         enddo
377
378         do nt=1,9
379             read(lunhsc,*)hsc_time
380             write(6,115)hsc_time
381             do lwl=1,5
382                 read(lunhsc,101)line
383                 write(6,101)line
384             enddo
385         enddo
386     else
387         write(lnforc(4),121)wlttime,wlnull
388         write(lnforc(5),121)wlttime,wlnull
389         if(nhr.eq.24)then
390             nmhsc = nmhsc +1
391         endif
392     endif

```

Program Listing 2.2. Read_nowforc.f (continued)

```
393
394     200 continue
395
396
397         close (lungbm)
398         close (lunhsc)
399     500 continue
400
401 *****
402
403     101 format(a70)
404     112 format(60x,f10.4)
405     113 format(70x,f10.4)
406     114 format(40x,f10.4)
407     115 format(1x,f9.4)
408     121 format(1x,2f9.4)
409
410     1001 format(a22)
411 *****
412
413
414         stop
415         end
```

Program Listing 2.2. Read_nowforc.f (continued)

```

1      subroutine modelfit(wlevel,b,rm,yprime)
2
3      c      Purpose : Using values for bias and gain, to
4      c      fit the model forecast results to more closely
5      c      follow the observed water level data.
6      c
7      c      Author : Phil Richardson
8      c
9      c      Date : October 4, 2000
10
11     *****
12
13     c      Input Arguments :
14     c
15     c          wlevel - value of water level read from Dick's
16     c                   nowcast/forecast (model) output
17     c                   b - bias, water level offset
18     c                   rm - gain
19     c
20     c      Output Arguments :
21     c
22     c          yprime - adjusted water level value
23
24
25     yprime = rm * wlevel + b
26
27     return
28     end

```

Program Listing 2.2. Read_nowforc.f (continued)

2.3. Program Match.event.f

The listing for Program Match.event.f is given in Program Listing 2.3. Following the parameter and dimension statements and after the character variables are declared, match.event.f will read necessary information from the control file. Variables read from the control file include idebug, monyr, headr1, startjd, and endjd. Idebug can be set from 0 to 5 for various debug output. Monyr is the month and year for output files. Headr1 is the header for output files. Startjd and endjd are the start and end times in Julian days (elapsed days since the beginning of the calendar year) of the analysis. The next variable read is crlevel. Crlevel determines the lower and upper critical water level values, as described below. Next is nsta, the number of stations included in the analysis. For each station, a station name is read, a range offset is read, then the observed and model filenames are read. Also read from the control file is option. Option determines whether the data are treated as forecast, nowcast, or astronomic tide prediction. For the astronomic tide prediction data, note lines 300 - 301 for the appropriate read statement. Also note line 268 for the format statement.

The range offset is defined as MHHW - MLLW and is particular to each station. The high water critical value is determined by adding crlevel to the range offset. The low water critical value is defined as $0 - \text{crlevel}$. All water level data are referenced to MLLW. In effect, a water level value above $\text{MHHW} + \text{crlevel}$ is a high water extreme value and part of a high water event, and a water level value below $\text{MLLW} - \text{crlevel}$ is a low water extreme value and part of a low water event.

The 100 loop is the station loop, where ns is equal to 1 through nsta (nsta being the number of stations). First, the observed and model water level files are opened for each station. The observed water level file is read, the number of data points counted, and the mean observed water level is calculated.

After the observed data file has been repositioned to the start time, and the model output file has been positioned, the model output and the observed data are read simultaneously. If either the model or observed data point are above the high water critical value or below the low water critical value, then the counter (nc2sig) is incremented and the water level values, model and observed, are stored in wlm and wlo, respectively. The corresponding times are stored in time_m and time_o, although the model time and the observed time are essentially equal. Each point is flagged as to whether it is part of a high water event, iflag = 1, or a low water event, iflag = 0. By considering model points and observed points simultaneously, all information for further analysis has already been gathered, and the corresponding model and observed data points are already paired. If either the model data point or the observed data point is not present, that data point will not be considered. In other words, any period of time in which there is a data gap will not be considered in the analysis. This method is more efficient than reading through one data stream, looking for events, then looking through the second data stream, and looking for events.

The 190 loop, which begins on line 395, loops through all extreme points, from 1 through nc2sig. All extreme points are grouped into events. There are two conditions which determine the grouping of events. All points grouped in one event are continuous in time. In other words, the time interval

separating each point in an event is one hour. An extreme point in which the time interval between it and the preceding extreme point is greater than one hour, will become the first data point of the next event. Points grouped in one event do not alternate between high and low water. For instance, if two extreme points are one hour apart, the first is a high water extreme point and the second is a low water extreme point, these points will be grouped in two separate events. The statements in lines 417-418 checks for the two conditions. The number of points in the event, `nevent`, is incremented. Events are then identified as being either high water events or low water events. Line 421 sets `iflg = 1` for a high water event, and line 427 sets `iflg = 0` for a low water event. For a high water event, `nevent_high` is incremented. For a low water event, `nevent_low` is incremented.

The 200 loop, which begins on line 490, loops through events from 1 through `nevent`, where `nevent` is the number of events for that station. The 210 loop, loops through the points of each event, from 1 through `nh(ns,np)`, where `ns` is the station number and `np` is the event number. The 210 loop determines the peak water level value and time for each high water event, and the low water level value and time for each trough. Also, the initial extreme point of each event is identified.

Finally, the 440 loop (line 745) loops through stations number 1 through `nsta`. The 450 loop (line 772) where `np = 1` through the number of events for that station, categorizes each event as to "success", "failure", or "false alarm". The program recognizes a "success" by `tm_pkvalWr` (the peak value of a particular event) being less than 90000.0 for both model and observed. In the case of a failure, `tm_pkvalWr` (model) will have been set to the null value (99999.9999). In the case of a false alarm, `tm_pkvalWr` (obs) will have been set to the null value.

The output is written to the monthly table files, `table1.out` and `table2.out`. `Table2.out` includes information on each event, whether the outcome of the forecast is a "success", "failure", or "false alarm". The information for each "success" includes the success number, the event number, the difference in start time between model and observed, and the difference in the duration of the event between model and observed. For failure and false alarm, the difference in start time and the difference in the duration of the event are not applicable. `Dpeak` time is the time difference of the model peak water level and the observed peak water level, in hours. Also included are the time of the observed peak, the peak model and observed water levels, and `dwl`, the absolute value of the difference between the model and the observed peak water level in meters. `Dwl` is added for each event. The mean `dwl` for each outcome (success, failure, false alarm) is calculated and presented in the table. Also included, in `table2`, is a summary, by station, of the total number of successes, failures, and false alarms for that month. If the forecast is working well, the number of successes should be high, and the number failures and false alarms should be low. `Table1.out` provides more detailed information on each event, and can be used as a debugging tool. The program generates two other tables. `Table_high` provides information on the high water events, while `table_low` provides information on the low water events.

```

1      c      Program Name : Match.event.f
2      c
3      c      Author : Phil Richardson
4      c
5      c      Version Date : June 26, 2000
6      c
7      c      Purpose : To search for wl values < -crlevel or >
8      c      (MHHW - MLLW) + crlevel, where crlevel is a critical
9      c      value read in. The program will read through model
10     c      and observed data simultaneously. Match.event.f
11     c      originally written to assist Eddie Shih with
12     c      his COFS vs. TDL analysis. Later, it was revised
13     c      to evaluate Houston/Galveston forecast and nowcast
14     c      wl data. In particular, the program will evaluate the
15     c      system with regard to event situations.
16     c      This is the final version of the program. It knows
17     c      to consider closely grouped spikes, such as un-
18     c      filtered cfs3.0, as one continuous spike. Also,
19     c      table with success, failure, and false alarm
20     c      statistics by station for each month.
21     c
22     c      Subroutines called : calcjd, timehi, jdgreg, wr_header
23     c
24     c      Revision : July 18, 2000
25     c      Column added to table2 for event number.
26     c
27     c      Location : OPSEA -
28     c      /usr/people/philr/galves/NF_eval/wlevel/sa.nowforc
29

```

```

30     *****

```

```

31
32     parameter(npts=745,npeaks=50,nstat=12)
33
34     character*1  formfd
35     character*8  option
36     character*10 filetab,filetab2,file23
37     character*13 stanam(nstat),monyr
38     character*80 file22
39     character*25 headr2
40     character*28 headr3(nstat)
41     character*60 headr1,file2sig
42     character*69 fileobs,filemod,filesumm
43
44     dimension iflg(npeaks),iflag(npts),i_flag(3,npeaks)
45     dimension nh(nstat,npeaks),npt_ospk(nstat,npeaks),
46     *         npt_modpk(nstat,npeaks),iflag_sta(nstat,npeaks)
47     dimension fileobs(nstat),filemod(nstat),
48     *         nevent_sta(nstat),neventl(nstat),
49     *         crlevelL(nstat),crlevelH(nstat)
50     dimension time_o(npts),time_m(npts),wlo(npts).
51     *         wlm(npts)
52     dimension time_peako(nstat,npeaks,npts),
53     *         time_peakm(nstat,npeaks,npts),
54     *         wlev_peako(nstat,npeaks,npts),
55     *         wlev_peakm(nstat,npeaks,npts)
56     dimension tm_pkval(nstat,2,npeaks),

```

Program Listing 2.3. Match.event.f

```

57      *          tm_pkvalWr(nstat,2,npeaks),
58      *          wl_pkval(nstat,2,npeaks),
59      *          wl_pkvalWr(nstat,2,npeaks),
60      *          cald_pkval(nstat,2,npeaks)
61      dimension wlmean(nstat)
62      dimension tm_1stval(nstat,2,npeaks)
63      dimension range_offset(nstat)
64      dimension idel_strtm(3,npeaks),idel_peaktm(3,npeaks),
65      *          ievent(3,npeaks),idel_duration(npeaks),
66      *          obs_wl(3,npeaks),delta_wl(3,npeaks),
67      *          forc_wl(3,npeaks),tm_obspeak(3,npeaks)
68
69      common/headrs/headr1,headr2,monyr,option,startjd,
70      *          endjd,crlevel
71
72      *****
73
74      c  Read from input :
75      c
76      c    idebug - debug switch
77      c    idebug = 1, write input files to 6
78      c    idebug = 2, events, low and high
79      c           = 3, points above critical water level value,
80      c           below critical water level value
81      c           = 4, check for tdiff (time difference)
82      c           = 5, low and high critical values
83      c    monyr - month and year
84      c    headr - header for output files
85      c    startjd - starting time
86      c    endjd - ending time
87      c    crlevel - critical value for wl
88      c    option - forecast, nowcast, or astronomic
89      c    nsta - number of stations
90      c    stanam - station name
91      c    fileobs - file containing observed water level data
92      c    filemod - file containing model water level data
93
94      read(5,*)idebug
95      read(5,1039)monyr
96      read(5,31)headr1
97      read(5,*)startjd
98      write(6,32)startjd
99      read(5,*)endjd
100     write(6,33)endjd
101
102     read(5,*)crlevel
103
104     read(5,38)option
105     read(5,'(a80)')file22
106     read(5,*)nsta
107     do ls=1,nsta
108         read(5,1039)stanam(ls)
109         if(idebug.eq.1)write(6,1040)stanam(ls)
110         read(5,*)range_offset(ls)
111         write(6,35)range_offset(ls)
112         read(5,34)fileobs(ls)

```

Program Listing 2.3. Match.event.f (continued)

```

113         if(idebug.eq.1)write(6,34)fileobs(1s)
114         read(5,34)filemod(1s)
115         if(idebug.eq.1)write(6,34)filemod(1s)
116     enddo
117
118
119     dayspl = endjd - startjd
120     c      initialization
121     formfd = CHAR(12)
122     small = 0.01
123     tlhour = 0.0417
124
125
126     c      Set time check interval for peaks
127
128     nh_ch = 1
129     peak_intrvl = float(nh_ch) * tlhour + .002
130     write(6,36)peak_intrvl
131     big_value = 99999.9999
132     write(6,37)
133
134
135     31 format(a60)
136     32 format(/,'Julian start time from control file',
137         *      f8.2)
138     33 format('Julian stop time from control file ',f8.2,/)
139     34 format(a69)
140     35 format('range offset : MHHW - MLLW = ',f7.3)
141     36 format(/,'Peak time interval = ',f8.4)
142     37 format(//)
143     38 format(a8)
144
145     *****
146
147     c      Open output files
148
149     c      Filenames :
150     c
151     c      filetab - table of event information
152     c      filetab2 - table of event information;
153     c                  includes success, failure, false alarm
154     c                  summary
155     c      file22 - same as filetab2, high water events
156     c      file23 - same as filetab2, low water events
157
158
159     filetab = 'table.out'
160     filetab2 = 'table2.out'
161     file23 = 'table_low'
162
163     open(18,file=filetab,form='formatted')
164     open(21,file=filetab2,form='formatted')
165     open(22,file=file22,form='formatted')
166     open(23,file=file23,form='formatted')
167
168     if(idebug.eq.5)then

```

Program Listing 2.3. Match.event.f (continued)


```

169         file2sig = 'critvals'
170         open(19,file=file2sig,form='formatted')
171         write(19,42)
172     endif
173
174
175     42 format(//,' station          crlevel_low    crlevel_high',
176            *          ' range_offset')
177
178     *****
179
180     c      Begin station loop (100).  Open observed and
181     c      model water level files.
182
183
184         lunobs = 7
185         lunmod = 8
186
187         do 100 ns=1,nsta
188             write(6,1040)stanam(ns)
189             open(lunobs,file=fileobs(ns),form='formatted',
190            *          status='old')
191             open(lunmod,file=filemod(ns),form='formatted',
192            *          status='old')
193
194     !-----
195
196     c      Read observed water level data, count number
197     c      of data points, calculate mean water level.
198     c
199     c      Variables :
200     c          nc - number of data points counter
201     c          wlsun - sum of water level values
202     c          wlmean - mean water level (by station)
203
204
205         if(option.eq.'forecast')then
206             read(lunmod,102)headr2
207             read(lunmod,1003)headr3(ns)
208         endif
209
210         nc = 0
211         wlsun = 0.0
212
213     45     continue
214         read(lunobs,*,end=105)t,wl
215         if(t.lt.startjd)goto 45
216         if(t.gt.endjd)goto 105
217         nc = nc + 1
218         wlsun = wlsun + wl
219         goto 45
220     105     continue
221
222
223     c      Write statistics to out file
224         write(6,111)nc

```

Program Listing 2.3. Match.event.f (continued)

```

225         wlmean(ns) = wlsum/float(nc)
226         write(6,112)wlmean(ns)
227
228
229         102 format(a25)
230         111 format(1x,i5,' data points in observed file')
231         112 format(1x,'mean water level (observed)',f9.4)
232
233         !-----
234
235         c           Reposition obs file to start time
236
237         rewind lunobs
238
239         610        continue
240         read(lunobs,*,end=615)t,wl
241         if(t.lt.startjd)goto 610
242         615        continue
243         write(6,121)t
244         backspace lunobs
245
246
247         c           Position mod file to start time
248
249         620        continue
250         if(option.eq.'forecast'.or.option.eq.'nowcast*')then
251             read(lunmod,*,end=625)t,wl
252         endif
253         if(option.eq.'astronom')then
254             read(lunmod,125,end=625)istation_id,iyear,
255             *      imonth,iday,ihr,imin
256             call calcjd(jday,imonth,iday,iyear)
257             rday = float(jday) + float(ihr)/24.0
258             t = rday
259         endif
260         if(t.lt.startjd)goto 620
261         625        continue
262         write(6,122)t
263         backspace lunmod
264
265
266         121 format('Start time (observed wl data)',f7.2)
267         122 format('Start time (model wl data)',f10.2)
268         125 format(1x,i7,1x,i4,4i3,f10.3)
269
270         !-----
271
272         c           Determine which points are above crlevel_high,
273         c           and which points are below crlevel_low.
274
275         c           Variables :
276         c           crlevel_low - low water critical value
277         c           crlevel_high - high water critical value
278         c           nc2sig - counter for number of points above
279         c                       or below specified critical value
280         c           tdiff - time check, time difference between

```

Program Listing 2.3. Match.event.f (continued)

```

281      c          observed and model
282      c          wlo - observed water level (extreme)
283      c          time_o - time which corresponds to the observed
284      c          extreme value
285      c          wlm - model water level (extreme)
286      c          time_m - time which corresponds to the model
287      c          extreme value
288
289      crlevel_low = -crlevel
290      crlevel_high = crlevel + range_offset(ns)
291      nc2sig = 0
292      tm_old = 0.0
293
294
295      175      continue
296      if(option.eq.'forecast'.or.option.eq.'nowcast*')then
297          read(lunmod,*,end=180)t_mod,wl_mod
298      endif
299      if(option.eq.'astronom')then
300          read(lunmod,125,end=180)istation_id,iyear,
301      *          imonth,iday,ihr,imin,wl_pred
302          write(6,*)istation_id,iyear,imonth,iday,
303      *          ihr,wl_pred
304          call calcjd(jday,imonth,iday,iyear)
305          write(6,*)jday
306          rday = float(jday) + float(ihr)/24.0
307          write(6,*)rday
308          t_mod = rday
309          wl_mod = wl_pred
310      endif
311      read(lunobs,*,end=180)t_obs,wl_obs
312      if(t_obs.gt.endjd)goto 180
313
314      177      continue
315      tdiff = abs(t_obs-t_mod)
316      if(tdiff.gt.small)then
317          write(6,*)stanam(ns),t_obs,t_mod
318          if(t_obs.gt.t_mod)then
319              read(lunmod,*,end=180)t_mod,wl_mod
320              goto 177
321          endif
322          if(t_obs.lt.t_mod)then
323              read(lunobs,*,end=180)t_obs,wl_obs
324              goto 177
325          endif
326          if(idebug.eq.4)then
327              write(18,1039)stanam(ns)
328              write(18,1006)t_obs,t_mod
329          endif
330      endif
331
332      if(tdiff.gt.small)then
333          write(6,131)t_obs,t_mod,stanam(ns)
334          stop
335      endif
336

```

Program Listing 2.3. Match.event.f (continued)

