CSDL Information Technical Note No. 3

ETSS VS. DGOM MODEL WATER LEVEL COMPARISONS: PROGRAM DOCUMENTATION AND MONTHLY ANALYSIS

Silver Spring, Maryland
February 2004

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

CSDL Informal Technical Note No. 3

ETSS VS. DGOM MODEL WATER LEVEL COMPARISONS: PROGRAM DOCUMENTATION AND MONTHLY ANALYSIS

Philip H. Richardson
Richard A. Schmalz, Jr.

February 2004

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

NOTICE

TABLE OF CONTENTS

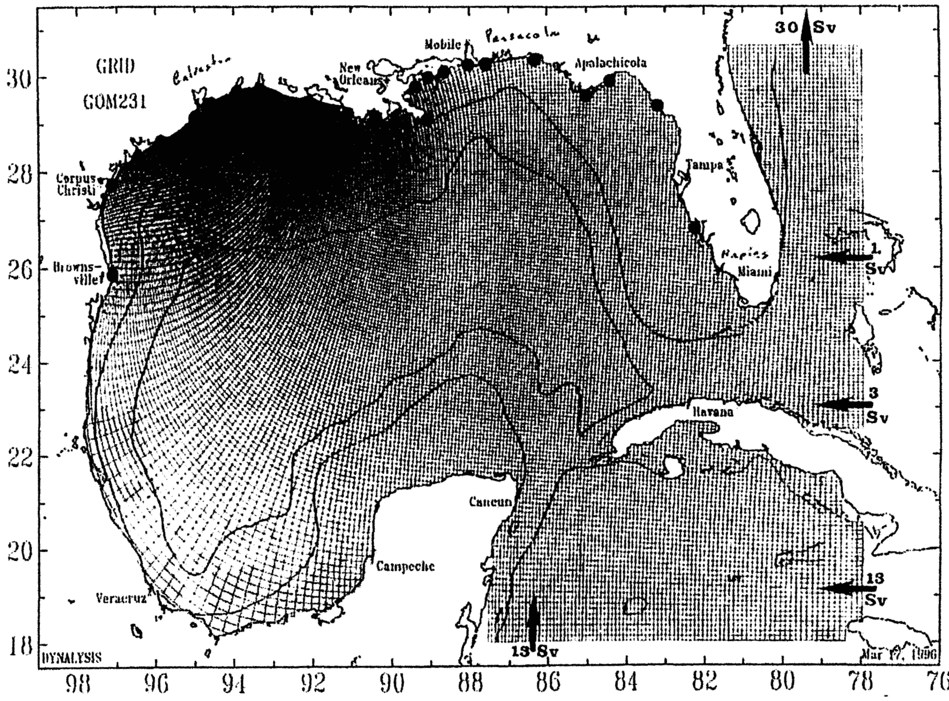
LIST OF FIGURES

LIST OF TABLES

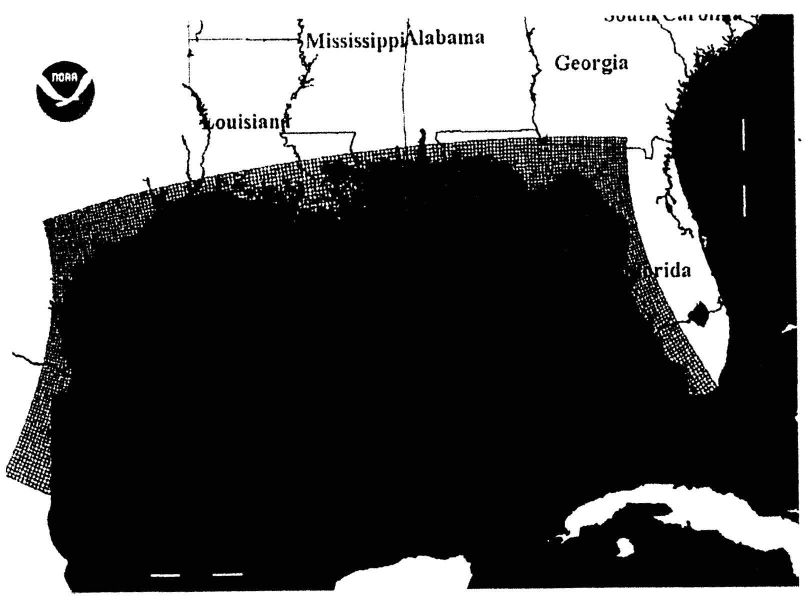
LIST OF COMPUTER PROGRAM LISTINGS

ABSTRACT

Nontidal water level comparisons of model versus observation at several locations (Galveston Pleasure Pier, TX; Pensacola, FL; Naples, FL) around the Gulf of Mexico shoreline were performed. Dynalysis Gulf of Mexico (DGOM) and NWS/MDL Extratropical Storm Surge (ETSS) model water levels were compared with the observations and the software is documented herein. Observed nontidal water levels were determined via 30 hour low pass filtering of the observed total water level. For November 2002, the observed nontidal water levels were also obtained by subtraction of the predicted astronomical tide from the observed total water level (detiding). Program descriptions and listings are provided along with a description of the processing steps. Script and program input files are given in Appendix A. Monthly comparisons are presented for November 2002, January 2003, May 2003, and July 2003. Event comparisons are performed at Galveston Pleasure Pier, TX, for a high water event in November 2002, a major low water event in January 2003 (associated with a cold frontal passage) and for Hurricane Claudette in July 2003. The DGOM and ETSS model water levels compared favorably to the observations and were of near equal quality during the energetic fall and winter months. The DGOM model water levels exhibit slightly more spread about the observations than the ETSS model water levels during the spring and summer months, which in the absence of tropical storms, tend to be quiescent. During Hurricane Claudette, nontidal water levels are under predicted by about 1.5 feet by the ETSS model and over predicted by nearly four feet by the DGOM model. The cause of the discrepancies in forecast water levels exhibited by the two models is under further investigation and a brief outline of future work is presented to conclude the report.



DGOM Model Grid



ETSS Model Grid

DGOM/ETSS Model Grids

1. INTRODUCTION

The Dynalysis Gulf of Mexico (DGOM) model employs a three dimensional split mode finite difference method and makes use of the USN COAMPS wind and sea level atmospheric pressure forcings. The horizontal resolution of these meteorological forcings is order 20 km. Additional model details may be found in Patchen et al. (1999). The DGOM model has also been used to simulate the water level response to tropical cyclones (Patchen and Herring, 1998). The DGOM model has been set-up in a forecast mode by Patchen and Blaha (2002) at NAVOCEANO and has been run in parallel at Dynalysis. Hourly forecast results over the 48 hour forecast period have been made available from the parallel Dynalysis runs to NOS for further analysis. While model data comparisons of currents have been performed along the Louisiana and Texas coastal shelves (Patchen et al., 1998; Patchen et al., 1999; Patchen et al., 2002), no basin wide assessment of water levels has been previously conducted. This report seeks to address this issue.

NOS/CSDL has utilized the nontidal water level forecasts produced by the NWS Marine Techniques Laboratory (formerly, the Techniques Development Laboratory (TDL)) Extratropical Storm Surge (ETSS) model for offshore water level boundary conditions for the New York Harbor /Port of New Jersey, Chesapeake Bay, and Galveston Bay forecast systems. Separate domains are run for the East Coast, West Coast, Alaskan Coast, and Gulf of Mexico. The NWS Aviation Model is used to provide the meteorological forcings at order 100 km resolution. The two-dimensional depth averaged shallow water equations are solved in complex variables via finite differences on an elliptical grid. See Chen et al. (1993) for additional model details. Periodic informal evaluations have been performed by Chen (2003) for storm events.

Here a set of new programs (read_tdl.f, read_dyn.f, adjust.f, wl_sa.ph.f, and plot_wlanal.pro) has been developed to compare the performance of the nontidal water level response throughout the Gulf of Mexico of the DGOM and TDL/ETTS models. We have compared daily forecast hours 6-36 of both the DGOM and TDL/ETSS models to observed nontidal water levels at several stations along the Gulf of Mexico coast based on methods previously developed by Richardson and Schmalz (2002).

Reform_coops was written to reformat either hourly or six minute observed water level data obtained from the Center for Operational Oceanographic Products and Services (CO-OPS) into a standard analysis format. CO-OPS water level data are acquired from their website at http://co-ops.nos.noaa.gov. Reform_coops.f is a standard program which requires that the water level data be given with respect to the MLLW datum. See Richardson and Schmalz (2002) for further details and a program listing.

Read_tdl.f was developed to read forecast water level results from the TDL/ETSS model. The program reads water level results from 12z files to produce forecast files for Naples, FL, Pensacola, FL, and Galveston Pleasure Pier, TX.

Read_dyn.f was developed to read DGOM model data files, search for the desired analysis stations, and write the selected water level data in a standard format for further analysis.

Adjust.f was written to adjust the daily forecast by adding or subtracting, to each forecast point, the offset obtained from the difference of the initial observed point and the initial forecast point.

The statistical analysis is performed by wl_sa.ph.f. RMS and standard deviation statistics of the error signal are calculated on a daily forecast basis, and combined for the entire month. The mean and standard deviation for the observed and model water levels are also calculated.