```

337         if(wl_mod.gt.99.90)goto 175
338         if(wl_obs.ge.crlevel_high.or.wl_obs.le.crlevel_low.or.
339         *   wl_mod.ge.crlevel_high.or.wl_mod.le.crlevel_low)then
340             if(wl_obs.ge.crlevel_high.and.wl_mod.le.crlevel_low)then
341                 write(6,132)stanam(ns),t_obs,wl_obs,t_mod,
342                 *   wl_mod
343             goto 175
344         endif
345         nc2sig = nc2sig + 1
346         time_o(nc2sig) = t_obs
347         wlo(nc2sig) = wl_obs
348         time_m(nc2sig) = t_mod
349         wlm(nc2sig) = wl_mod
350         if(wl_obs.ge.crlevel_high.or.wl_mod.ge.crlevel_high)then
351             iflag(nc2sig) = 1
352         endif
353         if(wl_obs.le.crlevel_low.or.wl_mod.le.crlevel_low)then
354             iflag(nc2sig) = 0
355         endif
356     endif
357     goto 175
358 180 continue
359
360
361     131 format(' Program stopped due to time discrepancy between ',
362     *   'obs ',f9.4,/,', and model ',f9.4,' for station ',
363     *   a13)
364     132 format(/,a13,/,',At time',f10.4,', observed wl =',f9.4,/,
365     *   'At time',f10.4,', model wl =',f9.4)
366
367     !-----
368
369     c   Loop through extreme points from 1 through nc2sig,
370     c   group extreme points together in events.  Identify
371     c   events as being either high water events or low
372     c   water events with iflg.
373
374     c   Variables :
375     c       nevent - counter for number of events (by station)
376     c       iflg(nt) - iflg = 1, high water
377     c                   - iflg = 0, low water
378     c   nh(ns,nevent) - counts number of points in each event
379     c   time_peako - time which corresponds to observed extreme
380     c                   value grouped by station, event no., and
381     c                   point number of event
382     c   wlev_peako - observed water level value grouped by
383     c                   station, event no., and point number
384     c   time_peakm - time which corresponds to model extreme
385     c                   value grouped by station, event no., and
386     c                   point number of event
387     c   wlev_peakm - model water level value grouped by station,
388     c                   event no., and point number
389
390
391     nevent = 0
392     nevent_high = 0

```

Program Listing 2.3. Match.event.f (continued)

```

393         nevent_low = 0
394
395     do 190 nt=1,nc2sig
396     c     Look at first 2sigma point
397         if(nt.eq.1)then
398             nevent = 1
399             if(iflag(nt).eq.1)then
400                 iflg(nevent) = 1
401                 iflag_sta(ns,nevent) = 1
402                 if(idebug.eq.2)write(6,192)nevent
403                 nevent_high = 1
404             endif
405             if(iflag(nt).eq.0)then
406                 iflg(nevent) = 0
407                 iflag_sta(ns,nevent) = 0
408                 if(idebug.eq.2)write(6,193)nevent
409                 nevent_low = 1
410             endif
411             nh(ns,nevent) = 1
412         endif
413
414     c     Look at remainder of extreme points
415         if(nt.gt.1)then
416             tmdiff = time_o(nt) - tm_old
417             if(tmdiff.gt.peak_intrvl.or.iflag(nt).ne.
418             *             iflag(nt-1))then
419                 nevent = nevent + 1
420                 if(iflag(nt).eq.1)then
421                     iflg(nevent) = 1
422                     iflag_sta(ns,nevent) = 1
423                     if(idebug.eq.2)write(6,192)nevent
424                     nevent_high = nevent_high + 1
425                 endif
426                 if(iflag(nt).eq.0)then
427                     iflg(nevent) = 0
428                     iflag_sta(ns,nevent) = 0
429                     if(idebug.eq.2)write(6,193)nevent
430                     nevent_low = nevent_low + 1
431                 endif
432                 nh(ns,nevent) = 1
433             else
434                 nh(ns,nevent) = nh(ns,nevent) + 1
435             endif
436         endif
437
438         time_peako(ns,nevent,nh(ns,nevent)) = time_o(nt)
439         wlev_peako(ns,nevent,nh(ns,nevent)) = wlo(nt)
440         time_peakm(ns,nevent,nh(ns,nevent)) = time_m(nt)
441         wlev_peakm(ns,nevent,nh(ns,nevent)) = wlm(nt)
442         tm_old = time_o(nt)
443     190 continue
444
445
446         if(idebug.eq.3)then
447             write(6,191)stanam(ns),nevent,nevent_high,
448             *             nevent_low

```

Program Listing 2.3. Match.event.f (continued)

```

449         endif
450
451         nevent1(ns) = nevent_low
452         nevent_sta(ns) = nevent
453
454
455         191 format(/, ' for station ', a13, /, 1x, i4,
456             *      ' events; ', i3, ' high water, ', i3, ' low water')
457         192 format(' Peak', i3, ' is a high water level event')
458         193 format(' Peak', i3, ' is a low water level event')
459
460     !-----
461
462     c      Loop (200) thru events (1 thru nevent) for each station;
463     c      iflg = 1 denotes peak, iflg = 0 denotes trough.  Loop
464     c      (210) through points (1 thru nh(ns,np)) for each
465     c      event to determine high point of each peak and low
466     c      point of each trough.  Also, determine initial
467     c      extreme point, model and observed, for each peak
468     c      and trough.
469
470
471         if(idebug.eq.3)write(6,199)
472
473     c      npt_ospk(ns,np) - number of points in observed event
474     c      npt_modpk(ns,np) - number of points in model event
475     c      time_chk - check for peak value occurring at start
476     c      of next month
477     c      tm_lstval(ns,2,np) - time of 1st wl value during observed
478     c      event
479     c      tm_lstval(ns,1,np) - time of 1st wl value during model
480     c      event
481     c      wl_higho - peak observed water level value
482     c      tm_higho - time of peak obs water level
483     c      wl_lowo - low observed water level value
484     c      tm_lowo - time of low obs water level
485     c      wl_highm - peak model water level value
486     c      tm_highm - time of peak model water level
487     c      wl_lowm - low model water level value
488     c      tm_lowm - time of low model water level
489
490     do 200 np=1,nevent
491         npt_ospk(ns,np) = 0
492         npt_modpk(ns,np) = 0
493         if(idebug.eq.3)write(6,201)np
494         wl_higho = -9.99
495         wl_lowo = 99.99
496         wl_highm = -9.99
497         wl_lowm = 99.99
498         do 210 n=1,nh(ns,np)
499             if(idebug.eq.3)write(6,1006)time_peako(ns,np,n),
500             *      wlev_peako(ns,np,n),wlev_peakm(ns,np,n)
501             if(iflg(np).eq.1)then
502                 if(wlev_peako(ns,np,n).ge.crlevel_high)then
503                     if(npt_ospk(ns,np).eq.0)then
504                         tm_lstval(ns,2,np) = time_peako(ns,np,n)

```

Program Listing 2.3. Match.event.f (continued)

```

505         endif
506         npt_ospk(ns,np) = npt_ospk(ns,np) + 1
507     endif
508     if(wlev_peako(ns,np,n).gt.wl_higho)then
509         wl_higho = wlev_peako(ns,np,n)
510         tm_higho = time_peako(ns,np,n)
511     endif
512     if(wlev_peakm(ns,np,n).ge.crlevel_high)then
513         if(npt_modpk(ns,np).eq.0)then
514             tm_lstval(ns,1,np) = time_peakm(ns,np,n)
515         endif
516         npt_modpk(ns,np) = npt_modpk(ns,np) + 1
517     endif
518     if(wlev_peakm(ns,np,n).gt.wl_highm)then
519         wl_highm = wlev_peakm(ns,np,n)
520         tm_highm = time_peakm(ns,np,n)
521     endif
522     endif
523     if(iflg(np).eq.0)then
524         if(wlev_peako(ns,np,n).le.crlevel_low)then
525             if(npt_ospk(ns,np).eq.0)then
526                 tm_lstval(ns,2,np) = time_peako
527                 *
528                     (ns,np,n)
529             endif
530             npt_ospk(ns,np) = npt_ospk(ns,np) + 1
531         endif
532         if(wlev_peako(ns,np,n).lt.wl_lowo)then
533             wl_lowo = wlev_peako(ns,np,n)
534             tm_lowo = time_peako(ns,np,n)
535         endif
536         if(wlev_peakm(ns,np,n).le.crlevel_low)then
537             if(npt_modpk(ns,np).eq.0)then
538                 *
539                 tm_lstval(ns,1,np) = time_peakm
540                 (ns,np,n)
541             endif
542             npt_modpk(ns,np) = npt_modpk(ns,np) + 1
543         endif
544         if(wlev_peakm(ns,np,n).lt.wl_lowm)then
545             wl_lowm = wlev_peakm(ns,np,n)
546             tm_lowm = time_peakm(ns,np,n)
547         endif
548     endif
549     210 continue
550
551     c Variables :
552     c wl_pkval(ns,1,np) - model peak water level value
553     c                       stored in array by event
554     c tm_pkval(ns,1,np) - time of model peak value stored in
555     c                       array by event
556     c wl_pkval(ns,2,np) - observed peak water level value stored
557     c                       in array by event
558     c tm_pkval(ns,2,np) - time of observed peak value stored in
559     c                       array by event
560
561     if(iflg(np).eq.1)then

```

Program Listing 2.3. Match.event.f (continued)

```

561         if(idebug.eq.2)write(6,202)wl_higho,tm_higho
562         call timehi(caldayo,tm_higho,idebug)
563         time_chk = caldayo - 1.000
564         if(abs(time_chk).lt.small.and.tm_higho.gt.
565         *         startjd+5.0)then
566             caldayo = caldayo + dayspl
567             write(6,204)stanam(ns),np,tm_higho,caldayo
568         endif
569         if(idebug.eq.2)write(6,203)wl_highm,tm_highm
570         call timehi(caldaym,tm_highm,idebug)
571         time_chk = caldaym - 1.000
572         if(abs(time_chk).lt.small.and.tm_highm.gt.
573         *         startjd+5.0)then
574             caldaym = caldaym + dayspl
575             write(6,204)stanam(ns),np,tm_highm,caldaym
576         endif
577         if(wl_highm.ge.crlevel_high)then
578             tm_pkval(ns,1,np) = tm_highm
579             tm_pkvalWr(ns,1,np) = tm_highm
580             cald_pkval(ns,1,np) = caldaym
581             wl_pkval(ns,1,np) = wl_highm
582             wl_pkvalWr(ns,1,np) = wl_highm
583         else
584             tm_1stval(ns,1,np) = big_value
585             tm_pkval(ns,1,np) = tm_highm
586             tm_pkvalWr(ns,1,np) = big_value
587             cald_pkval(ns,1,np) = big_value
588             wl_pkval(ns,1,np) = wl_highm
589             wl_pkvalWr(ns,1,np) = big_value
590         endif
591         if(wl_higho.ge.crlevel_high)then
592             tm_pkval(ns,2,np) = tm_higho
593             tm_pkvalWr(ns,2,np) = tm_higho
594             cald_pkval(ns,2,np) = caldayo
595             wl_pkval(ns,2,np) = wl_higho
596             wl_pkvalWr(ns,2,np) = wl_higho
597         else
598             tm_1stval(ns,2,np) = big_value
599             tm_pkval(ns,2,np) = tm_higho
600             tm_pkvalWr(ns,2,np) = big_value
601             cald_pkval(ns,2,np) = big_value
602             wl_pkval(ns,2,np) = wl_higho
603             wl_pkvalWr(ns,2,np) = big_value
604         endif
605     endif
606     if(iflg(np).eq.0)then
607         if(idebug.eq.2)write(6,205)wl_lowo,tm_lowo
608         call timehi(caldayo,tm_lowo,idebug)
609         time_chk = caldayo - 1.000
610         if(abs(time_chk).lt.small.and.tm_lowo.gt.
611         *         startjd+5.0)then
612             write(6,204)stanam(ns),np,tm_lowo,caldayo
613             caldayo = caldayo + dayspl
614         endif
615         if(idebug.eq.2)write(6,206)wl_lowm,tm_lowm
616         call timehi(caldaym,tm_lowm,idebug)

```

Program Listing 2.3. Match.event.f (continued)


```

617         time_chk = caldaym - 1.000
618         if(abs(time_chk).lt.small.and.tm_lowm.gt.
619            *          startjd+5.0)then
620             write(6,204)stanam(ns),np,tm_lowm,caldaym
621             caldaym = caldaym + dayspl
622         endif
623         if(wl_lowm.le.crlevel_low)then
624             tm_pkval(ns,1,np) = tm_lowm
625             tm_pkvalWr(ns,1,np) = tm_lowm
626             cald_pkval(ns,1,np) = caldaym
627             wl_pkval(ns,1,np) = wl_lowm
628             wl_pkvalWr(ns,1,np) = wl_lowm
629         else
630             tm_lstval(ns,1,np) = big_value
631             tm_pkval(ns,1,np) = tm_lowm
632             tm_pkvalWr(ns,1,np) = big_value
633             cald_pkval(ns,1,np) = big_value
634             wl_pkval(ns,1,np) = wl_lowm
635             wl_pkvalWr(ns,1,np) = big_value
636         endif
637         if(wl_lowo.le.crlevel_low)then
638             tm_pkval(ns,2,np) = tm_lowo
639             tm_pkvalWr(ns,2,np) = tm_lowo
640             cald_pkval(ns,2,np) = caldayo
641             wl_pkval(ns,2,np) = wl_lowo
642             wl_pkvalWr(ns,2,np) = wl_lowo
643         else
644             tm_lstval(ns,2,np) = big_value
645             tm_pkval(ns,2,np) = tm_lowo
646             tm_pkvalWr(ns,2,np) = big_value
647             cald_pkval(ns,2,np) = big_value
648             wl_pkval(ns,2,np) = wl_lowo
649             wl_pkvalWr(ns,2,np) = big_value
650         endif
651     endif
652
653     200 continue
654
655
656
657     199 format('      Time          obs          model')
658     201 format(' spike #',i3)
659     202 format(1x,'High water level (obs) is',f8.4,' occuring at',
660            *      f10.4)
661     203 format(1x,'High water level (mod) is',f8.4,' occuring at',
662            *      f10.4)
663     204 format(1x,a13,'Peak value event ',i3,2x,'occurs at',
664            *      2f9.3)
665     205 format(' Low water level (obs) is',f8.4,' occuring at',
666            *      f10.4)
667     206 format(' Low water level (mod) is',f8.4,' occuring at',
668            *      f10.4)
669
670     !-----
671
672     c Write out values for critical low and high values

```

Program Listing 2.3. Match.event.f (continued)

```

673
674         if(idebug.eq.5)then
675             write(19,151)stanam(ns),crlevel_low,crlevel_high,
676             *             range_offset(ns)
677         endif
678         crlevelL(ns) = crlevel_low
679         crlevelH(ns) = crlevel_high
680     100 continue
681
682
683
684     151 format(1x,a13,f10.3,2x,2(1x,f12.3))
685
686 *****
687
688 c     Group events as to success, failure, false.
689
690 c     Variables :
691 c
692 c     tm_pkval(ns,1,np) - time of model extreme value
693 c                       stored in array by event
694 c     tm_pkval(ns,2,np) - time of observed extreme value
695 c                       stored in array by event
696 c
697 c     int_big - integer dummy variable for ihr_diff
698 c     big_value - 99999.9999 (?)
699 c     nsuccess - number of times model successfully
700 c               predicts observed event.
701 c     nfailure - number of times model fails to predict
702 c               an observed event.
703 c     nfalse - number of times model predicts event
704 c             which was not observed.
705 c     nfalse_high - number of times high water event
706 c                 is forecast but not observed
707 c     nfalse_low - number of times low water event
708 c                 is forecast but not observed
709 c     nevent_sta(ns) - number of events for each station
710 c     dtime - time difference (real) between
711 c             observed and model event peaks
712 c     idel_strtm() - time difference in hours between
713 c                 observed and model event start times
714 c     ihr_diff - time difference in hours between
715 c                 observed and model event peaks
716 c     dwl - difference in water level between
717 c           observed and model event peaks
718
719
720 c     Initialization :
721 c     int_big = 999999
722 c     small = 0.01
723
724
725 c     Write header information to each output file
726
727 c     call wr_header(18)
728 c     call wr_header(21)

```

Program Listing 2.3. Match.event.f (continued)

```

729      call wr_header(22)
730      write(22,406)
731      call wr_header(23)
732      write(23,407)
733
734
735      402 format(/,20x,'Model data',15x,'Observed data',
736      *      /,'Station event# jul day cal day wl',
737      *      4x,'jul day cal day wl Dt(hour) Dwl(m)')
738      406 format(' high water events')
739      407 format(' low water events')
740
741      !-----
742
743      c      Loop 440 is the station loop. 450 loops through
744      c      events by station.
745      do 440 ns=1,nsta
746          nsuccess = 0
747          nsuccess_high = 0
748          nsuccess_low = 0
749          nfailure = 0
750          nfail_high = 0
751          nfail_low = 0
752          nfalse = 0
753          nfalse_high = 0
754          nfalse_low = 0
755          dwls_total = 0.0
756          dwlFail_total = 0.0
757          dwlFalse_total = 0.0
758          dwls_totalh = 0.0
759          dwls_totall = 0.0
760          dwlFail_totalh = 0.0
761          dwlFail_totall = 0.0
762          dwlFalse_totalh = 0.0
763          dwlFalse_totall = 0.0
764
765          if(option.eq.'forecast')then
766              write(18,420)stanam(ns),headr3(ns)
767          else
768              write(18,421)stanam(ns)
769          endif
770          write(18,402)
771
772          do 450 np=1,nevent_sta(ns)
773              dtime = tm_pkval(ns,1,np) - tm_pkval(ns,2,np)
774              hr_diff = dtime * 24.0
775              ihr_diff = nint(hr_diff)
776              del_strttm = (tm_1stval(ns,1,np)-
777      *                  tm_1stval(ns,2,np)) * 24.0
778
779      c      Case 1 : Success
780          if(tm_pkvalWr(ns,2,np).lt.90000.0.and.tm_pkvalWr
781      *      (ns,1,np).lt.90000.0)then
782              nsuccess = nsuccess + 1
783              dwls = wl_pkval(ns,1,np) - wl_pkval(ns,2,np)
784              ievent(1,nsuccess) = np

```

Program Listing 2.3. Match.event.f (continued)