Plot_wlanal.pro is written in the IDL programming language. The program will plot the observed water level along with points representing the high, low, start and end points for each daily forecast. Symbols used to represent these points are plus, square, triangle, and asterisk. Plot.wlanal.pro generates one plot per page.

Reform_coops.f, read_tdl.f, read_dyn.f, and wl_sa.ph.f are written in FORTRAN 77, while plot_wlanal.pro is written in IDL. All programs are run on an SGI workstation (Unix).

In Chapter 2, a description of each program is provided as well as program listings. In Chapter 3, instruction is provided to run the analysis program set. Analysis results for November 2002, January 2003, May 2003, and July 2003 are presented separately in Chapters 4-7, respectively. In Chapter 8, some conclusions are drawn from the work already completed, as well as recommendations for future subjects of study. Complete script and control file listings are given in Appendix A.

2. PROGRAM DESCRIPTIONS

2.1. Program Read_tdl.f

The listing for Read_tdl.f is given in Program Listing 2.1. The program first reads nstn_r, the number of stations to read. There are forecast data for 22 stations in a TDL (etss) file. Next read is nstn_wr, the number of stations to write forecast data for. For this comparison, nstn_wr is three. The program then reads tdl_file, the TDL forecast file located on OCEAN2 (an SGI workstation), in the adaser directory. For each of stations 1 through nstn_wr, a station number is read, a logical unit number, and the forecast output filename.

Read_tdl.f is a very straight forward program. The TDL forecast data file (12z) is opened, and the output files are opened for stations 1 through nstn_wr. Read_tdl will read water level values for hours 1 through 24, storing the values in the array iwl. The next line of data is read for hours 25 through 48. For each of stations 1 through nstn_wr, the time (Julian date) and water level value are written for hours 6 through 36. The output forecast file for a given station will begin at .75 of that Julian day, and will proceed from 0.0 to 1.00 of the following day. Basically, one skips 6 hours into the 12z forecast file and then uses the next 30 hours. This condition simulates the use of the forecast within a nowcast/forecast system mode; e.g., there is a 6 hour meteorological forecast processing time.

```
      program read_tdl

c Program to read slosh model predicted water elevations.
c The TDL forecast files are located on OCEAN2 - .
c     /ocean2dir1/adaser/ocean/fcsts/tdlet/archives/
c
c Author : Phil Richardson
c
c Language : FORTRAN
c
c Version Date : January 13, 2003

*******************************************************

      parameter (nstation=24,nhrs2=48)

      character*18 stn_name(nstation)
      character*22 tdlfile_out(nstation)
      character*69 tdl_file
      character*80 line

      dimension iwl(nhrs2),rwl(nhrs2),lunout(nstation),
     *          istn(nstation)

      conv_ft = 0.10

*******************************************************

c Read from the control file -
c
c        nstn_r - number of stations to read
c       nstn_wr - number of output files to write
c      tdl_file - slosh water elevation data file
c        istn() - TDL station number
c      lunout() - logical unit number for station output files
c tdlfile_out() - file names for station output files

      read(5,*)nstn_r
      read(5,*)nstn_wr
      read(5,31)tdl_file
      write(6,31)tdl_file
      do ns=1,nstn_wr
         read(5,*)istn(ns)
         read(5,*)lunout(ns)
         read(5,32)tdlfile_out(ns)
      enddo


   31 format(a69)
   32 format(a22)

*******************************************************

c Open daily TDL file.

      lun = 7
```

Program Listing 2.1 Read_tdl.f

```
      open(lun,file=tdl_file,form='formatted',
     *     status='old')

c Open station output files.

      do ns=1,nstn_wr
         open(lunout(ns),file=tdlfile_out(ns),form='formatted')
      enddo

********************************************************

c Read from TDL file

      read(tdl_file,'(6x,i4,i2,i2,a2)')iyear,imonr,idayr
      write(6,*)imonr,idayr,iyear

      ncnt = 1

      do l=1,3
         read(lun,'(a80)')line
      enddo

      do 100 ns=1,nstn_r
         read(lun,101)stn_name(ns),iwl_hr0
         write(6,101)stn_name(ns),iwl_hr0
         rwl_hr0 = iwl_hr0

         read(lun,102)(iwl(nhr),nhr=1,24)
         write(6,102)(iwl(nhr),nhr=1,24)

         read(lun,102)(iwl(nhr),nhr=25,48)
         write(6,102)(iwl(nhr),nhr=25,48)

         call calcjd(ijd,imonr,idayr,iyear)
         write(6,*)ijd

         rjd = ijd

         rwl_hr0 = rwl_hr0 * conv_ft

         if(ns.eq.istn(ncnt))then
            do 110 nhr=6,36
               rwl(nhr) = float(iwl(nhr)) * conv_ft
               rjdtime = rjd + float(nhr)/24.0 + 0.50
               write(lunout(ncnt),111)rjdtime,rwl(nhr)
  110       continue
            ncnt = ncnt + 1
         endif
  100 continue


  101 format(1x,a18,50x,i3)
  102 format(24i3)
  111 format(2f10.4)

********************************************************
```

Program Listing 2.1 Read_tdl.f (continued)

```
      end





      SUBROUTINE CALCJD (JDY,ICM,ICD,IYR)


C     This subroutine will convert the calender month (ICM) and
C     calender day (ICD) to the corresponding Julian day (JDY).
C
C     Input arguments -
C
C           ICM - Calender Month
C           ICD - Calender Day
C           IYR - Year
C
C     Output Argument -
C
C           JDY - Julian Day

      DIMENSION JDAY(12), JDAYL(12)
      DATA JDAY /0,31,59,90,120,151,181,212,243,273,304,334/
      DATA JDAYL /0,31,60,91,121,152,182,213,244,274,305,335/

      IF(MOD(IYR,4) .EQ. 0 .AND. MOD(IYR,100) .NE. 0 .OR.
     *   MOD(IYR,400) .EQ. 0) THEN
         RJD = FLOAT(JDAYL(ICM)) + FLOAT(ICD)
      ELSE
         RJD = FLOAT(JDAY(ICM)) + FLOAT(ICD)
      END IF

      JDY = INT(RJD)
      RETURN
      END
```

Program Listing 2.1 Read_tdl.f (continued)

2.2 Program Read_dyn.f

Read_dyn.f was developed to read the Dynalysis model data files, search for the desired station, and write the water level data out into a standard format (2f10.4). The listing for read_dyn.f is provided in Program Listing 2.2.

Variables read from the control file include idebug, filenm, ndyn, dynstat, and fileout. Idebug controls the debug option. Filenm is the filename of the Dynalysis model data file. Ndyn is the number of stations for which output files will be generated. Dynstat is the station name and fileout is the output filename (by station).

The program searches for a character string identical to the desired station. The program will find the specific station location, out of a set of station locations, which is closest to the TDL station location. The calendar date information is read, along with the water level value. The calendar date is converted to Julian time, and the time and depth are written to the appropriate output file.

The 12z Dynalysis forecast is processed as described previously for the ETSS forecast. The 12z cycle was selected for analysis rather than the 00z forecast cycle due to the greater reliability of the 12z cycle forecasts from Dynalysis.

```
c Program : read_dyn.f
c
c Author : Phil Richardson
c
c Purpose : To read Dynalysis model data (wl) from GOM
c           stations, then reformat into standard format
c           (2f10.4).  Dynalysis data is in feet, 12z
c           files begin on hour 13.
c
c Language : FORTRAN 77
c
c Version Date : December 2, 2002

*******************************************************

      parameter(nstats=5)

      character*8  stanam,dynstat(nstats)
      character*16 fileout(nstats)
      character*80 filenm,line

      dimension rdepth(48),lunout(nstats)

*******************************************************

c Read from input :
c
c     idebug = 1, echo input data in feet
c              2, write information to top of output file
c     filenm - name of Dynalysis model data file
c       ndyn - number of Dynalysis stations to read
c  dynstat() - Dynalysis station name
c  fileout() - output station filename


      read(5,*)idebug
      read(5,21)filenm
      read(5,*)ndyn
      do n=1,ndyn
         read(5,22)dynstat(n)
         read(5,23)fileout(n)
      enddo


   21 format(a80)
   22 format(1x,a8)
   23 format(1x,a16)

*******************************************************

c Open Dynalysis model file.

      lun = 7
      open(lun,file=filenm,form='formatted',status='old')

      do lf=1,ndyn
```

Program Listing 2.2 Read_dyn.f

```
         lunout(lf) = lf + 7
         write(6,*)lunout(lf)
         open(lunout(lf),file=fileout(lf),form='formatted')
      enddo

*******************************************************

c Find correct start point by searching file for 'FILE='.

   40 continue
      read(lun,21)line
      if(line(1:6).eq.'FILE= ')then
         write(6,*)line
      else
         goto 40
      endif
      backspace lun


      nc = 1

   50 continue

      read(lun,21)line
      if(line(9:16).eq.dynstat(nc))then
         write(6,31)line(9:16)

         backspace lun

         read(lun,32,err=100)stanam,sdepth,depth,iyr,jday,
     *               ihr,imin,samp_int
         iyear = iyr + 2000
         write(6,33)stanam,sdepth,depth,jday,iyear,ihr,imin,
     *              samp_int

         do n=1,24
            read(lun,36,err=105)idepth1,iwu,iwv,iat_press,idepth2
            rdepth1 = float(idepth1) * 0.001
            rdepth2 = float(idepth2) * 0.001
            if(idebug.eq.1)write(6,*)rdepth1,rdepth2

            rdepth(2*n-1) = rdepth1
            rdepth(2*n) = rdepth2
         enddo

         if(idebug.eq.2)write(lunout(nc),41)stanam
         if(idebug.eq.2)then
            write(lunout(nc),*)jday,iyear,ihr,imin
         endif

         rjday = jday
         if(idebug.eq.2)write(lunout(nc),42)rjday

         do nhr=6,36
            write(6,*)nhr,rdepth(nhr)
            rjtime = rjday + float(nhr+12)/24.0
```
Program Listing 2.2 Read_dyn.f

```
            write(lunout(nc),43)rjtime,rdepth(nhr)
         enddo

         if(nc.eq.ndyn)then
            stop
         endif
         nc = nc +1
      endif
      goto 50


      stop

  100 continue
      write(6,*)'ERROR - Reading header line'
      write(6,*)stanam,jday,iyear,ihr
      write(6,'(a80)')line
      stop

  105 continue
      write(6,*)stanam
      write(6,*)'ERROR - Reading data'
      stop

*******************************************************

   31 format(/,1x,'Station : ',a8)
   32 format(8x,a8,i4,i5,i2,1x,i3,2(1x,i2),10x,i2)
   33 format(1x,a8,/,' station depth =',i4,
     *          'm,',' meter depth =',
     *       i4,'m',2x,/,1x,'Julian day 'i3,2x,i4,2x,
     *       i2,':',i2,/,' Sampling interval = ',1x,i2,
     *          ' minutes',/)
   34 format(/)
   36 format(3(1x,i6),2i7)
   41 format('Station : ',a8)
   42 format('Julian start day is',f9.4,/)
   43 format(2(1x,f9.4))

*******************************************************

      end
```

Program Listing 2.2 Read_dyn.f

2.3 Program Adjust.f

The listing for Adjust.f is given in Program Listing 2.3. The purpose for Adjust.f is to adjust each data point of the daily forecast by adding the offset obtained from the difference between the initial observed point and the initial forecast point. While other adjustment methods are possible (based on longer term observations and associated ramping), these more elaborate techniques have not been used here. The program is generally run for all stations at once, then run for each day of the comparison period. The adjusted forecast files are used for all analysis work.

```
c Program Name : adjust.f
c
c Purpose : To take some wl data and add or subtract
c           some constant value.  This version of the
c           program is specifically revised for the
c           TDL, Dynalysis comparison.  The first
c           model value is compared with the first
c           obs value, then ajusted accordingly.
c
c Original Version Date : June 12, 1996
c
c Version Date : Jan 13, 2003

*******************************************************

      parameter(nstation=10)

      character*25 fileout(nstation)
      character*67 filein(nstation)
      character*69 fileobs(nstation)

      dimension lun(nstation), lunout(nstation),
     *          lunobs(nstation)

*******************************************************

c     Read from input :
c
c          lun() - logical unit number (model file)
c       filein() - input model file
c       lunobs() - logical unit number (obs file)
c      fileout() - adjusted model file


      read(5,*)nstn
      read(5,*)start_time

      do ns=1,nstn
         read(5,*)lun(ns)
         read(5,41)filein(ns)
         write(6,41)filein(ns)
         read(5,*)lunobs(ns)
         read(5,44)fileobs(ns)
         write(6,44)fileobs(ns)
         read(5,*)lunout(ns)
         read(5,42)fileout(ns)
         write(6,42)fileout(ns)
      enddo


   41 format(a67)
   42 format(a25)
   44 format(a69)

*******************************************************
```

Program Listing 2.3 Adjust.f

```
c      Open model water level data file, observed
c      water level file, then output (adjusted) file.

       do 80 ns=1,nstn
          open(lun(ns),file=filein(ns),form='formatted',
     *         status='old')

          open(lunobs(ns),file=fileobs(ns),form='formatted',
     *         status='old')

          open(lunout(ns),file=fileout(ns),form='formatted')
   80  continue

*******************************************************

       do 120 ns=1,nstn

  100     continue
          read(lun(ns),*)time,wl
          if(time.lt.start_time)goto 100

          tmstrt_mod = time
          wl_mod = wl
          write(6,101)time,wl_mod
          backspace lun(ns)
c         rewind lun(ns)

  105     continue
          read(lunobs(ns),*)time,wl
          if(time.lt.start_time)goto 105

          tmstrt_obs = time
          wl_obs = wl
          write(6,102)time,wl_obs
          rewind lunobs(ns)

          wl_diff = wl_obs - wl_mod
          write(6,*)wl_diff


  110     continue
          read(lun(ns),*,end=120)time,wl
          wl = wl + wl_diff
          write(lunout(ns),51)time,wl
          goto 110

  120 continue


  101 format(/,'first model data point : ',2f9.4)
  102 format(/,'first observed data point : ',2f9.4)

*******************************************************

   43 format('amount to be added to water level',f9.4)
   51 format(2f10.4)
```

Program Listing 2.3 Adjust.f

```
      stop
      end
```

Program Listing 2.3 Adjust.f

2.4 Program Wl_sa.ph.f

Wl_sa.ph.f is a revised version of the program used to assess the performance of the COFS model, originally written by Kate Bosley. The program was revised slightly in order to simultaneously assess the ETSS model and the DGOM model. The listing for Wl_sa.ph.f is given in Program Listing 2.4. Following the parameter and dimension statements, and after the character variables are declared, wl_sa.ph.f will read necessary information from the control file. Variables read from the control file include idebug, istat, statnam, fout, rjd_start, rjd_stop, and tdmax. Idebug controls the debug function. Istat is the number of stations. Statnam is the station name. Fout is the output file name. Rjd_start is the start time, and rjd_stop is the stop time. Tdmax is the maximum allowable time difference between two data points.

The 600 loop is the day loop, beginning with nd = 1, and finishing with nd = ndays. Both model files, TDL and Dynalysis, are opened, along with the observed data file. Wl_sa.ph.f calculates the variance and mean for each of the models, and for the observed data. Subroutine compare is called to calculate the rms difference between both models and the observed.

The daily statistics are written to output in the 850 loop, which begins with nd = 1 and ends with nd = ndays. The statistics for the entire month are calculated in the 1000 loop, and the results are written to the monthly summary table.

```
c  program Name : wl_sa.ph2
c
c  (water level skill assessment revised version)
c
c  Philip Richardson
c
c  (301) 713-2809x115
c
c  National Ocean Service
c
c  27 April 1993, revised March 24, 1995.
c  Latest revision : Aril 9, 2003
c
c  Purpose:
c     To compare water level time series data taken from
c     the TDL forecast with observed data, and water level
c     data from the Dynalysis model with observed data.  This
c     version of the program was specifically written for
c     this purpose (TDL vs Dynalysis comparison).
c
c
c  Revision June 20, 1996
c           Double loop in subroutine compare is eliminated
c
c  Execution : Run on opsea, wl_sa < control file

*****************************************************

      parameter (nsta=20, ndat=20000, nday=30)

      character*1 formfd
      character*4 stanam(nsta)
      character*10 filestat
      character*13 statnam(nsta),monyr
      character*50 fout,title,title2
      character*75 obsdat(nsta)
      character*78 moddat_tdl(nday,nsta),moddat_dyn(nday,nsta)

      dimension avgwlo(nday,nsta),sdwlo(nday,nsta),
     *          avgwlm1(nday,nsta),sdwlm1(nday,nsta),
     *          avgwlm2(nday,nsta),sdwlm2(nday,nsta),
     *          rmswl1(nday,nsta),rmswl2(nday,nsta),
     *          varim1(nday,nsta),varim2(nday,nsta),
     *          numc(nday,nsta)
      dimension rms1sq_day(nday,nsta),rms2sq_day(nday,nsta)
      dimension rjd_start(nday),rjd_stop(nday),jpts(nday)
      dimension lunobs(nsta),rms1day_tot(nsta),rms2day_tot(nsta),
     *          numc_tot(nsta),rms1_month(nsta),rms2_month(nsta)

      real timem1(ndat),wlm1(ndat),timeo(ndat),wlo(ndat),
     *     timem2(ndat),wlm2(ndat),wloadj(ndat)

      common/time/tdmax

      formfd = char(12)
```

Program Listing 2.4 Wl_sa.ph.f

```
******************************************************

c  Read from Input :
c
c  From the control file read the number of stations, the
c  name of the output file, the start and stop times.
c
c        idebug - debug switch
c           = 1 - statistical output to file 10
c         istat - number of stations
c          fout - name of output file
c         ndays - number of days
c   rjd_start() - start time
c    rjd_stop() - stop time
c         tdmax - maximum allowable time difference

      read(5,*)idebug
      read(5,'(i4)')istat
      do ns=1,istat
         read(5,46)statnam(ns)
      enddo

      read(5,47)fout
      read(5,47)title
      read(5,47)title2
      read(5,'(a13)')monyr

      read(5,*)ndays
      do nd=1,ndays
         read(5,*)rjd_start(nd)
         read(5,*)rjd_stop(nd)
      enddo

      read(5,*)tdmax

      write(6,47)title
      write(6,47)title2
      write(6,48)istat

      small = .000012


   46 format(a13)
   47 format(a50)
   48 format(1x,i4,1x,'stations')

******************************************************

c  open output file and write header
c                  {unit 9}

      open(9,file=fout,form='formatted')

      filestat='stat.out'
      open(10,file=filestat,form='formatted')