```

785         idel_peaktm(1,nsuccess) = ihr_diff
786         idel_strtm(1,nsuccess) = nint(del_strttm)
787         id_duration = npt_modpk(ns,np) -
788             *           npt_obs_pk(ns,np)
789         idel_duration(nsuccess) = id_duration
790         forc_wl(1,nsuccess) = wl_pkval(ns,1,np)
791         obs_wl(1,nsuccess) = wl_pkval(ns,2,np)
792         tm_obspeak(1,nsuccess) = tm_pkval(ns,2,np)
793         delta_wl(1,nsuccess) = dwls
794         dwls_total = dwls_total + abs(dwls)
795         if(iflag_sta(ns,np).eq.1)then
796             i_flag(1,nsuccess) = 1
797             nsuccess_high = nsuccess_high + 1
798             dwls_totalh = dwls_totalh + abs(dwls)
799         else
800             i_flag(1,nsuccess) = 0
801             nsuccess_low = nsuccess_low + 1
802             dwls_totall = dwls_totall +
803                 *           abs(dwls)
804             endif
805             dwl = dwls
806         endif
807
808     c     Case 2 : Failure
809         if(tm_pkvalWr(ns,1,np).gt.90000.0)then
810             nfailure = nfailure + 1
811             dwlFail = wl_pkval(ns,1,np) - wl_pkval(ns,2,np)
812             ievent(2,nfailure) = np
813             idel_peaktm(2,nfailure) = ihr_diff
814             ihr_diff = int_big
815             forc_wl(2,nfailure) = wl_pkval(ns,1,np)
816             obs_wl(2,nfailure) = wl_pkval(ns,2,np)
817             tm_obspeak(2,nfailure) = tm_pkval(ns,2,np)
818             delta_wl(2,nfailure) = dwlFail
819             dwlFail_total = dwlFail_total + abs(dwlFail)
820             if(iflag_sta(ns,np).eq.1)then
821                 i_flag(2,nfailure) = 1
822                 nfail_high = nfail_high + 1
823                 dwlFail_totalh = dwlFail_totalh +
824                     *           abs(dwlFail)
825             endif
826             if(iflag_sta(ns,np).eq.0)then
827                 i_flag(2,nfailure) = 0
828                 nfail_low = nfail_low + 1
829                 dwlFail_totall = dwlFail_totall +
830                     *           abs(dwlFail)
831             endif
832             dwl = big_value
833         endif
834
835     c     Case 3 : false alarm
836         if(tm_pkvalWr(ns,2,np).gt.90000.0)then
837             nfalse = nfalse + 1
838             dwlFalse = wl_pkval(ns,1,np) - wl_pkval(ns,2,np)
839             ievent(3,nfalse) = np
840             idel_peaktm(3,nfalse) = ihr_diff

```

Program Listing 2.3. Match.event.f (continued)

```

841             ihr_diff = int_big
842             forc_wl(3,nfalse) = wl_pkval(ns,1,np)
843             obs_wl(3,nfalse) = wl_pkval(ns,2,np)
844             tm_obspeak(3,nfalse) = tm_pkval(ns,2,np)
845             delta_wl(3,nfalse) = dwlFalse
846             dwlFalse_total = dwlFalse_total + abs(dwlFalse)
847             if(iflag_sta(ns,np).eq.1)then
848                 i_flag(3,nfalse) = 1
849                 nfalse_high = nfalse_high + 1
850                 dwlFalse_totalh = dwlFalse_totalh +
851                 *                               abs(dwlFalse)
852             endif
853             if(iflag_sta(ns,np).eq.0)then
854                 i_flag(3,nfalse) = 0
855                 nfalse_low = nfalse_low + 1
856                 dwlFalse_totall = dwlFalse_totall +
857                 *                               abs(dwlFalse)
858             endif
859             dwl = big_value
860         endif
861
862
863
864         !-----
865
866         c      Write output to table.out, the monthly output file.
867
868
869             write(18,442)(tm_1stval(ns,nm,np),nm=1,2)
870             write(18,445)np,(tm_pkvalWr(ns,nm,np),
871             *               cald_pkval(ns,nm,np),wl_pkvalWr
872             *               (ns,nm,np),nm=1,2),ihr_diff,dwl
873             write(18,446)nh(ns,np),npt_modpk(ns,np),
874             *               npt_obsPk(ns,np)
875         450     continue
876
877
878         442 format(17x,f9.3,16x,f9.3)
879         445 format(13x,i2,2(2x,f9.3,f7.3,f7.4),i5,f9.3)
880         446 format(7x,i3,' total points in event;',
881             *       i3,' model points,',i3,' observed')
882
883         *****
884
885         c Write output to table2.out.
886         c Calculate mean water level differences for three cases.
887         c
888         c Variables :
889         c
890         c     dwlS_mean - mean water level difference (success)
891         c     dwlFail_mean - mean water level difference (failure)
892         c     dwlFalse_mean - mean water level difference (false)
893
894
895             if(option.eq.'forecast')then
896                 write(21,420)stanam(ns),headr3(ns)

```

Program Listing 2.3. Match.event.f (continued)

```

897         write(22,420)stanam(ns),headr3(ns)
898         write(23,420)stanam(ns),headr3(ns)
899     endif
900     if(option.eq.'nowcast*'.or.option.eq.'astronom')then
901         write(21,421)stanam(ns)
902         write(22,421)stanam(ns)
903         write(23,421)stanam(ns)
904     endif
905     write(21,438)crlevelL(ns)
906     write(21,437)crlevelH(ns)
907     write(22,437)crlevelH(ns)
908     if(nsucces_low.gt.0)write(23,438)crlevelL(ns)
909
910
911     420 format(//,1x,a13,/,a28)
912     421 format(//,1x,a13)
913     437 format('High water critical level,',f10.3)
914     438 format(' Low water critical level,',f10.3)
915
916     !-----
917
918     c    Case 1 : Success
919
920         if(nsucces.gt.0)then
921             if(option.eq.'forecast')then
922                 write(21,471)
923             endif
924             if(option.eq.'nowcast*')then
925                 write(21,472)
926             endif
927             if(option.eq.'astronom')then
928                 write(21,473)
929             endif
930             write(21,474)
931         endif
932         if(nsucces_high.gt.0)then
933             if(option.eq.'forecast')write(22,471)
934             if(option.eq.'nowcast*')write(22,472)
935             if(option.eq.'astronom')write(22,473)
936             write(22,474)
937         endif
938         if(nsucces_low.gt.0)then
939             if(option.eq.'forecast')write(23,471)
940             if(option.eq.'nowcast*')write(23,472)
941             if(option.eq.'astronom')write(23,473)
942             write(23,474)
943         endif
944
945         do 455 n=1,nsucces
946             write(21,447)n,ievent(1,n),idel_strtm(1,n),
947             *         idel_duration(n),idel_peaktm(1,n),
948             *         forc_wl(1,n),obs_wl(1,n),delta_wl(1,n),
949             *         tm_obspeak(1,n)
950             if(i_flag(1,n).eq.1)then
951                 write(22,447)n,ievent(1,n),idel_strtm(1,n),
952             *         idel_duration(n),idel_peaktm(1,n),

```

Program Listing 2.3. Match.event.f (Continued)

```

953      *          forc_wl(1,n),obs_wl(1,n),delta_wl(1,n),
954      *          tm_obspeak(1,n)
955      endif
956      if(i_flag(1,n).eq.0)then
957          write(23,447)n,ievent(1,n),idel_strtm(1,n),
958      *          idel_duration(n),idel_peaktm(1,n),
959      *          forc_wl(1,n),obs_wl(1,n),delta_wl(1,n),
960      *          tm_obspeak(1,n)
961      endif
962 455      continue
963
964      if(nsucces.gt.0)then
965          dwls_mean = dwls_total/float(nsucces)
966          write(21,461)dwls_mean
967      endif
968
969      if(nsucces_high.gt.0)then
970          dwls_meanh = dwls_totalh/float(nsucces_high)
971          write(22,451)nsucces_high,dwls_meanh
972      endif
973      if(nsucces_low.gt.0)then
974          dwls_meanl = dwls_totall/float(nsucces_low)
975          write(23,452)nsucces_low,dwls_meanl
976      endif
977
978
979      447 format(1x,i3,4(3x,i5),2f11.4,f9.3,f9.3)
980      461 format(' mean difference of peak water levels for',
981      *          /,' "success" forecasts is',f7.4,' meters')
982      471 format(/,'success event dstart delta dpeak',
983      *          ' forecast observed dwl obs peak')
984      472 format(/,'success event dstart delta dpeak',
985      *          ' nowcast observed dwl obs peak')
986      473 format(/,'success event dstart delta dpeak',
987      *          ' predicted observed dwl obs peak')
988      474 format(' number number time duration time ',
989      *          ' wl(m) wl(m) peak(m) time(jd)')
990
991      !-----
992
993  c      Case 2 : Failure
994
995      if(nfailure.gt.0)then
996          if(option.eq.'forecast')then
997              write(21,481)
998          endif
999          if(option.eq.'nowcast*')then
1000              write(21,482)
1001          endif
1002          if(option.eq.'astronom')then
1003              write(21,483)
1004          endif
1005          write(21,494)
1006      endif
1007      if(nfail_high.gt.0)then
1008          if(option.eq.'forecast')write(22,481)

```

Program Listing 2.3. Match.event.f (continued)

```

1009         if(option.eq.'nowcast*')write(22,482)
1010         if(option.eq.'astronom')write(22,483)
1011         write(22,494)
1012     endif
1013     if(nfail_low.gt.0)then
1014         if(option.eq.'forecast')write(23,481)
1015         if(option.eq.'nowcast*')write(23,482)
1016         if(option.eq.'astronom')write(23,483)
1017         write(23,494)
1018     endif
1019
1020     do 456 n=1,nfailure
1021         write(21,448)n,ievent(2,n),idel_peaktm(2,n),
1022         *         forc_wl(2,n),obs_wl(2,n),delta_wl(2,n),
1023         *         tm_obspeak(2,n)
1024         if(i_flag(2,n).eq.1)then
1025             write(22,448)n,ievent(2,n),idel_peaktm(2,n);
1026             forc_wl(2,n),obs_wl(2,n),delta_wl(2,n),
1027             *         tm_obspeak(2,n)
1028         endif
1029         if(i_flag(2,n).eq.0)then
1030             write(23,448)n,ievent(2,n),idel_peaktm(2,n),
1031             *         forc_wl(2,n),obs_wl(2,n),delta_wl(2,n),
1032             *         tm_obspeak(2,n)
1033         endif
1034     456 continue
1035
1036     if(nfailure.gt.0)then
1037         dwlFail_mean = dwlFail_total/float(nfailure)
1038         write(21,462)dwlFail_mean
1039     endif
1040
1041     if(nfail_high.gt.0)then
1042         dwlFail_meanh = dwlFail_totalh/float(nfail_high)
1043         write(22,451)nfail_high,dwlFail_meanh
1044     endif
1045     if(nfail_low.gt.0)then
1046         dwlFail_meanl = dwlFail_totall/float(nfail_low)
1047         write(23,452)nfail_low,dwlFail_meanl
1048     endif
1049
1050     481 format(/,'failure event   dpeak forecast',
1051     *         '   observed   dwl   obs peak')
1052     482 format(/,'failure event   dpeak nowcast ',
1053     *         '   observed   dwl   obs peak')
1054     483 format(/,'failure event   dpeak predicted',
1055     *         '   observed   dwl   obs peak')
1056     462 format(' mean difference of peak water levels for',
1057     *         /,' "failure" forecasts is',f7.4,' meters')
1058
1059     !-----
1060
1061     c    Case 3 : False
1062
1063     if(nfalse.gt.0)then
1064         if(option.eq.'forecast')then

```

Program Listing 2.3. Match.event.f (continued)


```

1065         write(21,491)
1066     endif
1067     if(option.eq.'nowcast*')then
1068         write(21,492)
1069     endif
1070     if(option.eq.'astronom')then
1071         write(21,493)
1072     endif
1073     write(21,494)
1074 endif
1075 if(nfalse_high.gt.0)then
1076     if(option.eq.'forecast')write(22,491)
1077     if(option.eq.'nowcast*')write(22,492)
1078     if(option.eq.'astronom')write(22,493)
1079     write(22,494)
1080 endif
1081 if(nfalse_low.gt.0)then
1082     if(option.eq.'forecast')write(23,491)
1083     if(option.eq.'nowcast*')write(23,492)
1084     if(option.eq.'astronom')write(23,493)
1085     write(23,494)
1086 endif
1087
1088 do 457 n=1,nfalse
1089     write(21,448)n,ievent(3,n),idel_peaktm(3,n),
1090     *         forc_wl(3,n),obs_wl(3,n),delta_wl(3,n),
1091     *         tm_obspeak(3,n)
1092     if(i_flag(3,n).eq.1)then
1093         write(22,448)n,ievent(3,n),idel_peaktm(3,n),
1094     *         forc_wl(3,n),obs_wl(3,n),delta_wl(3,n),
1095     *         tm_obspeak(3,n)
1096     endif
1097     if(i_flag(3,n).eq.0)then
1098         write(23,448)n,ievent(3,n),idel_peaktm(3,n),
1099     *         forc_wl(3,n),obs_wl(3,n),delta_wl(3,n),
1100     *         tm_obspeak(3,n)
1101     endif
1102 457 continue
1103
1104     if(nfalse.gt.0)then
1105         dwlFalse_mean = dwlFalse_total/float(nfalse)
1106         write(21,463)dwlFalse_mean
1107     endif
1108     if(nfalse_high.gt.0)then
1109         dwlFalse_meanh = dwlFalse_totalh/float(nfalse_high)
1110         write(22,451)nfalse_high,dwlFalse_meanh
1111     endif
1112     if(nfalse_low.gt.0)then
1113         dwlFalse_meanl = dwlFalse_totall/float(nfalse_low)
1114         write(23,452)nfalse_low,dwlFalse_meanl
1115     endif
1116
1117     write(21,469)nsuccess,nfailure,nfalse
1118 440 continue
1119
1120

```

Program Listing 2.3. Match.event.f (continued)

```

1121     491 format(//, ' false event dpeak forecast',
1122     *           ' observed dwl obs peak')
1123     492 format(//, ' false event dpeak nowcast',
1124     *           ' observed dwl obs peak')
1125     493 format(//, ' false event dpeak predicted',
1126     *           ' observed dwl obs peak')
1127     494 format(' number number time wl(m) wl(m) ',
1128     *           ' peak(m) time(jd)')
1129     463 format(' mean difference of peak water levels for',
1130     *           //, ' "false" forecasts is',f7.4,' meters')
1131     469 format(//, ' Success Failure false',//,
1132     *           8x,i5,2(4x,i4)//)
1133
1134     !-----
1135
1136     448 format(1x,i3,2(3x,i5),2f11.4,f9.3,f9.3)
1137     451 format(' for ',i3,' high water events, mean difference',
1138     *           //, ' of the peak water levels is ',f7.4,' meters')
1139     452 format(' for ',i3,' low water events, mean difference',
1140     *           //, ' of the peak water levels is ',f7.4,' meters')
1141
1142
1143     *****
1144
1145     1003 format(a28)
1146     1039 format(a13)
1147     1040 format(//,1x,a13)
1148     1006 format(1x,3f10.4)
1149
1150
1151
1152     STOP
1153     END

```

Program Listing 2.3. Match.event.f (continued)

```

1      SUBROUTINE CALCJD (IJD,IMON,IDAY,IYR)
2      C
3      C**** THIS SUBROUTINE CONVERTS CALENDER TO JULIAN DAY (IJD)
4      C
5      DIMENSION  IDTBLE(12),ILTBLE(12)
6      C
7      DATA (IDTBLE(I),I=1,12)/1,32,60,91,121,152,182,213,244,
8      1      274,305,335/
9      DATA (ILTBLE(I),I=1,12)/1,32,61,92,122,153,183,214,245,
10     1      275,306,336/
11     C
12     C**** TEST FOR LEAP YEAR
13     C
14     ISW = 1
15     IF (MOD(IYR,4).EQ.0)  ISW = 2
16
17     GO TO (9,10)  ISW
18     9 IJD = IDTBLE(IMON) + IDAY - 1
19     RETURN
20    10 IJD = ILTBLE(IMON) + IDAY - 1
21     RETURN
22     END

```

Program Listing 2.3. Match.event.f (continued)

```

1      subroutine timehi(caldy,tmmax,ibug)
2
3      c      Purpose : To take the real value julian day
4      c      and convert to calendar day with fraction.
5      c      Version date : November 1, 1997
6      c
7      c      Author : Phil Richardson
8
9      *****
10
11      itmhi = tmmax
12      rtmhi = float(itmhi)
13      tmdiff = tmmax - rtmhi
14
15
16      call jdgreg(rtmhi,imonth,iday,iyear)
17
18      caldy = float(iday) + tmdiff
19      if(ibug.eq.2)then
20          write(6,101)imonth,iday,iyear,caldy
21      endif
22
23
24      101 format(1x,i2,'/',i2,'/',i4,' Calendar Day ',f9.4,/)
25
26      return
27      end

```

Program Listing 2.3. Match.event.f (continued)