```

Program Listing 2.4 Wl_sa.ph.f (continued)

```
*******************************************************

c Variables :
c
c Read from control file -
c      lunobs() - logical unit number for observed data file
c      obsdat() - observed data file
c
c   nobs - number of data points (obs) file


      do 300 ns=1,istat

*  open observed data file

         read(5,*)lunobs(ns)
         read(5,349)obsdat(ns)
         open(lunobs(ns),file=obsdat(ns),form='formatted',
     *        status='old')

         nobs = 0
  350    continue
         read(lunobs(ns),*,end=355)time,wl
         nobs = nobs + 1
         if(nobs.eq.1)write(6,311)time
         goto 350
  355    continue

         rewind lunobs(ns)

         write(6,312)nobs

  300 continue


  311 format(1x,'Observed file begins',f10.4)
  312 format(1x,i6,' data pts in observed file')
  349 format(a75)

*******************************************************


      do 600 nd=1,ndays

      write(6,10)rjd_start(nd),rjd_stop(nd)

      do 900 i=1,istat

         read(5,'(a4)')stanam(i)

c  open TDL model data file
         read(5,949)moddat_tdl(nd,i)
         write(6,949)moddat_tdl(nd,i)
         open(unit=18,file=moddat_tdl(nd,i),form='formatted',
     *        status='old')

```

Program Listing 2.4 Wl_sa.ph.f (continued)

```
c  open Dynalysis model data file
         read(5,949)moddat_dyn(nd,i)
         open(unit=19,file=moddat_dyn(nd,i),form='formatted',
     *        status='old')


  949 format(a78)


*  initialize before each station

         sumwlm1 = 0.0
         sumwlm2 = 0.0
         sqsumwlm1 = 0.0
         sqsumwlm2 = 0.0
         sumsqwlm1 = 0.0
         sumsqwlm2 = 0.0

         sumwlo = 0.0
         sqsumwlo = 0.0
         sumsqwlo = 0.0

*******************************************************

c     Count data points in TDL, and Dynalysis files.
c     Count number of points is observed data file.


      nmod_tdl = 0
  140 continue
      read(18,*,end=145)time,wl
      nmod_tdl = nmod_tdl + 1
      if(nmod_tdl.eq.1)write(6,141)time
      goto 140
  145 continue

      rewind 18

      write(6,51)nmod_tdl


      nmod_dyn = 0
  160 continue
      read(19,*,end=165)time,wl
      nmod_dyn = nmod_dyn + 1
      if(nmod_dyn.eq.1)write(6,161)time
      goto 160
  165 continue

      rewind 19

      write(6,52)nmod_dyn


  141 format(1x,'Model file begins',f10.4)
  161 format(1x,'Dynalysis file begins',f10.4)
```

Program Listing 2.4 Wl_sa.ph.f (continued)

```
    51 format(1x,i6,' data pts in TDL model file')
    52 format(1x,i6,' data pts in Dynalysis file')

*********************************************************

c  Find data within the time frame selected,
c  begin statistical calculations.
c
c  jm1 - TDL
c  jm2 - Dynalysis


         jm1 = 0
 180     continue
         read(18,*,end=185)ctimem1,cwlm1
         if(ctimem1.ge.rjd_start(nd).and.ctimem1.le.rjd_stop(nd))
     *      then
            jm1 = jm1 + 1
            timem1(jm1) = ctimem1
            wlm1(jm1) = cwlm1
            sumwlm1 = sumwlm1 + wlm1(jm1)
            sumsqwlm1 = sumsqwlm1 + wlm1(jm1)**2
         end if
         goto 180
 185     continue


         jm2 = 0
 190     continue
         read(19,*,end=195)ctimem2,cwlm2
         if(ctimem2.ge.rjd_start(nd).and.ctimem2.le.rjd_stop(nd))
     *      then
            jm2 = jm2 + 1
            timem2(jm2) = ctimem2
            wlm2(jm2) = cwlm2
            sumwlm2 = sumwlm2 + wlm2(jm2)
            sumsqwlm2 = sumsqwlm2 + wlm2(jm2)**2
         endif
         goto 190
 195     continue


         jo = 0
 200     continue
         read(lunobs(i),*,end=205)ctimeo,cwlo
            if(ctimeo.ge.rjd_start(nd).and.ctimeo.le.rjd_stop(nd))
     *         then
               jo = jo + 1
               timeo(jo) = ctimeo
               wlo(jo) = cwlo
               sumwlo = sumwlo + wlo(jo)
               sumsqwlo = sumsqwlo + wlo(jo)**2
            end if
         goto 200
 205     continue
         jpts(nd) = jo
```

Program Listing 2.4 Wl_sa.ph.f (continued)

```fortran


c      calculate mean and variance
c
c    Variables :
c     avgwlo() - mean observed wl
c    avgwlm1() - mean TDL wl
c    avgwlm2() - mean Dynalysis wl
c      sumwlm1 - sum of m1 (TDL) terms
c      sumwlm2 - sum of m2 (Dynalysis) terms
c    sumsqwlm1 - sum of m1 (TDL) terms squared
c    sumsqwlm2 - sum of m2 (Dynalysis) terms squared
c    sqsumwlm1 - sumwlm1 squared
c    sqsumwlm2 - sumwlm2 squared

        avgwlo(nd,i) = sumwlo/float(jo)
        sqsumwlo = sumwlo*sumwlo
        term1wlo = jo * sumsqwlo
        tnumwlo = term1wlo - sqsumwlo
        tdenomwlo = jo**2
        varwlo = tnumwlo/tdenomwlo
        sdwlo(nd,i) = sqrt(varwlo)

        write(10,211)nd,statnam(i)
        avgwlm1(nd,i) = sumwlm1/float(jm1)
        avgwlm2(nd,i) = sumwlm2/float(jm2)
        write(10,*)'sumwlm1 = ',sumwlm1
        sqsumwlm1 = sumwlm1 * sumwlm1
        sqsumwlm2 = sumwlm2 * sumwlm2
        term1wlm1 = jm1 * sumsqwlm1
        term1wlm2 = jm2 * sumsqwlm2
        write(10,*)term1wlm1,sqsumwlm1
        tnumwlm1 = term1wlm1 - sqsumwlm1
        write(10,*)'tnum = ',tnumwlm1
        if(abs(tnumwlm1).lt.small)tnumwlm1 = 0.0
        tnumwlm2 = term1wlm2 - sqsumwlm2
        tdenomwlm1 = jm1**2
        tdenomwlm2 = jm2**2
        varwlm1 = tnumwlm1/tdenomwlm1
        varwlm2 = tnumwlm2/tdenomwlm2
        sdwlm1(nd,i) = sqrt(varwlm1)
        sdwlm2(nd,i) = sqrt(varwlm2)
        write(10,213)sdwlm1(nd,i)
        varim1(nd,i) = sdwlm1(nd,i)/sdwlo(nd,i)
        varim2(nd,i) = sdwlm2(nd,i)/sdwlo(nd,i)

        write(6,212)jo,jm1,jm2

        nmin = min0(jm1,jo)


  211 format(/,1x,'day ',i2,',',1x,a13)
  212 format(1x,i5,' observed points in period',
     *       1x,i5,'TDL model points in period',
     *       1x,i5,' Dynalysis points in period')
  213 format(1x,'standard deviation = ',f9.4)
```

Program Listing 2.4 Wl_sa.ph.f (continued)

```
!---------------------------------------------------------------------

c  Call subroutine compare to calculate rms difference.

c Variables :
c
c       wlm1 - TDL model wl's
c       wlm2 - Dynalysis model wl's
c
c     rmswl1 - rms(TDL) by day
c     rmswl2 - rms(Dynalysis) by day

           write(6,*)stanam(i)

           call compare(nmin,timem1,wlm1,timeo,wlo,avgwlo(nd,i),
     *          avgwlm1(nd,i),numc(nd,i),rms1sq_day(nd,i),rmswl1(nd,i))

           call compare(nmin,timem2,wlm2,timeo,wlo,avgwlo(nd,i),
     *          avgwlm2(nd,i),numc(nd,i),rms2sq_day(nd,i),rmswl2(nd,i))

!---------------------------------------------------------------------

        close (18)
        close (19)
        rewind (lunobs(i))

  900 continue

  600 continue

*******************************************************

c Write output for TDL (model 1) vs. observed, Dynalysis
c (model 2) vs. observed.


        write(9,849)monyr

        do 850 nd=1,ndays
           ndpage = mod(nd,3)
           if(nd.gt.1.and.ndpage.eq.1)write(9,12)formfd
           write(9,10)rjd_start(nd),rjd_stop(nd)
           write(9,801)title
           write(9,11)
           do 700 ns=1,istat
              write(9,701)stanam(ns),jpts(nd),avgwlo(nd,ns),
     *              sdwlo(nd,ns),jm1,avgwlm1(nd,ns),sdwlm1(nd,ns),
     *              numc(nd,ns),rmswl1(nd,ns),varim1(nd,ns)
  700      continue


c Write output for Dynalysis (model 2) vs. observed

           write(9,801)title2
           write(9,11)
```

Program Listing 2.4 Wl_sa.ph.f (continued)

```
         do 800 ns=1,istat
            write(9,701)stanam(ns),jpts(nd),avgwlo(nd,ns),
     *         sdwlo(nd,ns),jm2,avgwlm2(nd,ns),sdwlm2(nd,ns),
     *         numc(nd,ns),rmswl2(nd,ns),varim2(nd,ns)
  800    continue
  850 continue


  701 format(1x,a4,2x,i5,2x,2f7.3,4x,i5,2x,2f7.3,3x,i5,
     *       3x,f7.3,f9.3)
  801 format(a50)
  849 format(/,13x,'TDL vs DYNALYSIS Comparison, ',a13)

*****************************************************

c Calculate total number of points over comparison period,
c summation of dwl's for each model, by station.  Then
c calculate the rms value for both models over the entire
c month.
c
c Variables :
c
c   numc_tot() - total number points of comparison for station
c  rms1day_tot - summation of dwl's for TDL model
c  rms2day_tot - summation of dwl's for Dynalysis model
c   rms1_month - rms difference (TDL vs OBS)
c   rms2_month - rms difference (DYNALYSIS vs OBS)


      write(9,12)formfd
      write(9,1003)monyr

      do 1000 ns=1,istat
         numc_tot(ns) = 0
         rms1day_tot(ns) = 0.0
         rms2day_tot(ns) = 0.0
         nrms1_win = 0
         nrms2_win = 0
         write(10,1011)statnam(ns)

         do 1050 nd=1,ndays
            if(idebug.eq.1)write(10,*)numc(nd,ns)
            numc_tot(ns) = numc_tot(ns) + numc(nd,ns)
            if(idebug.eq.1)then
               write(10,*)rms1sq_day(nd,ns)
               write(10,*)rms2sq_day(nd,ns)
            endif
            if(rmswl1(nd,ns).lt.rmswl2(nd,ns))then
               nrms1_win = nrms1_win + 1
            else
               nrms2_win = nrms2_win + 1
            endif
            rms1day_tot(ns) = rms1day_tot(ns) + rms1sq_day(nd,ns)
            rms2day_tot(ns) = rms2day_tot(ns) + rms2sq_day(nd,ns)
 1050    continue
```

Program Listing 2.4 Wl_sa.ph.f (continued)

```
          write(10,1012)numc_tot(ns)
          write(10,1013)rms1day_tot(ns),rms2day_tot(ns)

          rms1_month(ns) = sqrt(rms1day_tot(ns)/numc_tot(ns))
          rms2_month(ns) = sqrt(rms2day_tot(ns)/numc_tot(ns))

          write(10,1014)rms1_month(ns),rms2_month(ns)
          write(9,1004)statnam(ns),rms1_month(ns),nrms1_win,
     *                rms2_month(ns),nrms2_win
 1000 continue


 1003 format(/,13x,'TDL vs DYNALYSIS Comparison, ',a13,//,
     *        18x,'TDL',17x,'DYNALYSIS',/,18x,'rms',17x,'rms')
 1004 format(1x,a13,f8.4,i6,6x,f8.4,i6)
 1011 format(/,' Station ',a13',')
 1012 format(' A total of ',i3,' points ',
     *       'of comparison')
 1013 format(2(2x,f10.4))
 1014 format(2(2x,f10.4))

*********************************************************

c     Format Statements

   10 format(///,x,'start time =',f9.4,2x,'stop time =',f9.4,/)
   11 format(x,'stat  num obs   mean     sd   num mod    mean',
     *  '         sd      num     rms diff  variability')
   12 format(a1)


      stop
      end

*********************************************************
*********************************************************

      subroutine compare(jmin,timem,wlm,timeo,wlo,avgwlo,
     *           avgwlm,num,rms_sqr,rms)


      parameter(nd=20000)

      dimension timem(nd),wlm(nd)
      dimension timeo(nd),wlo(nd)

      common/time/tdmax


*  initialize
      rms = 0.0
      num = 0
      dwl = 0.0

*  find how many points to compare
      do 1500 i=1,jmin
```

Program Listing 2.4 Wl_sa.ph.f (continued)

```
            td = abs(timeo(i)-timem(i))
            if(td.lt.tdmax)then
               num = num + 1
               if(num.eq.1)write(6,21)timeo(i),timem(i)
               dwl = wlo(i) - wlm(i)
               rms = rms + dwl**2
            else
               write(6,22)timeo(i),timem(i),tdmax
c              write(9,22)timeo(i),timem(i),tdmax
               stop
            end if
 1500   continue

        rms_sqr = rms
        rms = sqrt(rms/float(max0(1,num)))

        write(6,301)num


     21 format(1x,'first observed data point ',f10.4,/,
       *         1x,'first model data point ',f10.4)
     22 format(1x,'Time difference exceeds tdmax',/,
       *         1x,'observed time = ',f8.3,2x,'model time =',
       *         1x,f8.3,f10.3,/,6x,'program terminated')
    301 format(1x,i6,' Pts of Comparison',/)


        return
        end


******************************************************************

        subroutine ncrght (line,nc)
*
*   Returns the last non blank character position in a string
*
        character*(*) line
        ilim = len (line)
        do 100 i = 1,ilim
          if(line(i:i) .ne. ' ') then
            nc = i
          end if
    100 continue
        return
        END
```

Program Listing 2.4 Wl_sa.ph.f (continued)

2.5 Program Plot_wlanal.pro

The listing for plot_wlanal.pro is given in Program Listing 2.5. This is an IDL program used to plot a month of observed water level data, along with points from each of the daily forecasts. From each daily forecast, four points are plotted : the start, the end, the max, and the min. The symbols used to represent forecast values include pluses, triangles, squares, and asterisks.

From the control file is read ptype, idebug, stat_name, titlnam, strttime, and endtime. Ptype is for plot type, in this case postscript. Idebug controls the debug function. Stat_name is the station name, titlnam is the plot title. Strttime and endtime specify start and end times.

Plot_wlanal.pro is a conventional IDL program. The "plot" command is used to plot the observed curve, while oplot is used to plot the forecast points. The plots are annotated with a title, station name, and a legend.

```
; Program : plot_wlanal.pro
;
; Purpose : This program makes use of IDL graphics
; and is written in the IDL language.  The program
; plots observed time series water levels on one
; plot (per page).  The program plots points from
; the forecast model on the same plot.
;    The program contains an option to print a legend.
; For postscript ('ps') plots, there is an option for
; either landscape or portrait.
; The program also contains an option for the use of
; Julian dates or calendar days to define the time axis.
;
;
; Language : IDL
;
; Version date : March 3, 2003
;
; Location : On OPSEA, /usr/people/philr/dynalysis/plot
;
; Author : Phil Richardson

;**************************************************

   im = 2000
   numdays = 31


  filemod = ' '
  filedata = ' '
  legend = ' '
  cntrl_file = ' '
  time_axis = ' '
  y_axis = ' '
  stat_name = ' '
  ptype=' '
  plottype=' '
  rmspr = ' '
  titlnam = ' '
  time_opt = ' '

; Initialize Integer Variables
   idebug = 0
   nticks = 0
   iyear = 0
   ndays = 0

; Dimension arrays
   lunmod=intarr(numdays)
   t=fltarr(im)


   legnd = strarr(2)
   filemodl = strarr(numdays)

   wlplt = fltarr(im)
```
Program Listing 2.5 Plot_wlanal.pro

```
  xpos=fltarr(2)
  y1=fltarr(2)
  x1=fltarr(2,2)

  time_strt = fltarr(2)

  xst=fltarr(numdays)
  xf=fltarr(numdays)
  yst=fltarr(numdays)
  yf=fltarr(numdays)

  xhigh=fltarr(numdays)
  xlow=fltarr(numdays)
  yhigh=fltarr(numdays)
  ylow=fltarr(numdays)

;*************************************************

; Open control file, read from control file

;        ptype - x, ps, or tek
;    idebug = 1, times (Julian dates)
;           = 2, EOF result
;           = 3, plotting of Legend
;    stat_name - station name
;     strttime - start time (Julian date)
;      endtime - end time (Julian date)
;        ndays - number of daily forecast files to read
;       nticks - number of tick marks (time axis)
;        iyear - year of plot
;       ilegnd - option to print legend
;        legnd - character string, for legend
;      filedat - obs data filename
;      filemod - model data filename
;     time_opt - calendar day or Julian day
;    time_axis - time axis name
;       y_axis - Y axis name

  print,'Enter name of control file '
  read,cntrl_file
  openr,1,cntrl_file

  readf,1,ptype
  if(ptype eq 'ps')then begin
     readf,1,plottype
  endif
  readf,1,idebug
  readf,1,stat_name
  readf,1,titlnam
  readf,1,strttime
  readf,1,endtime
  readf,1,ndays
  ndym1 = ndays - 1
  readf,1,nticks
  readf,1,ilegnd
```