```

1      ! SUBROUTINE NAME : JDGREG
2      !
3      ! PURPOSE : This subroutine converts the Julian date
4      !           to Gregorian date.
5      !
6      !
7      ! VARIABLE NAMES :
8      ! DOUBLE PRECISION RJD - - - - - Julian date
9      ! DIMENSION JDAY(13) - - - - - Non-leap year
10     ! DIMENSION JDAYL(13) - - - - - Leap year
11     !
12     !-----
13
14     SUBROUTINE JDGREG(rjdy,imon,ida,iyr)
15
16
17     !:    DOUBLE PRECISION RJD
18     DIMENSION JDAY(13), JDAYL(13)
19
20     DATA JDAY/0,31,59,90,120,151,181,212,243,273,304,334,365/
21     DATA JDAYL/0,31,60,91,121,152,182,213,244,274,305,335,366/
22
23     iyr = 2000
24
25     rjd = rjdy
26     IDA = INT(RJD)
27
28
29     IF(MOD(IYR,4).EQ.0.AND.MOD(IYR,100).NE.0
30     & .OR. MOD(IYR,400).EQ.0)THEN
31
32     ! Find the month for IDAY --- leap year calender
33     !
34         ILEAP = 1
35         DO I = 1,12
36             IF (IDA.GT.JDAYL(I)) IMO = I
37         END DO
38
39     ! Day of the month
40
41         IDY = IDA - JDAYL(IMO)
42     ELSE
43
44     ! Find the month for IDAY --- non-leap year
45
46         ILEAP = 0
47         DO I=1,12
48             IF(IDA.GT.JDAY(I)) IMO = I
49         ENDDO
50
51     ! Day of the month
52
53         IDY = IDA-JDAY(IMO)
54     ENDIF
55
56     IMON = IMO

```

Program Listing 2.3. Match.event.f (continued)

```
57         IDA = IDY
58
59     RETURN
60 END
```

Program Listing 2.3. Match.event.f (continued)

```

1      subroutine wr_header(lunout)
2
3      c Author : Phil Richardson
4      c
5      c Date : June 26, 2001
6
7      *****
8
9      character*8  option
10     character*13 mnyr
11     character*60 head1
12     character*25 head2
13     character*29 head3
14
15     common/headrs/head1,head2,mnyr,option,strtjd,
16     *           endjd,clevel
17
18     *****
19
20     write(6,*)option
21     write(lunout,1)head1
22     if(option.eq.'forecast')then
23         write(lunout,2)head2
24     endif
25     write(lunout,3)mnyr
26     write(lunout,4)strtjd,endjd,clevel
27
28     *****
29
30     1 format(1x,a60)
31     2 format(a25)
32     3 format(/,1x,a13)
33     4 format(' Start time of comparison = ',f7.3,', End time = ',
34     *           f7.3,/, ' critical level = ',f6.2)
35
36     return
37     end

```

Program Listing 2.3. Match.event.f (continued)

2.4 Program Wl.sigma.pro

The listing for Wl.sigma.pro is given in Program Listing 2.4. This is an IDL program used to plot a month of observed, forecast, or nowcast water level data. In addition to plotting the water level data, the program will plot two lines which designate the low water critical level and the high water critical level. The low water critical level value is the negative of crlevel, a variable read from the control file. The high water critical level value is determined by adding crlevel to the range offset. The range offset (range_offset) is also read from the control file.

The plots are annotated with a title, station name, and a legend. More than one water level signal can be put on one plot, if desired. However, the plot program generates only one plot per page.


```

1      ; Program : wl.sigma.pro
2      ;
3      ; Purpose : This program, written in the IDL programming
4      ; language, was developed to help evaluate Houston/Galveston
5      ; nowcast/forecast water level data. The program will
6      ; plot the observed and model water levels with a high
7      ; critical value and a low critical value (line) to
8      ; determine events. Wl.sigma.pro has been modified in
9      ; order to handle gaps in the data stream.
10     ;
11     ; Language : IDL
12     ;
13     ; Location : /usr/people/philr/galves/NF_eval/wlevel/plot
14     ;
15     ; Version date : July 3, 2000
16     ;
17     ; Author : Phil Richardson
18
19     ;*****
20
21     im = 1500
22     nline = 12
23
24
25     ;Initialize character strings
26     line = ' '
27     wltitle = ' '
28     filedata = ' '
29     legend = ' '
30     cntrl_file = ' '
31     time_axis = ' '
32     stat_name = ' '
33     ptype=' '
34     plotype = ' '
35     typedata = ' '
36
37     ;Initialize integer variables
38     idebug = 0
39     iyear = 0
40     ncurve = 0
41     lun = 0
42
43     ;Dimension arrays
44     ncg = intarr(nline)
45     ncntg = intarr(nline)
46
47     ilun=intarr(2)
48     numb_pts = intarr(2)
49
50     filedat = strarr(2)
51     legnd = strarr(2)
52     ravg = strarr(2)
53
54     t=fltarr(im,nline)
55     wlplt = fltarr(im,nline)
56     xl=fltarr(2,2)

```

Program Listing 2.4. Wl.sigma.pro

```

57     yl=fltarr(2)
58     xpos = fltarr(2)
59     time_strt = fltarr(2)
60     xline = fltarr(2)
61     crlevel_high = fltarr(2)
62     crlevel_low = fltarr(2)
63
64
65     small = 0.001
66     crlevel = 0.0
67     hr_intrvl = 0.09
68
69     ;*****
70
71     ; Open control file, read from control file
72
73     ;     ptype - x, ps, or tek
74     ;     idebug = 1, times (Julian dates)
75     ;           = 2, EOF result
76     ;     stat_name - station name
77     ;     tmin - start point for time (x) axis
78     ;     strttime - start time (Julian date)
79     ;     tmax - end point for time (x) axis
80     ;     endtime - end time (Julian date)
81     ;     iyear - year of plot
82     ;     ncurve - number of curves to plot
83     ;     legnd(nc) - character string, for legend
84     ;     filedat(nc) - data filenames
85     ;     time_axis - time axis name
86     ;     crlevel - critical value
87
88
89     print,'Enter name of control file '
90     read,cntrl_file
91     openr,1,cntrl_file
92
93     readf,1,ptype
94     if(ptype eq 'ps')then begin
95         readf,1,plotype
96     endif
97     readf,1,idebug
98     print,idebug,format='(2x,"idebug = ",i3)'
99
100    readf,1,stat_name
101    readf,1,typedata
102    readf,1,tmin
103    readf,1,strttime
104    readf,1,tmax
105    readf,1,endtime
106    readf,1,iyear
107    readf,1,ncurve
108    print,ncurve,format='(1x,i2," curves to be plotted",/)'
109    ncurvml = ncurve - 1
110
111    readf,1,wlttitle
112

```

Program Listing 2.4. Wl.sigma.pro

```

113     for nc=0,ncurvm1 do begin
114         get_lun,lun
115         ilun(nc) = lun
116         readf,1,legend
117         legnd(nc) = legend
118         readf,1,filedata
119         filedat(nc) = filedata
120     endfor
121
122     readf,1,ymin,ymax,ytcks
123     readf,1,time_axis
124
125     readf,1,crlevel
126     readf,1,range_offset
127
128
129     close,1
130
131     ;-----
132
133     ;set plot type : x, ps, or tek
134     set_plot,ptype
135
136     ;set the plot scaling
137     aspect=1.5
138     isize=1024.
139     jsize=isize*aspect
140
141     if (ptype eq 'x') then window,0,xsize=isize,ysize=jsize
142     xs=8.0
143     ys=8.0*aspect
144
145     if(ptype eq 'ps')then begin
146         if(plotype eq 'portrait')then begin
147             device, xsize=xs,$
148             ysize=ys,/inch,xoffs=0.25,yoffs=0.
149         endif
150         if(plotype eq 'landscape')then begin
151             device, ysize=10.0, /landscape,$
152             /inches, xoffs=-2.0
153         endif
154     endif
155
156     ;*****
157
158     ; Open observed wl data file and model wl data file
159
160     ; variables :
161     ;   ndatpts - number of data points
162
163
164     for nc=0,ncurvm1 do begin
165         openr,ilun(nc),filedat(nc),error=err
166         if(err ne 0) then print, !err_string
167         print,nc,filedat(nc), $
168         format='(1x,"Curve ",i2," ; file",a67)'

```

Program Listing 2.4. Wl.sigma.pro

```

169     endfor
170
171     ;-----
172
173     ; Read data from files
174
175     if(idebug eq 1)then openw,4,'time.out'
176
177
178     for nc=0,ncurvml do begin
179         if(typedata eq 'forcast')then begin
180             readf,ilun(nc),line
181             readf,ilun(nc),line
182         endif
183         ncpl1 = nc + 1
184         nlin = 1
185         nlinml = nlin - 1
186         wlevel_tot = 0.0
187         ncount = 0
188         if(idebug eq 1)then begin
189             printf,4,filedat(nc),format='(1x,a67)'
190         endif
191
192         readf,ilun(nc),time
193         print,ncpl1,time,          $
194         format='(/,1x,"file (",il,") starts at time =",f8.3)'
195         print,time,strtttime
196         point_lun,ilun(nc),0
197         if(typedata eq 'forcast')then begin
198             readf,ilun(nc),line
199             readf,ilun(nc),line
200         endif
201
202         READDATA: readf,ilun(nc),time,wlevel
203         result = EOF(ilun(nc))
204         if(idebug eq 2)then print,result
205         if(time lt strtttime)then goto, READDATA
206         if(time gt endtime)then begin
207             ncount = ncount - 1
208             ndatpts = ncount + 1
209             goto, ENDLOOP
210         endif
211         if(ncount eq 0)then begin
212             time_old = time
213             print,ncpl1,time,          $
214             format='(1x,"start time file (",il,") = ",f8.3)'
215             time_strt(nc) = time
216         endif
217
218         wlevel_tot = wlevel_tot + wlevel
219
220         if(time lt 366.0)then begin
221             jd_offset = 0.0
222         endif
223         time = time - jd_offset
224

```

Program Listing 2.4. Wl.sigma.pro

```

225     if(result lt 1)then begin
226         if(idebug eq 1)then printf,4,ncount,time
227         t(ncount) = time
228         wlplt(ncount,nc) = wlevel
229         ncount = ncount + 1
230         time_dif = time - time_old
231         if(time_dif gt hr_intrvl)then begin
232             print,format='("gap in data file")'
233             if(idebug eq 1)then begin
234                 printf,4,time_old,time,time_dif,      $
235                 format='(1x,3f9.3," gap in data file")'
236             endif
237             nlin = nlin + 1
238             nlinml = nlin - 1
239             ncg(nlinml) = 0
240             t(ncg(nlinml),nlinml) = time
241             wlplt(ncg(nlinml),nlinml) = wlevel
242             ncg(nlinml) = ncg(nlinml) + 1
243         endif
244         if(time_dif lt hr_intrvl)then begin
245             t(ncg(nlinml),nlinml) = time
246             wlplt(ncg(nlinml),nlinml) = wlevel
247             ncg(nlinml) = ncg(nlinml) + 1
248         endif
249         time_old = time
250         goto, READDATA
251     endif
252     if(result gt 0)then begin
253         if(idebug eq 1)then printf,4,ncount,time
254         print,ilun(nc),      $
255         format='(" End of file (" ,il," ) reached")'
256         t(ncount) = time
257         wlplt(ncount,nc) = wlevel
258     endif
259     close,ilun(nc)
260     ndatpts = ncount + 1
261     ENDLOOP: print,ndatpts,      $
262     format='(i4," data points, End of loop")'
263     numb_pts(nc) = ndatpts
264     endfor
265
266
267     print,ncount,format='(/,1x,i4)'
268
269     print,ndatpts,format='(1x,i4)'
270
271     !p.multi=[0,0,1]
272
273     ;-----
274
275
276     ; make the plot
277
278     !P.CHARSIZE=1.0
279
280

```

Program Listing 2.4. Wl.sigma.pro

```

281     nticks = 5
282     ncnt1 = ncg(0) - 1
283     print,ncnt1
284
285 @plot01
286     plot,t[0:ncnt1],wlplt[0:ncnt1,0],           $
287         title=wltitle,                          $
288         yrange=[ymin,ymax],                      $
289         xtitle=time_axis,                        $
290         ytitle='meters',                        $
291         xmargin=[0,0],                          $
292         ymargin=[0,0],                          $
293         xstyle=1,ystyle=1,                      $
294         linestyle=0,                            $
295         xrange=[tmin,tmax],                     $
296         xticks = nticks,                        $
297         yticks = ytcks,                         $
298         position=[0.10,0.52,0.90,0.87]
299     for nl=2,nlin do begin
300         nlm1 = nl - 1
301         ncntg(nlm1) = ncg(nlm1) - 1
302         oplot,t[0:ncntg(nlm1),nlm1],wlplt[0:ncntg(nlm1),nlm1]
303     endfor
304
305
306     xyouts,0.50,0.55,stat_name,size=1.5,/normal,alignment=0.5
307
308 ;*****
309
310 ; Draw Legend
311
312 ; Establish x,y coordinates for legend
313
314
315     x1(0,0) = 0.36
316     x1(0,1) = 0.44
317     y1(0) = 0.825
318     x1(1,0) = 0.62
319     x1(1,1) = 0.70
320     y1(1) = 0.825
321
322     xpos(0) = 0.25
323     xpos(1) = 0.49
324     ypos = 0.83
325
326     for nc=0,ncurvm1 do begin
327         xyouts,xpos(nc),ypos,legnd(nc),size=1.4,/NORMAL
328         if(nc eq 0)then linest = 0
329         if(nc eq 1)then linest = 1
330         plots,[x1(nc,0),x1(nc,1)],y1,linestyle=linest, $
331             /normal
332     endfor
333
334 ;-----
335
336 ; Draw solid lines representing low and high critical

```

Program Listing 2.4. Wl.sigma.pro

```
337 ; values for events.
338
339 xline(0) = tmin
340 xline(1) = tmax
341
342 crlevel_high(0) = crlevel + range_offset
343 crlevel_high(1) = crlevel + range_offset
344 crlevel_low(0) = -crlevel
345 crlevel_low(1) = -crlevel
346
347
348 plots,xline,crlevel_high
349 plots,xline,crlevel_low
350
351 -----
352
353 if(ptype eq 'ps') then device,/close
354
355
356 end
```

Program Listing 2.4. Wl.sigma.pro (continued)

2.5 Program Wl.multcur.pro

Wl.multcur.pro, also an IDL plot program, is an improved version of wl.sigma.pro. The program will generate plots of observed versus nowcast and observed versus predicted water levels on one page, then observed versus forecast and observed versus adjusted forecast water levels on the second page. The listing for Wl.multcur.pro is given in Program Listing 2.5.


```

1      ; Program : wl.multcur.pro
2      ;
3      ; Purpose : This program, written in the IDL programming
4      ; language, was developed to help evaluate Houston/Galveston
5      ; nowcast/forecast water level data. The program will
6      ; produce four separate plots : OBS vs. Nowcast,
7      ; OBS vs. Predicted, OBS vs. Forecast, and OBS vs. adjusted
8      ; Forecast. On each plot, the high and low critical values,
9      ; which determine events, are depicted.
10     ;
11     ; Language : IDL
12     ;
13     ; Location : /usr/people/philr/galves/NF_eval/wlevel/plot
14     ;
15     ; Version date : May 22, 2001
16     ;
17     ; Author : Phil Richardson
18
19     ;*****
20
21     im = 1500
22     nline = 12
23     ncurves = 5
24
25
26     ;Initialize character strings
27     line = ' '
28     wltitle = ' '
29     filedata = ' '
30     legend = ' '
31     cntrl_file = ' '
32     time_axis = ' '
33     stat_name = ' '
34     ptype=' '
35     plotype = ' '
36     typedata = ' '
37
38     ;Initialize integer variables
39     idebug = 0
40     ncurve = 0
41     lun = 0
42     ncleg = 0
43     ; ytcks = 0
44
45     ;Dimension arrays
46     ncg = intarr(nline)
47     ncntg = intarr(nline)
48
49     ilun = intarr(ncurves)
50     numb_pts = intarr(ncurves)
51
52     filedat = strarr(ncurves)
53     typedat = strarr(ncurves)
54     legnd = strarr(ncurves)
55     ravg = strarr(2)
56

```

Program Listing 2.5. Wl.multcur.pro

```

57     t = fltarr(im,nline)
58     wlplt = fltarr(im,nline)
59     xl=fltarr(ncurves,2)
60     yl=fltarr(2)
61     yll=fltarr(2)
62     xpos = fltarr(ncurves)
63     time_strt = fltarr(ncurves)
64     xline = fltarr(2)
65     crlevel_high = fltarr(2)
66     crlevel_low = fltarr(2)
67
68
69     small = 0.001
70     crlevel = 0.0
71     hr_intrvl = 0.09
72
73     ;*****
74
75     ; Open control file, read from control file
76
77     ;     ptype - x, ps, or tek
78     ;     ideo = 1, times (Julian dates)
79     ;     = 2, EOF result
80     ;     stat_name - station name
81     ;     tmin - start point for time (x) axis
82     ;     strttime - start time (Julian date)
83     ;     tmax - end point for time (x) axis
84     ;     endtime - end time (Julian date)
85     ;     ncurve - number of curves to plot
86     ;     legnd(nc) - character string, for legend
87     ;     filedat(nc) - data filenames
88     ;     time_axis - time axis name
89     ;     crlevel - critical value
90
91
92     print,'Enter name of control file '
93     read,cntrl_file
94     openr,1,cntrl_file
95
96     readf,1,ptype
97     if(ptype eq 'ps')then begin
98         readf,1,plotype
99     endif
100    readf,1,ideo
101    print,ideo,format='(2x,"ideo = ",i3)'
102
103    readf,1,stat_name
104    readf,1,tmin
105    readf,1,strttime
106    readf,1,tmax
107    readf,1,endtime
108    readf,1,ncurve
109    print,ncurve,format='(1x,i2," curves to be plotted",/)'
110    ncurvml = ncurve - 1
111
112    readf,1,wltitle

```

Program Listing 2.5. Wl.multcur.pro (continued)