Program Listing 2.5 Plot_wlanal.pro (continued)

```
  readf,1,ymin,ymax,ytcks
  readf,1,time_opt
  readf,1,time_axis
  readf,1,y_axis

  if(ilegnd gt 0)then readf,1,legend
  legnd = legend
  readf,1,filedata
  filedat = filedata

  for nd=0,ndym1 do begin
      readf,1,filemod
      filemodl(nd) = filemod
      print,filemodl(nd)
  endfor

  close,1

;-----------------------------------------------------

; set plot type : x, ps, or tek
  set_plot,ptype

; set the plot scaling
  aspect=1.5
  isize = 1024
  jsize = 1200


  xs=8.0
  ys=8.0*aspect

 if(ptype eq 'ps')then begin
    if(plottype eq 'portrait')then begin
       device, xsize=xs,$
               ysize=ys,/inch,xoffs=0.25,yoffs=0.
    endif
    if(plottype eq 'landscape')then begin
       device, ysize=10.0, /landscape,$
              /inches, xoffs=-2.0
    endif
 endif

;*******************************************************

; Open observed wl data file.
;
; Read data from OBS file
;
; variables :
;   ndatpts - number of data points


  print,time_opt,format='("time option is ",a4)'

```

Program Listing 2.5 Plot_wlanal.pro (continued)

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


  get_lun, lun
  openr,lun,filedat,error=err

  if(err ne 0) then begin
     print, !err_string
     goto, ENDPROG
  endif

  print,filedat,    $
        format='(1x,"file",a66)'

;*******************************************************

     if(ptype eq 'x')then begin
        window,0,xsize=isize,ysize=jsize
     endif
     wlevel_tot = 0.0
     ncount = 0
     if(idebug eq 1)then begin
        printf,4,filedat,format='(1x,a64)'
     endif

;*******************************************************

; Open daily forecast files.  Loop thru days from nd=0 to ndym1,
; read from daily forecast files.


  for nd=0,ndym1 do begin
      get_lun, lunm
      lunmod(nd) = lunm
      openr,lunmod(nd),filemodl(nd),error=err
      if(err ne 0)then begin
         print, !err_string
         goto, ENDPROG
      endif
      print,filemodl(nd),    $
            format='(1x,"file ",a71)'

      wlmin = ymax
      wlmax = ymin
      nptm = 0

      READMOD: readf,lunmod(nd),timem,wlm
      resultm = EOF(lunmod(nd))
      if(idebug eq 2)then print,timem,wlm,resultm

      if(resultm lt 1)then begin
         if(nptm eq 0)then begin
            timem1 = timem
            xst(nd) = timem1
            yst(nd) = wlm
            print,timem1,format='("model file begins at ",f8.3)'
```

Program Listing 2.5 Plot_wlanal.pro (continued)

```
            print,wlm,format='("water level = ,",f8.3)'
         endif

         if(wlm gt wlmax)then begin
            xhigh(nd) = timem
            wlmax = wlm
         endif
         if(wlm lt wlmin)then begin
            xlow(nd) = timem
            wlmin = wlm
         endif

         nptm = nptm + 1
         goto, READMOD
      endif

      if(resultm gt 0)then begin
         xf(nd) = timem
         yf(nd) = wlm
         if(wlm gt wlmax)then begin
            xhigh(nd) = timem
            wlmax = wlm
         endif
         if(wlm lt wlmin)then begin
            xlow(nd) = timem
            wlmin = wlm
         endif
         print,timem,format='("End of model file reached at time",f8.3)'
      endif

      print,nptm,wlmin,wlmax

      yhigh(nd) = wlmax
      ylow(nd) = wlmin


      free_lun, lunm
   endfor

;*******************************************************

; Read from observed data file

     readf,lun,time
     print,time,    $
     format='(/,1x,"file (",i1,") starts at time =",f8.3)'
     point_lun,lun,0

     READDATA: readf,lun,time,wlevel
     result = EOF(lun)
     if(idebug eq 2)then print,result
     if(time lt strttime)then goto, READDATA
     if(time gt endtime)then begin
        ncount = ncount - 1
        ndatpts = ncount + 1
        goto, ENDLOOP
```

Program Listing 2.5 Plot_wlanal.pro (continued)

```
      endif
      if(ncount eq 0)then begin
         print,lun,time,    $
         format='(1x,"start time (obs) file (",i1,") = ",f8.3)'
         time_strt = time
      endif

      wlevel_tot = wlevel_tot + wlevel


;     times (Julian dates) from year 1995, relative
;     to 1993.
      if(time lt 1096.0) and (time gt 731.0)then begin
         jd_offset = 730.0
      endif

;     times (Julian dates) not referenced to 1993
      if(time lt 366.0)then begin
         jd_offset = 0.0
      endif

;     times (Julian dates) from year 1998
      if(time gt 1826.9)then begin
         jd_offset = 1826.0
      endif

      time = time - jd_offset
      if(result lt 1)then begin
         if(idebug eq 1)then printf,4,ncount,time
         t(ncount) = time
         wlplt(ncount) = wlevel
         ncount = ncount + 1
         goto, READDATA
      endif
      if(result gt 0)then begin
         if(idebug eq 1)then printf,4,ncount,time
         print,lun,         $
               format='(" End of file (",i1,") reached")'
         t(ncount) = time
         wlplt(ncount) = wlevel
      endif
   close,lun
   free_lun, lun

   ndatpts = ncount + 1
   ENDLOOP: print,ndatpts,       $
           format='(i4," data points, End of loop")'
   numb_pts = ndatpts

;  Calculate mean
   rmean = wlevel_tot/ndatpts


   ncount = ndatpts - 1
   print,ncount,format='(/,1x,i4)'

```

Program Listing 2.5 Plot_wlanal.pro (continued)

```
  print,ndatpts,format='(1x,i4)'

!p.multi=[0,0,1]

;-------------------------------------------------------

; make the plot

!P.CHARSIZE=1.0

  if(time_opt ne 'juld')then begin
     dummy = label_date(date_format='%D')
  endif



@plot01
  plot,t[0:ncount],wlplt[0:ncount],        $
      title = titlnam,             $
      yrange=[ymin,ymax],          $
      xtitle=time_axis,            $
      ytitle=y_axis,               $
      xmargin=[0,0],               $
      ymargin=[0,0],               $
      xstyle=1,ystyle=1,           $
      linestyle=0,                 $
;    xtickformat = 'Label_date',$
      xticks = nticks,             $
      yticks = ytcks,              $
      position=[0.10,0.52,0.90,0.87]
      for nd=0,ndym1 do begin
          if(nd eq 0)then isymb = 1
          if(nd eq 1)then isymb = 2
          if(nd eq 2)then isymb = 5
          if(nd eq 3)then isymb = 6
          if(nd eq 4)then isymb = 1
          if(nd eq 5)then isymb = 2
          if(nd eq 6)then isymb = 5
          if(nd eq 7)then isymb = 6
          if(nd eq 8)then isymb = 1
          if(nd eq 9)then isymb = 2
          if(nd eq 10)then isymb = 5
          if(nd eq 11)then isymb = 6
          if(nd eq 12)then isymb = 1
          if(nd eq 13)then isymb = 2
          if(nd eq 14)then isymb = 5
          if(nd eq 15)then isymb = 6
          if(nd eq 16)then isymb = 1
          if(nd eq 17)then isymb = 2
          if(nd eq 18)then isymb = 5
          if(nd eq 19)then isymb = 6
          if(nd eq 20)then isymb = 1
          if(nd eq 21)then isymb = 2
          if(nd eq 22)then isymb = 5
          if(nd eq 23)then isymb = 6
          if(nd eq 24)then isymb = 1
```

Program Listing 2.5 Plot_wlanal.pro (continued)

```
              if(nd eq 25)then isymb = 2
              if(nd eq 26)then isymb = 5
              if(nd eq 27)then isymb = 6
              if(nd eq 28)then isymb = 1
              if(nd eq 29)then isymb = 2
              oplot,xhigh[nd:nd],yhigh[nd:nd],psym=isymb,symsize=1.0
              oplot,xlow[nd:nd],ylow[nd:nd],psym=isymb,symsize=1.0
              oplot,xst[nd:nd],yst[nd:nd],psym=isymb,symsize=1.0
              oplot,xf[nd:nd],yf[nd:nd],psym=isymb,symsize=1.0
          endfor

   xyouts,0.50,0.55,stat_name,size=1.5,/normal, alignment=0.5

 ;*******************************************************

 ; Draw Legend

  if(idebug eq 3)then begin
     print,strttime,endtime,jd_offset
  endif


 ; Establish x,y coordinates for legend

  x1(0,0) = 0.38
  y1(0) = 0.825
  x1(0,1) = 0.46
  y1(1) = 0.825
  x1(1,0) = 0.61
  x1(1,1) = 0.69

   if(idebug eq 3)then begin
      print,format='(/,3x,"For plotting of Legend -")'
      print,x1(0,0),x1(0,1),format='(3x,"x1 points :",2f7.1)'
      print,y1(0),     $
      format='(3x,"Y position (data coordinate) is",f7.3)'
   endif


   xpos(0) = 0.25
   xpos(1) = 0.50
   ypos = 0.83
   if(ilegnd gt 0)then begin
      xyouts,xpos(0),ypos,legnd(0),size=1.4,/NORMAL
         linesty=0
         plots,[x1(0,0),x1(0,1)],y1,linestyle=linesty, $
               /normal
   endif