```

113
114     for nc=0,ncurvm1 do begin
115         get_lun,lun
116         ilun(nc) = lun
117         readf,1,legend
118         legnd(nc) = legend
119         readf,1,typedata
120         readf,1,filedata
121         typedat(nc) = typedata
122         filedat(nc) = filedata
123     endfor
124
125     readf,1,ymin,ymax,ytcks
126     readf,1,time_axis
127
128     readf,1,crlevel
129     readf,1,range_offset
130
131
132     close,1
133
134 ;-----
135
136 ;set plot type : x, ps, or tek
137     set_plot,ptype
138
139 ;set the plot scaling
140     aspect=1.5
141     isize=1024.
142     jsize=isize*aspect
143
144     if (ptype eq 'x') then window,0,xsize=isize,ysize=jsize
145     xs=8.0
146     ys=8.0*aspect
147
148     if(ptype eq 'ps')then begin
149         if(plotttype eq 'portrait')then begin
150             device, xsize=xs,$
151                 ysize=ys,/inch,xoffs=0.25,yoffs=0.
152         endif
153         if(plotttype eq 'landscape')then begin
154             device, ysize=10.0, /landscape,$
155                 /inches, xoffs=-2.0
156         endif
157     endif
158
159 ;*****
160
161 ; Open observed wl data file and model wl data file
162
163 ; variables :
164 ;   ndatpts - number of data points
165
166
167     for nc=0,ncurvm1 do begin
168         openr,ilun(nc),filedat(nc),error=err

```

Program Listing 2.5. Wl.multcur.pro (continued)

```

169         if(err ne 0) then print, !err_string
170         print,nc,filedat(nc),      $
171             format='(lx,"Curve ",i2," ; file",a67)'  

172     endfor  

173  

174     ;-----  

175  

176     ; Read data from files  

177  

178     if(idebug eq 1)then openw,4,'time.out'  

179  

180  

181     for nc=0,ncurvm1 do begin  

182         if(typedat(nc) eq 'forcast')then begin  

183             readf,ilun(nc),line  

184             readf,ilun(nc),line  

185         endif  

186         ncpl1 = nc + 1  

187         nlin = 1  

188         nlinm1 = nlin - 1  

189         wlevel_tot = 0.0  

190         ncount = 0  

191         if(idebug eq 1)then begin  

192             printf,4,filedat(nc),format='(lx,a67)'  

193         endif  

194  

195         readf,ilun(nc),time  

196         print,ncpl1,time,      $  

197         format='(/,lx,"file (",i1,") starts at time =",f8.3)'  

198         print,time,strttime  

199         point_lun,ilun(nc),0  

200         if(typedat(nc) eq 'forcast')then begin  

201             readf,ilun(nc),line  

202             readf,ilun(nc),line  

203         endif  

204  

205         READDATA: readf,ilun(nc),time,wlevel  

206         result = EOF(ilun(nc))  

207         if(idebug eq 2)then print,result  

208         if(time lt strttime)then goto, READDATA  

209         if(time gt endtime)then begin  

210             ncount = ncount - 1  

211             ndatpts = ncount + 1  

212             goto, ENDLOOP  

213         endif  

214         if(ncount eq 0)then begin  

215             time_old = time  

216             print,ncpl1,time,      $  

217             format='(lx,"start time file (",i1,") = ",f8.3)'  

218             time_strt(nc) = time  

219         endif  

220  

221         wlevel_tot = wlevel_tot + wlevel  

222  

223         if(time lt 366.0)then begin  

224             jd_offset = 0.0

```

Program Listing 2.5. Wl.multcur.pro (continued)

```

225     endif
226     time = time - jd_offset
227
228     if(result lt 1)then begin
229         if(idebug eq 1)then printf,4,ncount,time
230         t(ncount,nc) = time
231         wlplt(ncount,nc) = wlevel
232         ncount = ncount + 1
233         time_dif = time - time_old
234         if(time_dif gt hr_intrvl)then begin
235             print,format='("gap in data file")'
236             if(idebug eq 1)then begin
237                 printf,4,time_old,time,time_dif,      $
238                 format='(lx,3f9.3," gap in data file")'
239             endif
240             nlin = nlin + 1
241             nlinml = nlin - 1
242             ncg(nlinml) = 0
243             t(ncg(nlinml),nlinml) = time
244             wlplt(ncg(nlinml),nlinml) = wlevel
245             ncg(nlinml) = ncg(nlinml) + 1
246         endif
247     ;     if(time_dif lt hr_intrvl)then begin
248     ;         t(ncg(nlinml),nlinml) = time
249     ;         wlplt(ncg(nlinml),nlinml) = wlevel
250     ;         wlplt(ncount,nc) = wlevel
251     ;         ncg(nlinml) = ncg(nlinml) + 1
252     ;     endif
253     time_old = time
254     goto, READDATA
255     endif
256     if(result gt 0)then begin
257         if(idebug eq 1)then printf,4,ncount,time
258         print,ilun(nc),      $
259         format='(" End of file (" ,il,") reached")'
260         t(ncount,nc) = time
261         wlplt(ncount,nc) = wlevel
262     endif
263     close,ilun(nc)
264     ndatpts = ncount + 1
265     ENDLLOOP: print,ndatpts,      $
266     format='(i4," data points, End of loop")'
267     numb_pts(nc) = ndatpts - 1
268     endfor
269
270
271
272     !p.multi=[0,0,1]
273
274     ;*****
275
276     ; make the plot
277
278     !P.CHARSIZE=1.0
279
280     ; Variables :

```

Program Listing 2.5. Wl.multcur.pro (continued)

```

281 ;
282 ;   yt1 - Y coordinate (window) of top plot
283 ;   yb1 - Y coordinate (window) of bottom of top plot
284 ; ystnmT - Y coordinate of station name (top plot)
285 ; ystnmB - Y coordinate of station name (bottom plot)
286 ; ypostT - Y coordinate of legend (top plot)
287 ; ypostB - Y coordinate of legend (bottom plot)
288 ;   y1 - Y coordinate of legend (line), top plot
289 ;   y11 - Y coordinate of legend (line), bottom plot
290
291 ; Establish x,y coordinates for legend
292
293     xl(0,0) = 0.36
294     xl(0,1) = 0.44
295 ; y1(0) = 0.91
296     y1(0) = 0.71
297     xl(1,0) = 0.62
298     xl(1,1) = 0.70
299     y1(1) = 0.71
300     y11(0) = 0.33
301     y11(1) = 0.33
302
303     xpos(0) = 0.25
304     xpos(1) = 0.49
305     ypostT = 0.71
306     ypostB = 0.33
307
308
309     xline(0) = tmin
310     xline(1) = tmax
311
312     crlevel_high(0) = crlevel + range_offset
313     crlevel_high(1) = crlevel + range_offset
314     crlevel_low(0) = -crlevel
315     crlevel_low(1) = -crlevel
316
317
318     nticks = 5
319     ncnt1 = numb_pts(0)
320     ncnt2 = numb_pts(1)
321     ncnt3 = numb_pts(2)
322     ncnt4 = numb_pts(3)
323     ncnt5 = numb_pts(4)
324     print,ncnt1
325     print,ncnt2
326     print,ncnt3
327     print,ncnt4
328     print,ncnt5
329
330 ;-----
331
332 ; 2 plots per page -
333
334 !P.Multi = [0,1,2,0,0]
335
336

```

Program Listing 2.5. Wl.multcur.pro (continued)

```

337 ; OBS vs. Nowcast
338
339 @plot01
340
341     yt1 = 0.95
342     yb1 = 0.68
343     ystnmT = 0.70
344
345     plot,t[0:ncnt1,0],wlplt[0:ncnt1,0],           $
346         title=wltitle,                           $
347         yrange=[ymin,ymax],                       $
348         xtitle=time_axis,                         $
349         ytitle='meters',                          $
350         xmargin=[0,0],                            $
351         ymargin=[0,0],                            $
352         xstyle=1,ystyle=1,                        $
353         linestyle=0,                              $
354         xrange=[tmin,tmax],                      $
355         xticks = nticks,                          $
356         yticks = ytcks,                           $
357         position=[0.10,yb1,0.90,yt1]
358     oplot,t[0:ncnt2,1],wlplt[0:ncnt2,1],         $
359         linestyle=2
360
361
362 ; xyouts,0.50,ystnmT,stat_name,size=1.5,/normal,alignment=0.5
363
364     for nc=0,1 do begin
365         xyouts,xpos(nc),yposT,legnd(nc),size=1.4,/NORMAL
366         if(nc eq 0)then linest = 0
367         if(nc eq 1)then linest = 1
368         plots,[x1(nc,0),x1(nc,1)],y1,linestyle=linest, $
369             /NORMAL
370     endfor
371
372     plots,xline,crlevel_high
373     plots,xline,crlevel_low
374
375 ; -----
376
377 ; OBS vs. Predicted
378
379     yt2 = 0.57
380     yb2 = 0.30
381     ystnmB = 0.32
382
383     plot,t[0:ncnt1,0],wlplt[0:ncnt1,0],           $
384         title=wltitle,                           $
385         yrange=[ymin,ymax],                       $
386         xtitle=time_axis,                         $
387         ytitle='meters',                          $
388         xmargin=[0,0],                            $
389         ymargin=[0,0],                            $
390         xstyle=1,ystyle=1,                        $
391         linestyle=0,                              $
392         xrange=[tmin,tmax],                      $

```

Program Listing 2.5. Wl.multcur.pro (continued)

```

393         xticks = nticks,           $
394         yticks = ytcks,           $
395         position=[0.10,yb2,0.90,yt2]
396         oplot,t[0:ncnt3,2],wlplt[0:ncnt3,2],      $
397         linestyle=2
398
399     ; xyouts,0.50,ystnmB,stat_name,size=1.5,/normal,alignment=0.5
400
401
402     ; Draw Legend
403
404     for nc=0,1 do begin
405         ncleg = nc * 2
406         xyouts,xpos(nc),yposB,legnd(ncleg),size=1.4,/NORMAL
407         if(nc eq 0)then linest = 0
408         if(nc eq 1)then linest = 1
409         plots,[x1(nc,0),x1(nc,1)],y11,linestyle=linest,  $
410         /normal
411     endfor
412
413
414     ; Draw solid lines representing low and high critical
415     ; values for events.
416
417
418     plots,xline,crlevel_high
419     plots,xline,crlevel_low
420
421     ;-----
422
423     ; OBS vs. Forecast
424
425     plot,t[0:ncnt1,0],wlplt[0:ncnt1,0],           $
426         title=wttitle,                           $
427         yrange=[ymin,ymax],                       $
428         xtitle=time_axis,                         $
429         ytitle='meters',                          $
430         xmargin=[0,0],                            $
431         ymargin=[0,0],                            $
432         xstyle=1,ystyle=1,                        $
433         linestyle=0,                              $
434         xrange=[tmin,tmax],                       $
435         xticks=nticks,                            $
436         yticks=ytcks,                             $
437         position=[0.10,yb1,0.90,yt1]
438         oplot,t[0:ncnt4,3],wlplt[0:ncnt4,3],      $
439         linestyle=2
440
441     ; xyouts,0.50,ystnmT,stat_name,size=1.5,/normal,alignment=0.5
442
443     for nc=0,1 do begin
444         ncleg = nc * 3
445         xyouts,xpos(nc),yposT,legnd(ncleg),size=1.4,/NORMAL
446         if(nc eq 0)then linest = 0
447         if(nc eq 1)then linest = 1
448         plots,[x1(nc,0),x1(nc,1)],y1,linestyle=linest,  $

```

Program Listing 2.5. Wl.multcur.pro (continued)


```

449         /normal
450     endfor
451
452     plots,xline,crlevel_high
453     plots,xline,crlevel_low
454
455 ;-----
456
457 ; OBS vs. adjusted Forecast
458
459     plot,t[0:ncnt1,0],wlplt[0:ncnt1,0],           $
460         title=wttitle,                           $
461         yrange=[ymin,ymax],                       $
462         xtitle=time_axis,                         $
463         ytitle='meters',                          $
464         xmargin=[0,0],                            $
465         ymargin=[0,0],                            $
466         xstyle=1,ystyle=1,                        $
467         linestyle=0,                              $
468         xrange=[tmin,tmax],                       $
469         xticks=nticks,                            $
470         yticks=ytcks,                             $
471         position=[0.10,yb2,0.90,yt2]
472         oplot,t[0:ncnt5,4],wlplt[0:ncnt5,4],     $
473         linestyle=2
474
475 ; xyouts,0.50,ystnmB,stat_name,size=1.5,/normal,alignment=0.5
476
477     for nc=0,1 do begin
478         ncleg = nc * 4
479         xyouts,xpos(nc),yposB,legnd(ncleg),size=1.4,/NORMAL
480         if(nc eq 0)then linest = 0
481         if(nc eq 1)then linest = 1
482         plots,[xl(nc,0),xl(nc,1)],yll,linestyle=linest, $
483         /normal
484     endfor
485
486     plots,xline,crlevel_high
487     plots,xline,crlevel_low
488
489 ;*****
490
491     spawn, 'lp -dqms2 idl.ps'
492
493 ;*****
494
495     if(ptype eq 'ps') then device,/close
496
497     ENDPROG:
498
499     end

```

Program Listing 2.5. Wl.multcur.pro (continued)

3. APRIL 2000 SAMPLE APPLICATION

To perform the water level event analysis, each program is executed in the order shown in Table 3.1. For each program a separate directory is recommended as shown below, where ~ designates the users home area, galves the project directory, and NF_eval/wlevel the nowcast/forecast water level event evaluation directory.