 ;-------------------------------------------------------

   if(ptype eq 'ps') then device,/close

   if(ptype eq 'ps')then begin
      spawn,' lp -dqms2 idl.ps'
   endif
```

Program Listing 2.5 Plot_wlanal.pro (continued)

```
ENDPROG:
end
```

Program Listing 2.5 Plot_wlanal.pro (continued)

3. ANALYSIS PROCEDURE

To perform the comparison analysis, each program is executed in the order shown in Table 3.1. For each program, a separate directory is recommended as shown. ~ designates the users home area, and dynalysis is the project directory.

~/dynalysis/observed/reform_coops.f
~/dynalysis/dynal/read_dyn.f
~/dynalysis/tdl/read_tdl.f
~/dynalysis/compar/adjust.f
~/dynalysis/compar/wl_sa.ph.f
~/dynalysis/wlevel/plot/plot_wlanal.pro

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

Script	Source File	Example Control file
reform.sh	reform_coops.f	reform.n
readtdl.sh	read_tdl.f	read_nov02.n
readdyn.sh	read_dyn.f	readdyn.n
adjust.sh	adjust.f	adj_tdl.nov02.n
wl_sa.sh	wl_sa.ph.f	wl.nov02.n
	plot_wlanal.pro	cnt.pleas_dyn.nov02

Listings for script and control files are provided in Appendix A. The IDL plot program does not have a script file. To run the IDL program, type idl <return>, then type .r plot_wlanal.pro <return>.

4. NOVEMBER 2002

The observed subtidal water level plot at Galveston Pleasure Pier for November 2002 depicts numerous events. We chose the high water event of November 2 through November 4, Julian dates 306.0 through 309.0, for further examination. The observed water level plot is presented in Figure 4.1. The daily statistical tables, for Julian days 305.75 through 309.0, are presented in Table 4.1.

The observed mean values from Table 4.1 indicate that the event reaches its peak during the day beginning on Julian date 306.75. The observed mean is 1.454 feet at Pleasure Pier. The rms difference between the ETSS model forecast and the observed values is 0.200 feet. The rms difference between the DGOM model forecast and the observed values is 0.231 feet. Clearly, on this occasion, the DGOM forecast and the ETSS forecast are comparable. More evidence of this can be seen by looking at the event plots in Figures 4.2 and 4.3. In this instance, the DGOM model slightly under predicts the observed peak.

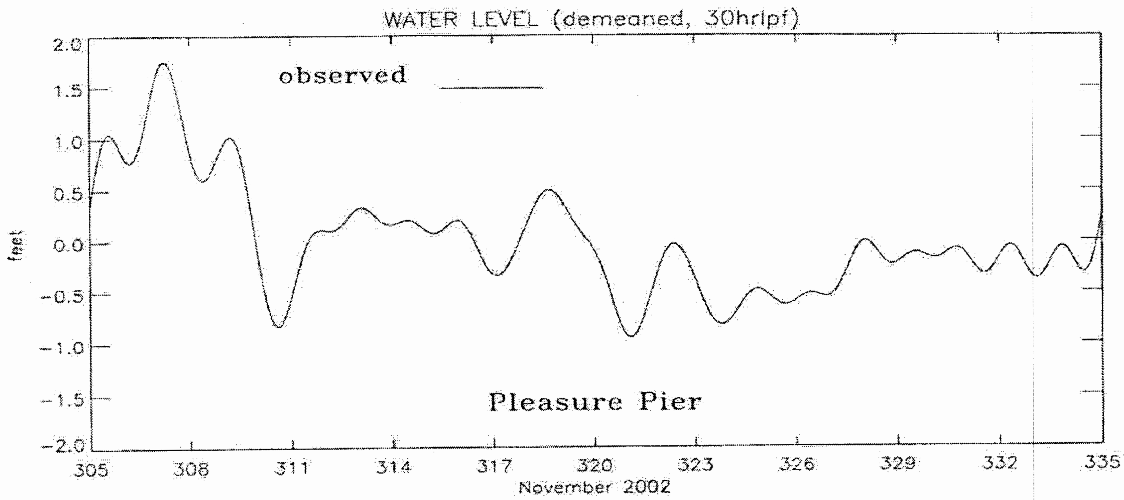


Figure 4.1 Observed Subtidal Water Level at Galveston Pleasure Pier, TX, November 2002