```
~/galves/NF_eval/wlevel/observed/reform_coops.f
~/galves/NF_eval/wlevel/read_nowforc.f
~/galves/NF_eval/wlevel/sa.nowforc/match.event.f
~/galves/NF_eval/wlevel/plot/wl.sigma.pro
~/galves/NF_eval/wlevel/plot/wl.multcur.pro
```

Table 3.1. Job Control, Source File, and Control File Inventory

Job Control	Source File	Control File
reform.jcl	reform_coops.f	reform.n
readnf.jcl	read_nowforc.f	readnf.apr00.n
match.jcl	match.event.f	match_forc.apr00.n
	wl.sigma.pro	cntrl.pleas
	wl.multcur.pro	c.pleas_apr00

Listings for jcl files and control files are provided in Appendix A. The two IDL plot programs do not have job control files. To run the IDL programs, type `idl <return>`, then type `.r filename.pro <return>`.

Program output files for the April 2000 evaluation of the forecast water levels at Galveston Pleasure Pier in lower Galveston Bay and at Morgans Point in upper Galveston Bay are given. For this application an observed water level exceeding MHHW by 10 cm is considered a high water event, while an observed water level less than MLLW by 10 cm is considered a low water event. A linear regression (gain and bias) is used to adjust the forecast.

Three tables (table2.out, table_high, and table_low) generated by match.event.f are included for two cases of bias and gain. In Table 3.2, results for bias = 0. and gain = 1.0 corresponding to the original forecast are given. Results for the adjusted forecast are presented in Table 3.3. The adjustment at Pleasure Pier is bias = -0.05 and gain = 1.05. The adjustment at Morgans Point is bias = -0.075 and gain = 1.05. A gain of 1.05 results in the amplification of the water levels by five percent. Tables 3.2 and 3.3 indicate that the adjustment (bias and gain) does improve forecast results. For both stations, the number of successes increases while the number of failures remain constant or decrease. The adjustment does cause an increase (by two) in the number of false alarms at Pleasure Pier. However, the number of successes increases by two and the number of failures decreases by three. Overall, the adjustment improves the forecast at both stations. This is further demonstrated by viewing the plots generated by wl.sigma.pro at Pleasure Pier in Figure 3.1 and at Morgans Point in Figure 3.2.

The plots in Figures 3.3 and 3.4 were generated by wl.multcur.pro.

Table 3.2a. Table2.out for Case One Bias=0.0 and Gain=1.0

Observed wl data vs. forecast wl data, event analysis
forecast hours 1 - 24

April 2000

Start time of comparison = 92.000, End time = 122.000
critical level = 0.10

Pleasure Pier

bias = 0.000, gain = 1.00

Low water critical level, -0.100
High water critical level, 0.749

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	1	-3	4	-1	0.8346	0.8140	0.021	92.625
2	5	0	1	0	-0.1531	-0.1340	-0.019	95.833
3	7	2	-3	0	-0.1079	-0.2250	0.117	99.958

mean difference of peak water levels for
"success" forecasts is 0.0523 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	2	0	0.0366	-0.3040	0.341	94.917
2	3	1	-0.0467	-0.1120	0.065	95.250
3	4	-1	-0.0465	-0.1470	0.101	95.417
4	6	-1	-0.0350	-0.1070	0.072	99.000
5	12	0	0.0151	-0.1550	0.170	111.917
6	15	0	0.6252	0.7880	-0.163	121.542

mean difference of peak water levels for
"failure" forecasts is 0.1519 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	8	0	0.7871	0.7480	0.039	100.375
2	9	0	0.8129	0.6130	0.200	102.417
3	10	0	-0.1851	-0.0690	-0.116	104.167
4	11	-1	0.8596	0.6630	0.197	111.208
5	13	0	-0.1250	-0.0990	-0.026	112.958
6	14	0	0.7631	0.7070	0.056	114.333

mean difference of peak water levels for
"false" forecasts is 0.1056 meters

Success	Failure	false
3	6	6

Table3.2a. Table2.out for Case One Bias=0.0 and Gain=1.0

Morgans Pt
 bias = 0.00, gain = 1.00
 Low water critical level, -0.100
 High water critical level, 0.496

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	1	0	3	2	0.6674	0.5710	0.096	92.083
2	3	6	-4	1	-0.2529	-0.5110	0.258	95.500
3	6	4	-6	-1	-0.1145	-0.2260	0.111	100.208
4	8	0	2	-1	0.5202	0.5190	0.001	100.875
5	9	0	2	6	0.5889	0.6090	-0.020	101.708
6	12	1	0	1	-0.1938	-0.2170	0.023	104.375
7	16	1	-2	0	-0.1576	-0.2100	0.052	113.167
8	17	2	-1	1	0.5017	0.5170	-0.015	113.875
9	18	-1	2	0	0.7018	0.6860	0.016	114.750
10	19	1	-1	1	-0.1353	-0.2030	0.068	116.250

mean difference of peak water levels for
 "success" forecasts is 0.0662 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	4	-2	0.0119	-0.2990	0.311	99.292
2	5	0	0.0147	-0.1080	0.123	99.792
3	11	0	-0.0428	-0.1880	0.145	103.375
4	13	0	-0.0192	-0.1700	0.151	105.417
5	15	-2	-0.0651	-0.2420	0.177	112.208
6	21	0	0.4933	0.6160	-0.123	121.708

mean difference of peak water levels for
 "failure" forecasts is 0.1715 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	2	0	0.6118	0.4650	0.147	92.708
2	7	0	0.5055	0.4750	0.031	100.708
3	10	0	0.6557	0.4220	0.234	102.708
4	14	0	0.6356	0.3730	0.263	111.417
5	20	0	0.4962	0.4230	0.073	121.167

mean difference of peak water levels for
 "false" forecasts is 0.1494 meters

Success	Failure	false
10	6	5

Table 3.2b. Table_high for Case One Bias=0.0 and Gain=1.0

Observed wl data vs. forecast wl data, event analysis
forecast hours 1 - 24

April 2000

Start time of comparison = 92.000, End time = 122.000

critical level = 0.10

high water events

Pleasure Pier

bias = 0.000, gain = 1.00

High water critical level, 0.749

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	1	-3	4	-1	0.8346	0.8140	0.021	92.625

for 1 high water events, mean difference
of the peak water levels is 0.0206 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
6	15	0	0.6252	0.7880	-0.163	121.542

for 1 high water events, mean difference
of the peak water levels is 0.1628 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	8	0	0.7871	0.7480	0.039	100.375
2	9	0	0.8129	0.6130	0.200	102.417
4	11	-1	0.8596	0.6630	0.197	111.208
6	14	0	0.7631	0.7070	0.056	114.333

for 4 high water events, mean difference
of the peak water levels is 0.1229 meters

Table 3.2b. Table_high for Case One Bias=0.0 and Gain=1.0

Morgans Pt
 bias = 0.00, gain = 1.00
 High water critical level, 0.496

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	1	0	3	2	0.6674	0.5710	0.096	92.083
4	8	0	2	-1	0.5202	0.5190	0.001	100.875
5	9	0	2	6	0.5889	0.6090	-0.020	101.708
8	17	2	-1	1	0.5017	0.5170	-0.015	113.875
9	18	-1	2	0	0.7018	0.6860	0.016	114.750

for 5 high water events, mean difference of the peak water levels is 0.0298 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
6	21	0	0.4933	0.6160	-0.123	121.708

for 1 high water events, mean difference of the peak water levels is 0.1227 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	2	0	0.6118	0.4650	0.147	92.708
2	7	0	0.5055	0.4750	0.031	100.708
3	10	0	0.6557	0.4220	0.234	102.708
4	14	0	0.6356	0.3730	0.263	111.417
5	20	0	0.4962	0.4230	0.073	121.167

for 5 high water events, mean difference of the peak water levels is 0.1494 meters

Table 3.2c. Table_low for Case One Bias= 0.0 and Gain=1.0

Observed wl data vs. forecast wl data, event analysis
forecast hours 1 - 24

April 2000

Start time of comparison = 92.000, End time = 122.000

critical level = 0.10

low water events

Pleasure Pier

bias = 0.000, gain = 1.00

Low water critical level, -0.100

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
2	5	0	1	0	-0.1531	-0.1340	-0.019	95.833
3	7	2	-3	0	-0.1079	-0.2250	0.117	99.958

for 2 low water events, mean difference
of the peak water levels is 0.0681 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	2	0	0.0366	-0.3040	0.341	94.917
2	3	1	-0.0467	-0.1120	0.065	95.250
3	4	-1	-0.0465	-0.1470	0.101	95.417
4	6	-1	-0.0350	-0.1070	0.072	99.000
5	12	0	0.0151	-0.1550	0.170	111.917

for 5 low water events, mean difference
of the peak water levels is 0.1497 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
3	10	0	-0.1851	-0.0690	-0.116	104.167
5	13	0	-0.1250	-0.0990	-0.026	112.958

for 2 low water events, mean difference
of the peak water levels is 0.0711 meters

Table 3.2c. Table_low for Case One Bias= 0.0 and Gain=1.0

Morgans Pt
 bias = 0.00, gain = 1.00
 Low water critical level, -0.100

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
2	3	6	-4	1	-0.2529	-0.5110	0.258	95.500
3	6	4	-6	-1	-0.1145	-0.2260	0.111	100.208
6	12	1	0	1	-0.1938	-0.2170	0.023	104.375
7	16	1	-2	0	-0.1576	-0.2100	0.052	113.167
10	19	1	-1	1	-0.1353	-0.2030	0.068	116.250

for 5 low water events, mean difference
 of the peak water levels is 0.1026 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	4	-2	0.0119	-0.2990	0.311	99.292
2	5	0	0.0147	-0.1080	0.123	99.792
3	11	0	-0.0428	-0.1880	0.145	103.375
4	13	0	-0.0192	-0.1700	0.151	105.417
5	15	-2	-0.0651	-0.2420	0.177	112.208

for 5 low water events, mean difference
 of the peak water levels is 0.1813 meters

Table 3.3a. Table2.out for Case Two Bias=-0.05 and Gain=1.05

Observed wl data vs. forecast wl data, event analysis
forecast hours 1 - 24

April 2000

Start time of comparison = 92.000, End time = 122.000

critical level = 0.10

Pleasure Pier

bias = -0.050, gain = 1.05

Low water critical level, -0.100

High water critical level, 0.749

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	1	-2	2	-1	0.8082	0.8140	-0.006	92.625
2	3	1	-3	-3	-0.1171	-0.1470	0.030	95.417
3	4	-1	3	0	-0.2289	-0.1340	-0.095	95.833
4	6	1	-2	-1	-0.1049	-0.1070	0.002	99.000
5	7	2	-2	0	-0.1814	-0.2250	0.044	99.958

mean difference of peak water levels for
"success" forecasts is 0.0353 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	2	0	-0.0297	-0.3040	0.274	94.917
2	13	0	-0.0522	-0.1550	0.103	111.917
3	16	0	0.5884	0.7880	-0.200	121.542

mean difference of peak water levels for
"failure" forecasts is 0.1922 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	5	0	-0.1508	-0.0430	-0.108	97.917
2	8	0	0.7584	0.7480	0.010	100.375
3	9	0	0.7854	0.6130	0.172	102.417
4	10	-1	-0.1596	-0.0460	-0.114	103.167
5	11	0	-0.2625	-0.0690	-0.193	104.167
6	12	-1	0.8345	0.6630	0.172	111.208
7	14	0	-0.1993	-0.0990	-0.100	112.958
8	15	0	-0.1418	-0.0120	-0.130	116.042

mean difference of peak water levels for
"false" forecasts is 0.1249 meters

Success	Failure	false
5	3	8

Table 3.3a. Table2.out for Case Two Bias=-0.075 and Gain=1.05

Morgans Pt

bias = -0.075, gain = 1.05

Low water critical level, -0.100

High water critical level, 0.496

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	1	0	2	2	0.6159	0.5710	0.045	92.083
2	3	3	0	1	-0.3504	-0.5110	0.161	95.500
3	6	2	-2	-1	-0.2051	-0.2260	0.021	100.208
4	9	2	-1	6	0.5334	0.6090	-0.076	101.708
5	11	-1	2	0	-0.1298	-0.1880	0.058	103.375
6	12	-2	4	1	-0.2884	-0.2170	-0.071	104.375
7	13	1	-3	0	-0.1051	-0.1700	0.065	105.417
8	15	1	-4	-2	-0.1533	-0.2420	0.089	112.208
9	16	-1	1	0	-0.2504	-0.2100	-0.040	113.167
10	18	1	-1	0	0.6520	0.6860	-0.034	114.750
11	19	-1	2	1	-0.2270	-0.2030	-0.024	116.250

mean difference of peak water levels for "success" forecasts is 0.0621 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	4	-2	-0.0724	-0.2990	0.227	99.292
2	5	0	-0.0695	-0.1080	0.039	99.792
3	7	0	0.4498	0.4970	-0.047	100.792
4	8	0	0.4521	0.5190	-0.067	100.875
5	17	0	0.4340	0.5170	-0.083	113.875
6	20	0	0.4331	0.6160	-0.183	121.708

mean difference of peak water levels for "failure" forecasts is 0.1075 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	2	0	0.5575	0.4650	0.093	92.708
2	10	0	0.6036	0.4220	0.182	102.708
3	14	0	0.5825	0.3730	0.209	111.417

mean difference of peak water levels for "false" forecasts is 0.1612 meters

Success	Failure	false
11	6	3

Table 3.3b. Table_high for Case Two Bias=-0.05 and Gain=1.05

Observed wl data vs. forecast wl data, event analysis
forecast hours 1 - 24

April 2000

Start time of comparison = 92.000, End time = 122.000

critical level = 0.10

high water events

Pleasure Pier

bias = -0.050, gain = 1.05

High water critical level, 0.749

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	1	-2	2	-1	0.8082	0.8140	-0.006	92.625

for 1 high water events, mean difference
of the peak water levels is 0.0058 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
3	16	0	0.5884	0.7880	-0.200	121.542

for 1 high water events, mean difference
of the peak water levels is 0.1996 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
2	8	0	0.7584	0.7480	0.010	100.375
3	9	0	0.7854	0.6130	0.172	102.417
6	12	-1	0.8345	0.6630	0.172	111.208

for 3 high water events, mean difference
of the peak water levels is 0.1181 meters

Table 3.3b. Table_high for Case Two Bias=-0.075 and Gain=1.05

Morgans Pt
 bias = -0.075, gain = 1.05
 High water critical level, 0.496

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	1	0	2	2	0.6159	0.5710	0.045	92.083
4	9	2	-1	6	0.5334	0.6090	-0.076	101.708
10	18	1	-1	0	0.6520	0.6860	-0.034	114.750

for 3 high water events, mean difference of the peak water levels is 0.0515 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
3	7	0	0.4498	0.4970	-0.047	100.792
4	8	0	0.4521	0.5190	-0.067	100.875
5	17	0	0.4340	0.5170	-0.083	113.875
6	20	0	0.4331	0.6160	-0.183	121.708

for 4 high water events, mean difference of the peak water levels is 0.0950 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	2	0	0.5575	0.4650	0.093	92.708
2	10	0	0.6036	0.4220	0.182	102.708
3	14	0	0.5825	0.3730	0.209	111.417

for 3 high water events, mean difference of the peak water levels is 0.1612 meters

Table 3.3c. Table_low for Case Two Bias=-0.05 and Gain=1.05

Observed wl data vs. forecast wl data, event analysis
forecast hours 1 - 24

April 2000

Start time of comparison = 92.000, End time = 122.000

critical level = 0.10

low water events

Pleasure Pier

bias = -0.050, gain = 1.05

Low water critical level, -0.100

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
2	3	1	-3	-3	-0.1171	-0.1470	0.030	95.417
3	4	-1	3	0	-0.2289	-0.1340	-0.095	95.833
4	6	1	-2	-1	-0.1049	-0.1070	0.002	99.000
5	7	2	-2	0	-0.1814	-0.2250	0.044	99.958

for 4 low water events, mean difference
of the peak water levels is 0.0426 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	2	0	-0.0297	-0.3040	0.274	94.917
2	13	0	-0.0522	-0.1550	0.103	111.917

for 2 low water events, mean difference
of the peak water levels is 0.1885 meters

false number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	5	0	-0.1508	-0.0430	-0.108	97.917
4	10	-1	-0.1596	-0.0460	-0.114	103.167
5	11	0	-0.2625	-0.0690	-0.193	104.167
7	14	0	-0.1993	-0.0990	-0.100	112.958
8	15	0	-0.1418	-0.0120	-0.130	116.042

for 5 low water events, mean difference
of the peak water levels is 0.1290 meters

Table 3.3c. Table_low for Case Two Bias=-0.075 and Gain=1.05

Morgans Pt

bias = -0.075, gain = 1.05

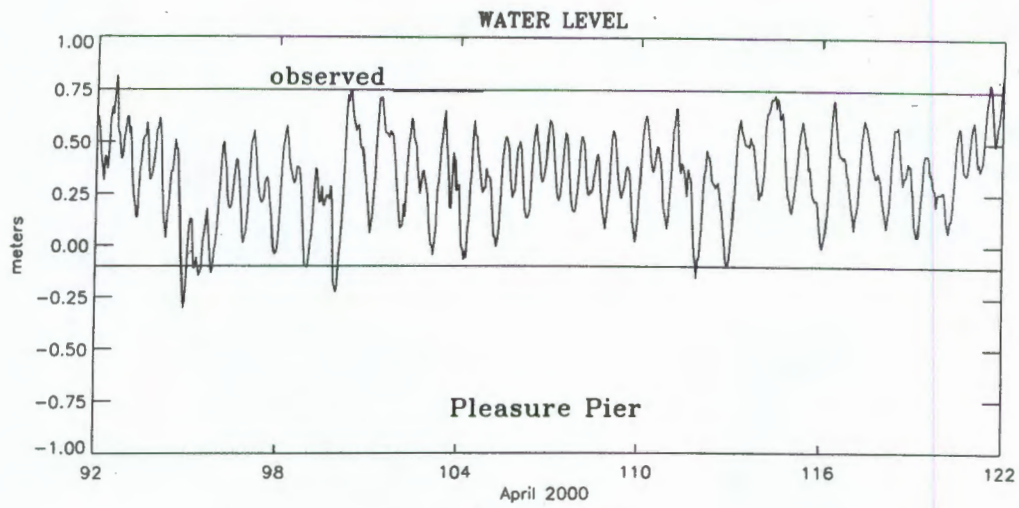
Low water critical level, -0.100

success number	event number	dstart time	delta duration	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
2	3	3	0	1	-0.3504	-0.5110	0.161	95.500
3	6	2	-2	-1	-0.2051	-0.2260	0.021	100.208
5	11	-1	2	0	-0.1298	-0.1880	0.058	103.375
6	12	-2	4	1	-0.2884	-0.2170	-0.071	104.375
7	13	1	-3	0	-0.1051	-0.1700	0.065	105.417
8	15	1	-4	-2	-0.1533	-0.2420	0.089	112.208
9	16	-1	1	0	-0.2504	-0.2100	-0.040	113.167
11	19	-1	2	1	-0.2270	-0.2030	-0.024	116.250

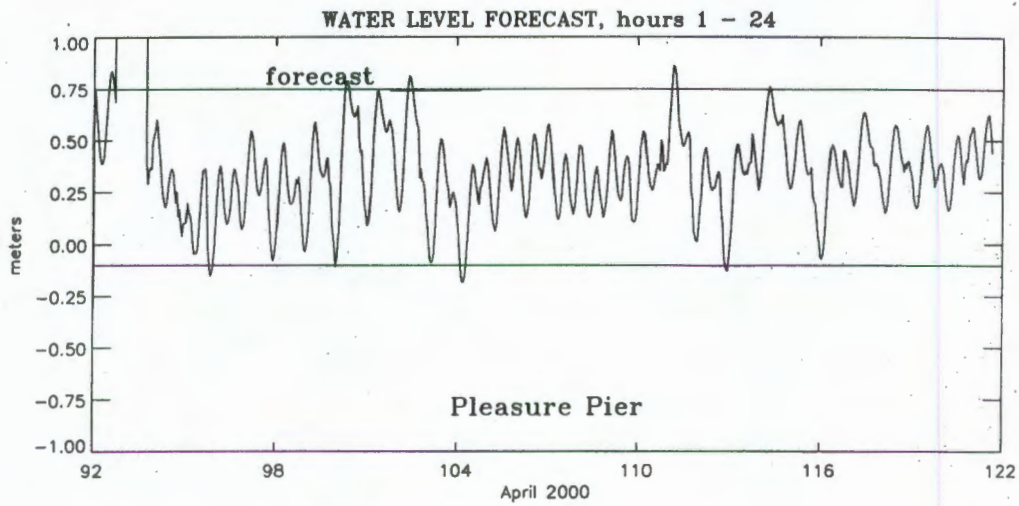
for 8 low water events, mean difference of the peak water levels is 0.0661 meters

failure number	event number	dpeak time	forecast wl(m)	observed wl(m)	dwl peak(m)	obs peak time(jd)
1	4	-2	-0.0724	-0.2990	0.227	99.292
2	5	0	-0.0695	-0.1080	0.039	99.792

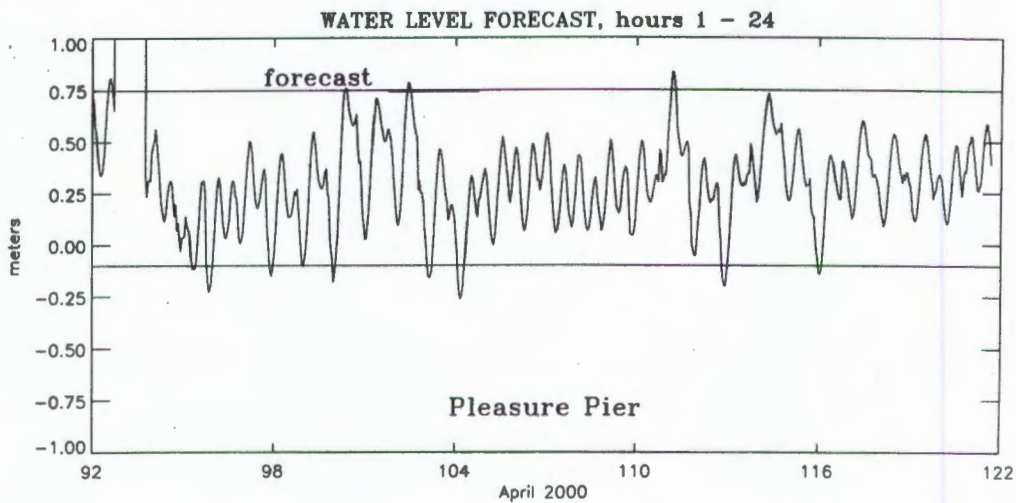
for 2 low water events, mean difference of the peak water levels is 0.1326 meters



Observations

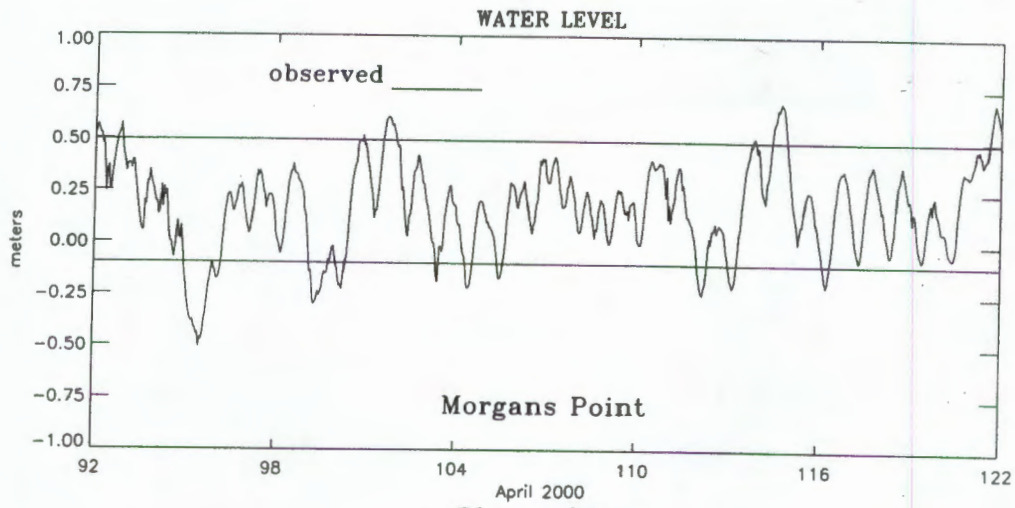


Case One, Bias=0.0, Gain=1.0

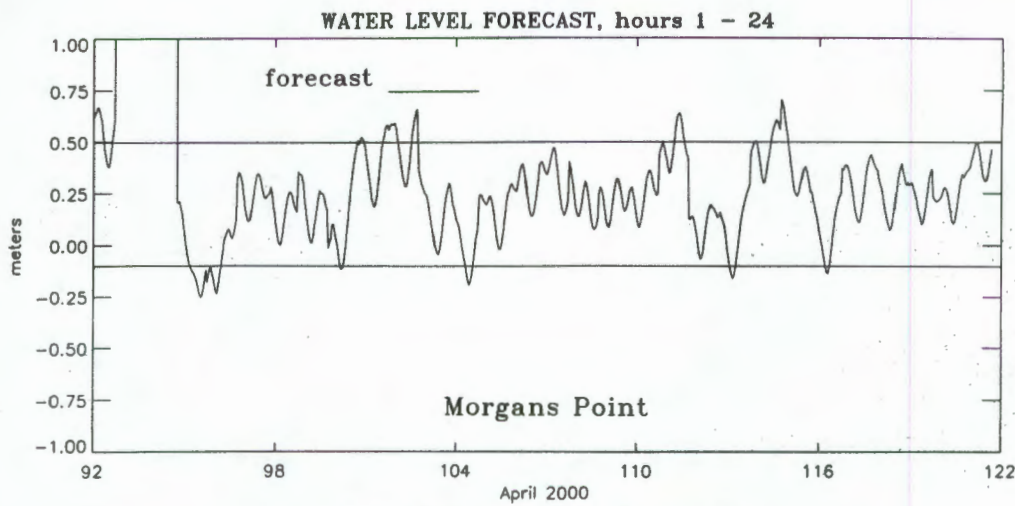


Case Two, Bias=-0.05, Gain=1.05

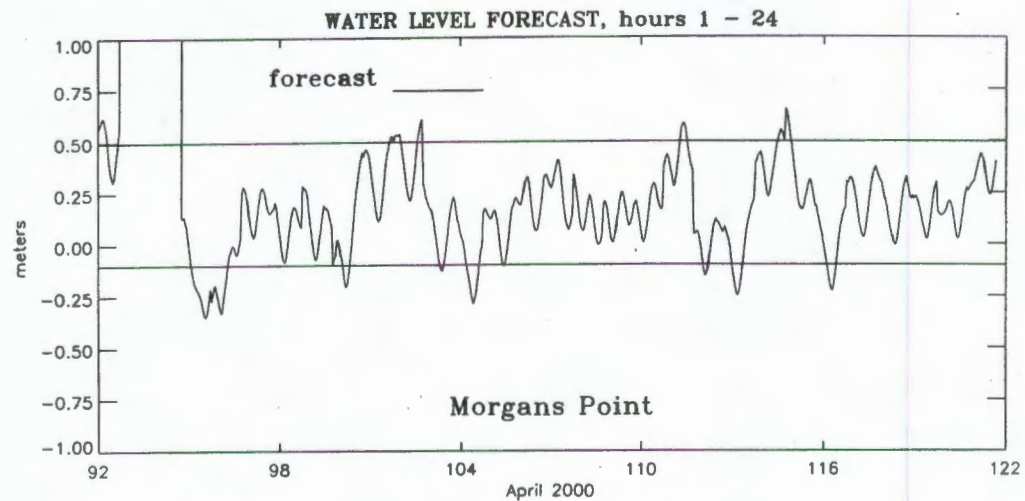
**Figure 3.1. Event Analysis Plots at Galveston Pleasure Pier for April 2000:
Observed, Forecast, and Adjusted Forecast**



Observations



Case One, Bias=0.0, Gain=1.0



Case Two, Bias=-0.075, Gain=1.05

**Figure 3.2. Event Analysis Plots at Morgans Point for April 2000:
Observed, Forecast, and Adjusted Forecast**

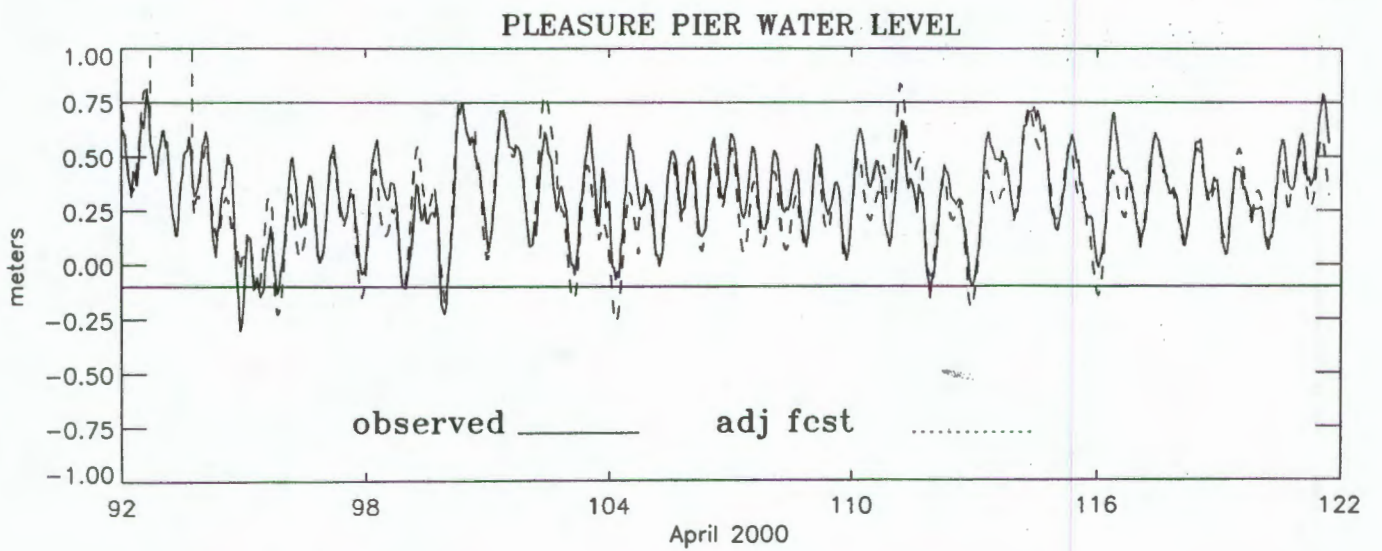
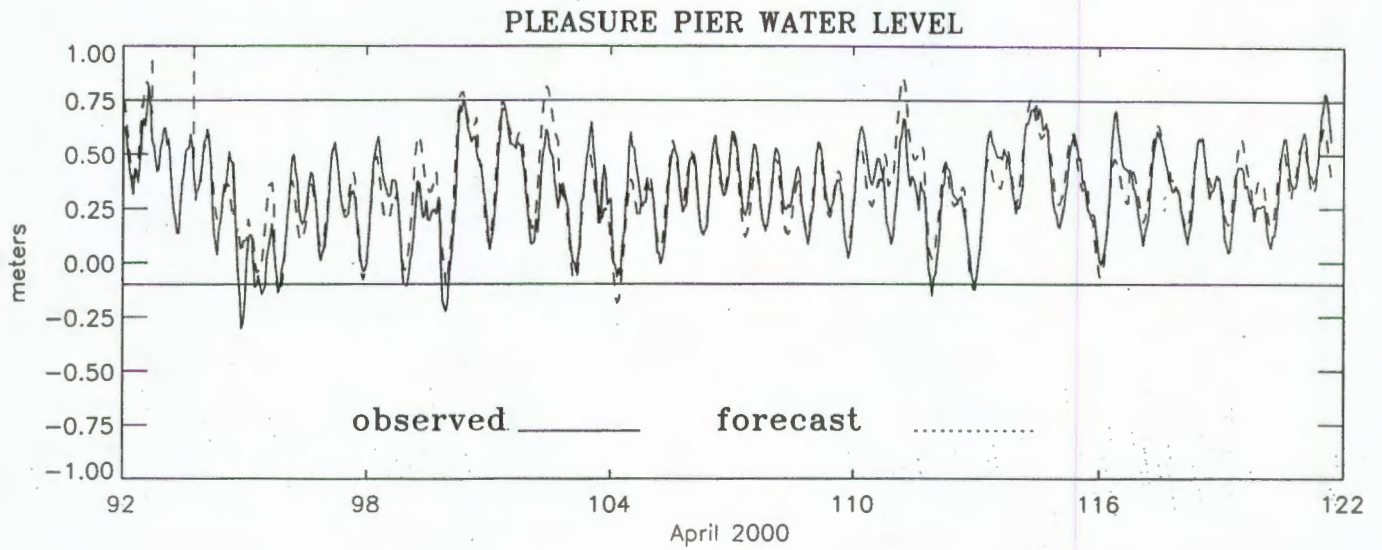


Figure 3.3. Event Analysis Plots at Galveston Pleasure Pier for April 2000:
Forecast and Adjusted Forecast

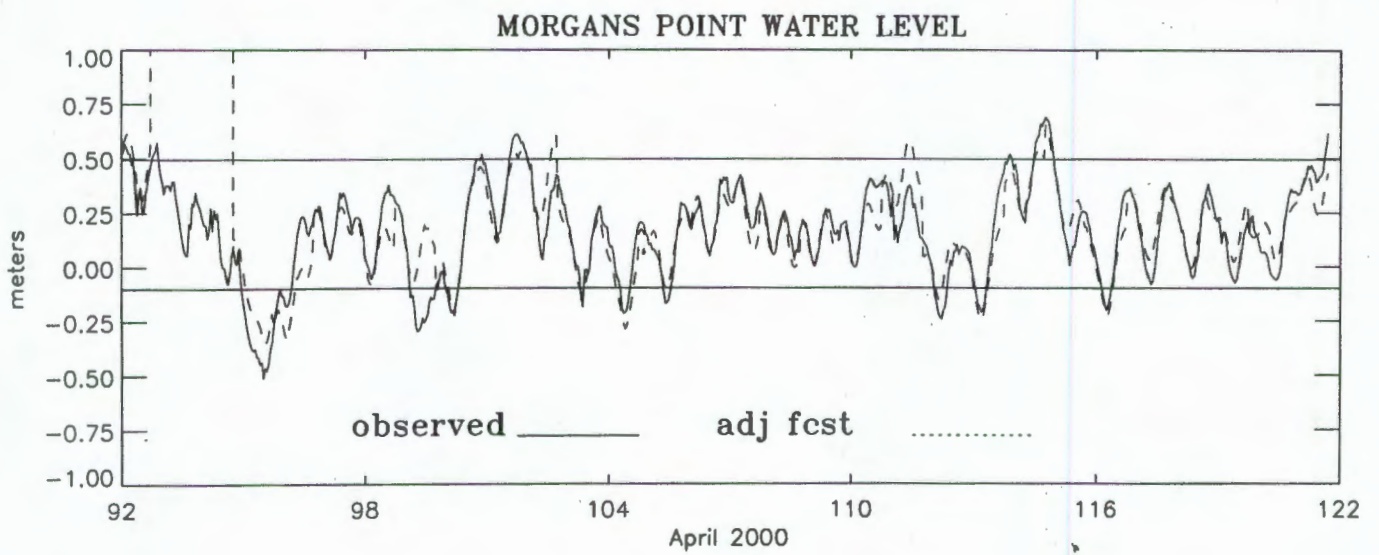
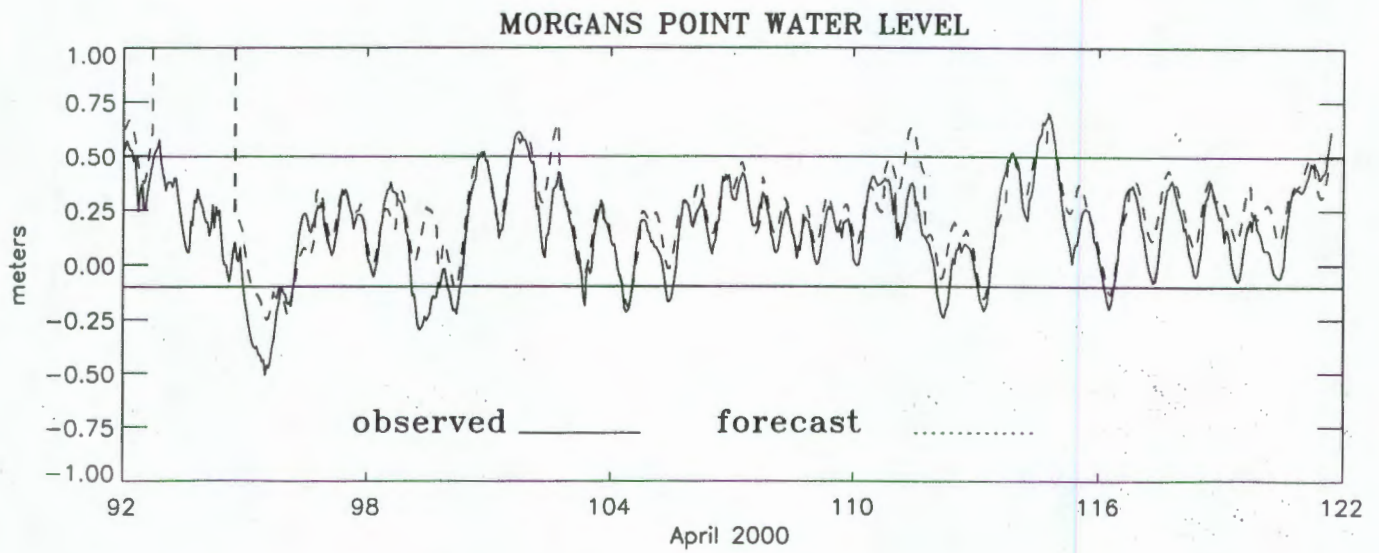


Figure 3.4. Event Analysis Plots at Morgans Point for April 2000:
Forecast and Adjusted Forecast

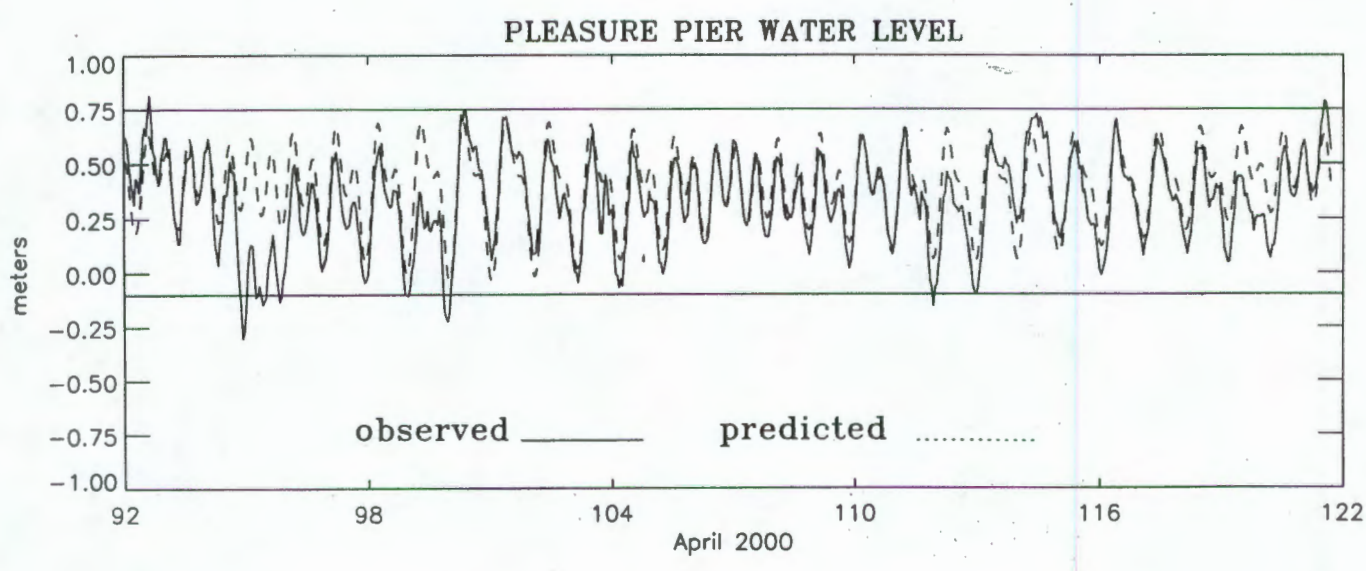
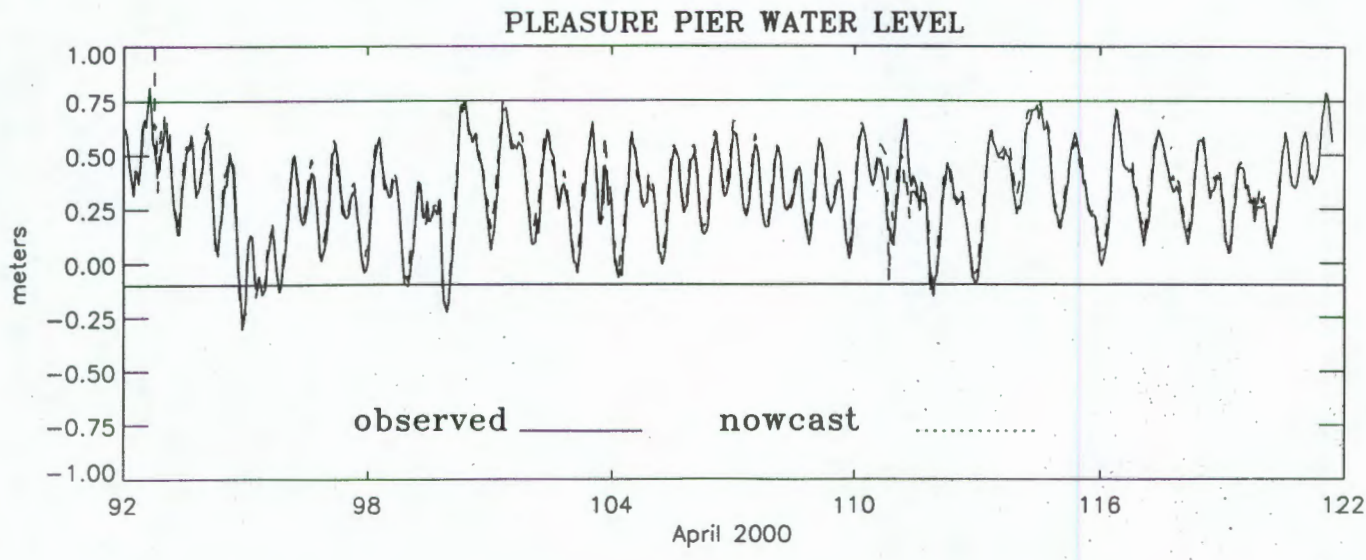


Figure 3.5. Event Analysis Plots at Galveston Pleasure Pier for April 2000:
Nowcast and Astronomical Tide Prediction

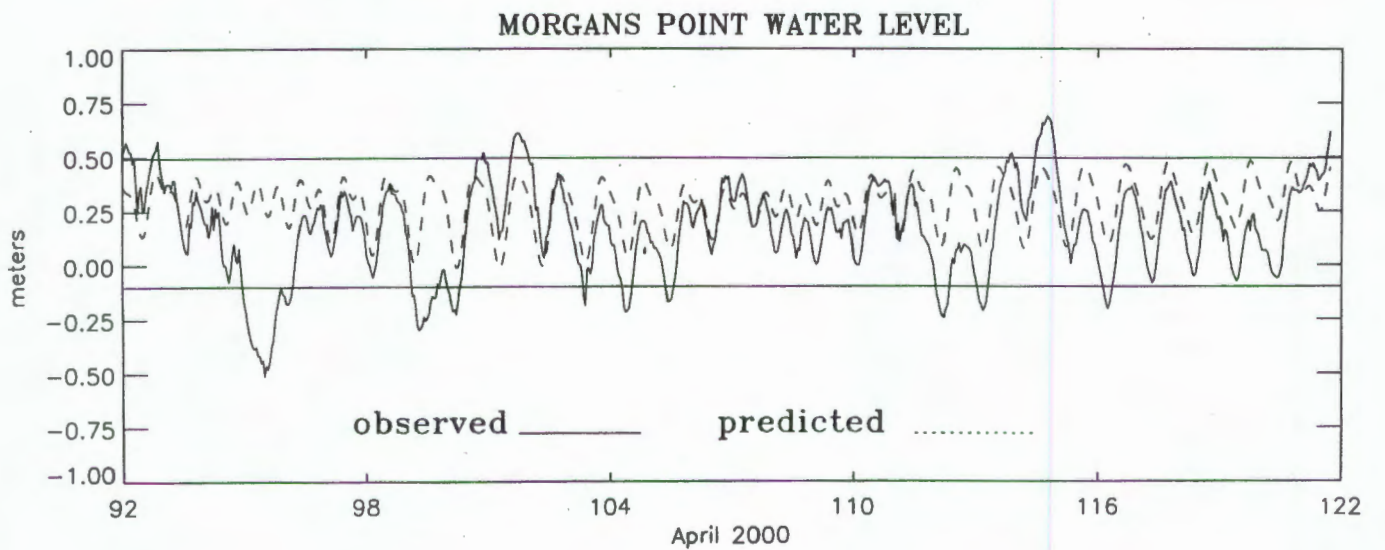
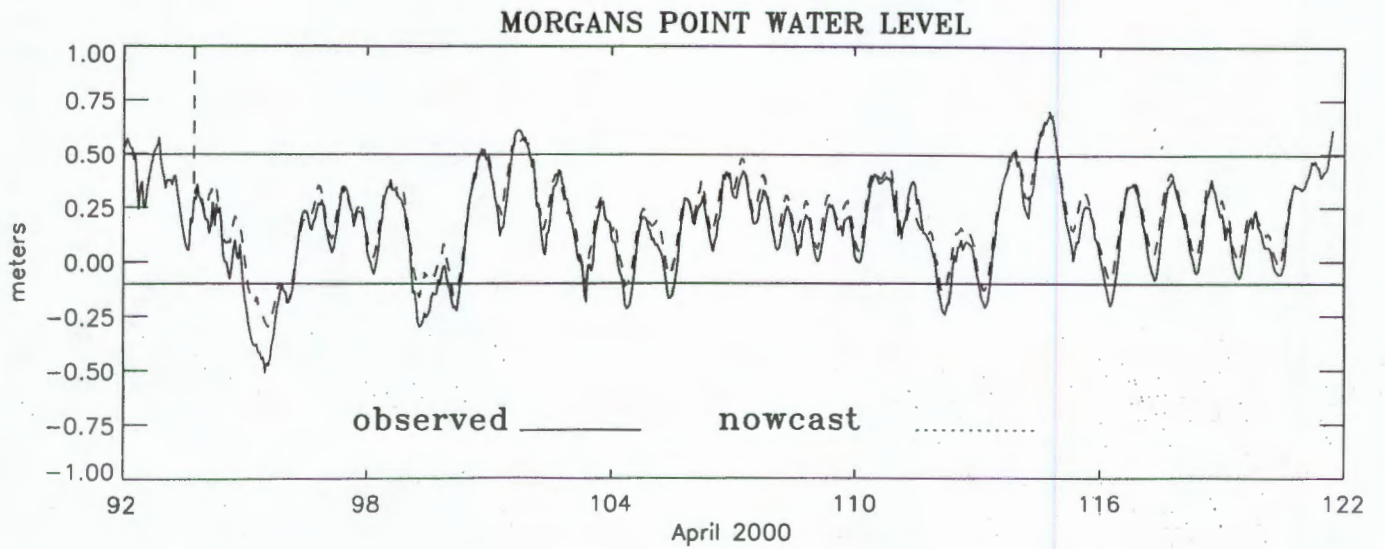


Figure 3.6. Event Analysis Plots at Morgans Point for April 2000:
Nowcast and Astronomical Tide Prediction

4. OPERATIONAL USE AND ENHANCEMENTS

In the nowcast/forecast system operational environment, it will be necessary to evaluate and assess the quality of the water level forecasts for major low water events with the associated potential for groundings and oil spills as well as high water events with associated flooding of low-lying marsh areas. While the majority of the formal acceptance statistics (NOS, 1999) will be met prior to operations, the ability of the nowcast/forecast system to forecast extreme water level events will need to be assessed on an ongoing event by event basis.

It is envisioned that the operational institution will perform the event evaluation on a monthly basis and that this evaluation will be included in the monthly nowcast/forecast system evaluation bulletins. At the end of each year, a yearly summary will be performed based on these monthly bulletins. As a result, programs similar to these documented herein will be required. It is hoped that these programs will serve as a guide for the final implementation programs.

The specification of critical values to define high and low water events may alter the number of failures and false alarms with water levels which are not significantly different from the observed water levels. As a result, Program Match.event.f provides event tables for failures and false alarms as well as for successes. In these event tables, the difference between the model peak water level and the observed peak water level for each event is given. In addition, the time of the observed peak water level and the model peak water level are now included as well as the time difference between the model peak and the observed peak. Also given is the difference in event duration and start time. At the present time, only the mean difference in water level is computed. Additional statistics such as rms error might also be determined. In the forecast adjustments, only a non time-lagged linear regression is considered. It might be useful in the future to investigate time lagged relationships.

ACKNOWLEDGMENTS

Dr. Eddie H. Shih, NOS/CO-OPS, and formally of NOS/CSDL is especially acknowledged for providing essential design inputs in the development of the program set, which was initially used to assess the East Coast Ocean Forecast System and the NWS/TDL Extratropical Storm Surge Model forecasts at NOS/NWLON water level stations along the East Coast.

REFERENCES

- Bethem, T. D., and H. R. Frey, 1991: Operational physical oceanographic real-time data dissemination, *Proceedings, IEEE Oceans 91*, 865 - 867.
- Frey, H. R., 1991: Physical oceanographic real-time systems for operational use. *Proceedings, IEEE Oceans 91*, 855 - 858.
- NOS, 1999: NOS procedures for developing and implementing operational nowcast and forecast systems for PORTS, *NOAA Technical Report NOS/CO-OPS 20*, Silver Spring, MD..
- Richardson, P. H. and R.A. Schmalz, 1999: Evaluation of forecast inputs: supplemental program documentation. NOAA, National Ocean Service, Coast Survey Development Laboratory, *NOS/CSDL Galveston Bay Nowcast/Forecast System Report Series, User Information Report 1, unpublished working report*, Silver Spring, MD.
- Schmalz, R. A., 1996: National Ocean Service Partnership: DGPS-supported hydrosurvey, water level measurement, and modeling of Galveston Bay: development and application of the numerical circulation model. NOAA, National Ocean Service, Office of Ocean and Earth Sciences, *NOAA Technical Report NOS OES 012*, Silver Spring, MD.
- Schmalz, R. A. and P.H. Richardson, 1996: Preliminary nowcast/forecast requirements for Galveston Bay. NOAA, National Ocean Service, Coast Survey Development Laboratory, *NOS/CSDL Galveston Bay Nowcast/Forecast System Report Series, System Requirements Report 1, unpublished working report*, Silver Spring, MD.
- Schmalz, R. A. and P.H. Richardson, 1998a: Development of a prototype nowcasting/forecasting System for Galveston Bay: Hindcast Studies. NOAA, National Ocean Service, Coast Survey Development Laboratory, *NOS/CSDL Galveston Bay Nowcast/Forecast System Report Series, Technical Development Report 1, unpublished working report*, Silver Spring, MD.
- Schmalz, R. A. and P.H. Richardson, 1998b: Development of a prototype nowcasting/forecasting System for Galveston Bay: Nowcast/Forecast Studies. NOAA, National Ocean Service, Coast Survey Development Laboratory, *NOS/CSDL Galveston Bay Nowcast/Forecast System Report Series, Technical Development Report 2, unpublished working report*, Silver Spring, MD.
- Williams, R. G., H. R. Frey, T. Bethem, 1990. Houston Ship Channel/Galveston Bay current prediction quality assurance miniproject, *NOAA Technical Memorandum NOS OMA 53*, Rockville, MD.

APPENDIX A. JCL AND CONTROL FILES

JCL and control files for each of the five programs are provided below as shown in Table 3.1.

reform.jcl

```
# f77 reform_coops.f calcjd.f -o reformx
```

```
# rm *.o
```

```
reformx < reform.n
```

reform.n

```
/usr/people/philr/galves/NF_eval/wlevel/observed/raw/pleas.2000.raw  
pleas.2000.obsch
```

```
60.0 time interval of raw data set
```

```
no* - designate start and end time, yes or no
```

```
1.0 start time
```

```
366.96 end time
```


readnf.jcl

f77 read_nowforc.f modelfit.f -o nowforc

rm *.o

nowforc < readnf.sep00.n
nowforc < readnf.oct00.n
nowforc < readnf.nov00.n
nowforc < readnf.dec00.n
nowforc < readnf.feb01.n
nowforc < readnf.may00.n

rm *.now.*
rm *.*N
rm *F.*

readnf.apr00.n

92.0 startday
21
pleas.now.apr
22
pleas.apr00F.050
-0.05 1.05 bias, gain
23
pie21.now.apr
24
pie21.apr00F.050
-0.05 1.05
25
bolvr.now.apr
26
bolvr.apr00F.050
-0.05 1.05
27
morgn.now.apr
28
morgn.apr00F.075
-0.075 1.05
29
eagle.now.apr
30
eagle.apr00F.050
-0.05 1.05
0 nhr_skip
forecast hours 1 - 24
30 nfiles
1 nmiss_gbm
2
2 nmiss_hsc
2
3
/opseadisk2/HGOPS.dt/gbm2000/200004/20000401/wl.092.00z
/opseadisk2/HGOPS.dt/hsc2000/200004/20000401/wl.092.00z

match.jcl

```
# f77 match.event.f calcjd.f timehi.f jdgreg.f wr_header.f -o match.ph
```

```
# rm *.o
```

```
# match.ph < match.sep00N.n > out
```

```
# match.ph < match_pred.dec00.n > out
```

```
match.ph < match.apr00F.n > out
```

```
# match.ph < match.pleasF.n > out
```

```
rm out
```

```
rm table.out
```

```
# rm table2.out
```

```
# rm table_high
```

```
# rm table_low
```

match.apr00F.n

```
0      idebug
```

```
April 2000
```

```
Observed wl data vs. forecast wl data, event analysis
```

```
92.000 startjd
```

```
122.000 endjd
```

```
0.1 crlevel - critical value
```

```
forecast - option, forecast or astronomic
```

```
table_highF.apr00.075
```

```
2 number of stations
```

```
Pleasure Pier
```

```
0.649
```

```
/usr/people/philr/galves/NF_eval/wlevel/observed/pleas.2000.obs
```

```
/usr/people/philr/galves/NF_eval/wlevel/forecast/pleas.apr00F.050
```

```
Morgans Pt
```

```
0.396
```

```
/usr/people/philr/galves/NF_eval/wlevel/observed/morgn.2000.obs
```

```
/usr/people/philr/galves/NF_eval/wlevel/forecast/morgn.apr00F.075
```

```
Pier 21
```

```
0.430
```

```
/usr/people/philr/galves/NF_eval/wlevel/observed/pie21.2000.obs
```

```
/usr/people/philr/galves/NF_eval/wlevel/forecast/pie21.apr00F.050
```

```
Port Bolivar
```

```
0.430
```

```
/usr/people/philr/galves/NF_eval/wlevel/observed/bolvr.2000.obs
```

```
/usr/people/philr/galves/NF_eval/wlevel/forecast/bolvr.apr00F.050
```

```
Eagle Point
```

```
0.336
```

```
/usr/people/philr/galves/NF_eval/wlevel/observed/eagle.2000.obs
```

```
/usr/people/philr/galves/NF_eval/wlevel/forecast/eagle.apr00F.050
```

```
IDL< .r wl.sigma.pro
```

```
cntrl.pleas
```

```
ps
landscape
0      idebug
!17Pleasure Pier!X
forecast
  92.0  tmin
  92.00 start time
 122.0  tmax
 122.0  end time
2000    year
1       number of curves
!17WATER LEVEL FORECAST, hours 1 - 24!X
!17forecast!X
/usr/people/philr/galves/NF_eval/wlevel/forecast/base/pleas.apr00F.b
-1.00 1.00 8  yrange, and number of tick marks
April 2000
  0.100  critical value
  0.649  range_offset
```

```
IDL< .r wl.multcur.pro
```

```
c.pleas_apr00
```

```
ps
landscape
0      idebug
!17Pleasure Pier!X
  92.00 tmin
  92.00 start time
 122.00 tmax
 121.71 end time
5      number of curves
!17PLEASURE PIER WATER LEVEL!X
!17observed!X
observed
/usr/people/philr/galves/NF_eval/wlevel/observed/pleas.2000.obs
!17nowcast!X
nowcast
/usr/people/philr/galves/NF_eval/wlevel/nowcast/pleas.apr00N
!17predicted!X
predicted
/usr/people/philr/galves/NF_eval/wlevel/astron/pleas.2000.pred_ref
!17forecast!X
forecast
/usr/people/philr/galves/NF_eval/wlevel/forecast/base/pleas.apr00F.b
!17adj fcst!X
forecast
/usr/people/philr/galves/NF_eval/wlevel/forecast/pleas.apr00F.050
-1.00 1.00 8  yrange, and number of tick marks
April 2000
  0.100  crlevel
  0.649  range_offset
```