The water level analysis summary table (Table 4.2) indicates that the DGOM model forecasts and the ETSS model forecasts are comparable for the month of November. At all three stations, the difference in rms values (ETSS model rms - DGOM model rms) is less than 5/100 of a foot. At Pleasure Pier, the two forecast models perform almost equally well. Npf is defined to be the number of times that the given model produces the better forecast in terms of rms error. At Naples, the ETSS model forecast is better thirteen times to three for the DGOM model. This comparison is somewhat misleading because the differences in rms error are quite small.

Table 4.1 Water Level Event Analysis :
November 1 through 4, 2002

Note : all dimensioned quantities are in feet.

TDL vs DYNALYSIS Comparison, November 2002

start time = 305.7500 stop time = 307.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.165	0.087	31	0.089	0.177	31	0.125	2.042
pens	31	0.146	0.077	31	0.301	0.062	31	0.168	0.805
plea	31	1.036	0.268	31	1.095	0.086	31	0.245	0.321

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.165	0.087	31	0.126	0.174	31	0.148	2.006
pens	31	0.146	0.077	31	0.305	0.044	31	0.173	0.579
plea	31	1.036	0.268	31	1.093	0.161	31	0.309	0.599

start time = 306.7500 stop time = 308.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.362	0.097	31	0.343	0.128	31	0.141	1.325
pens	31	0.299	0.078	31	0.237	0.116	31	0.091	1.477
plea	31	1.454	0.271	31	1.382	0.242	31	0.200	0.895

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.362	0.097	31	0.247	0.127	31	0.169	1.307
pens	31	0.299	0.078	31	0.180	0.071	31	0.128	0.909
plea	31	1.454	0.271	31	1.255	0.185	31	0.231	0.683

start time = 307.7500 stop time = 309.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.436	0.018	31	0.572	0.085	31	0.155	4.644
pens	31	0.549	0.077	31	0.342	0.057	31	0.229	0.748
plea	31	0.796	0.173	31	0.680	0.227	31	0.136	1.313

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.436	0.018	31	0.281	0.172	31	0.230	9.402
pens	31	0.549	0.077	31	0.590	0.112	31	0.082	1.458
plea	31	0.796	0.173	31	0.837	0.180	31	0.169	1.044

Note : napl = Naples, Fl, pens = Pensacola, Fl, and plea = Galveston Pleasure Pier, TX.

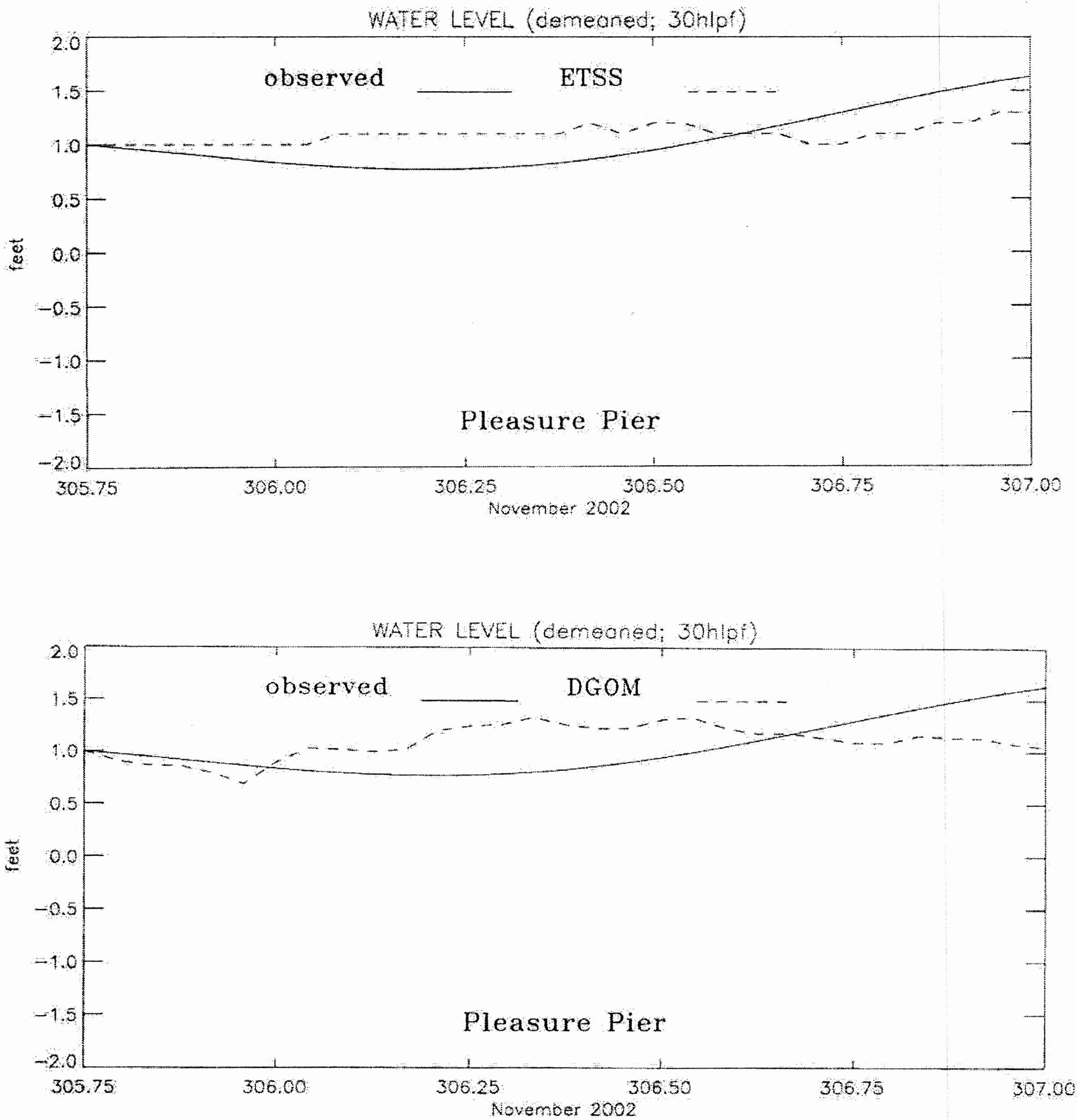
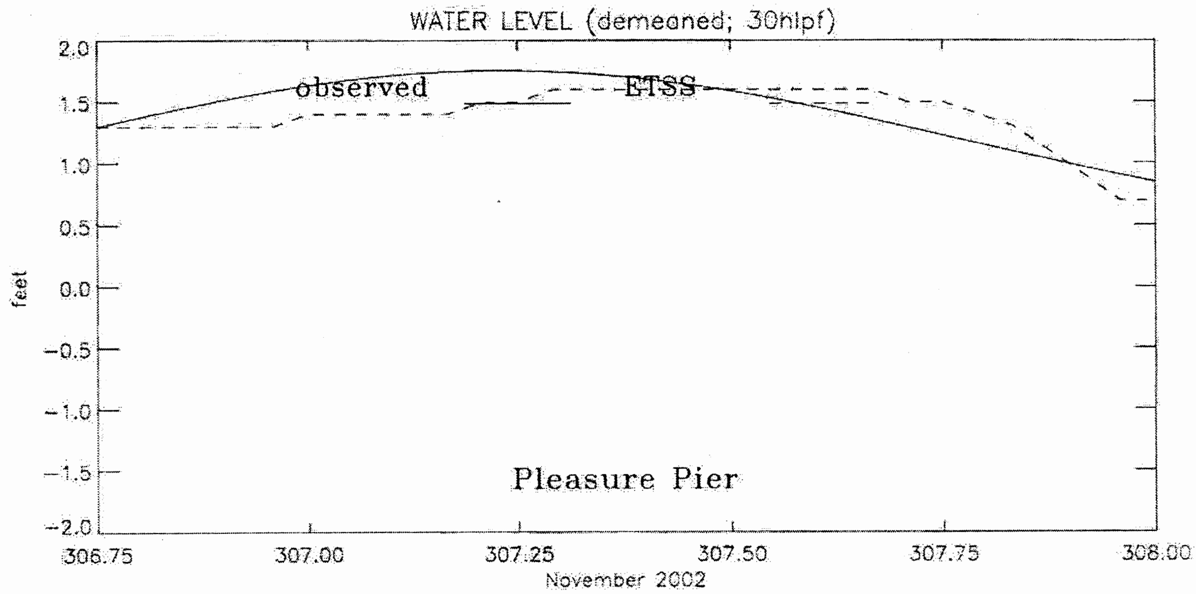


Figure 4.2 Simulated vs. Observed Water Levels from November 2002, Day 1, at Galveston Pleasure Pier, TX



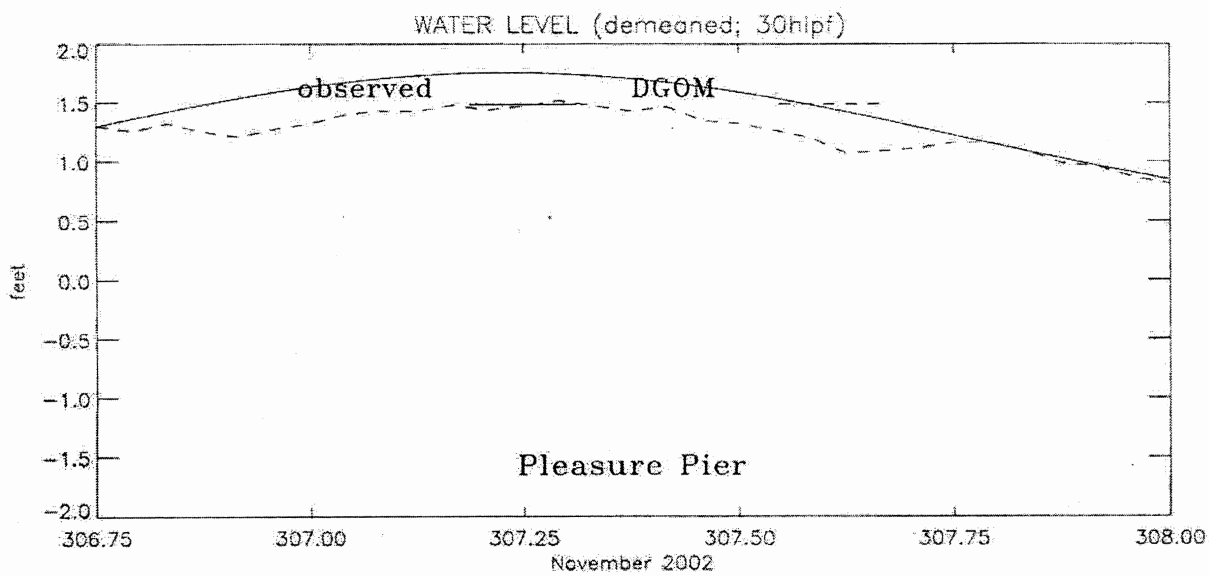


Figure 4.3 Simulated vs. Observed Water Levels from November 2002, Day 2, at Galveston Pleasure Pier, TX

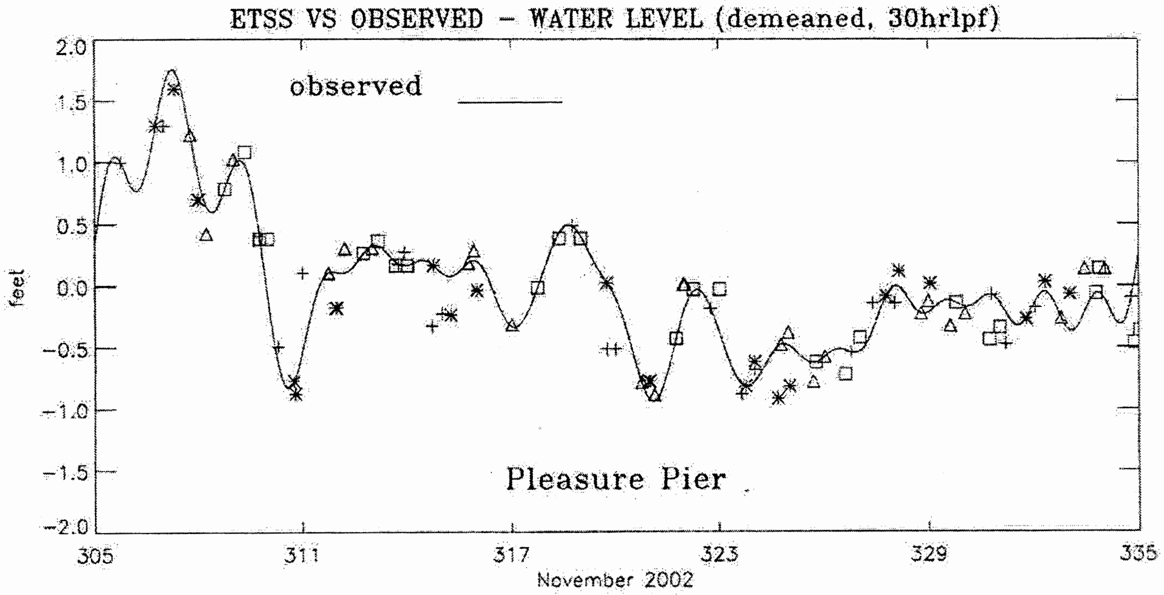
Table 4.2 Water Level Analysis Summary for November 2002

ETSS vs. DGOM Model Water Level Comparison

	ETSS		DGOM	
	rmse (ft)	npf (-)	rmse (ft)	npf (-)
Naples	0.2465	13	0.2917	3
Pensacola	0.1744	10	0.2030	6
Pleasure Pier	0.2146	9	0.2171	7

Note : rmse is root mean square error and npf is defined to be the number of preferred forecasts.

More evidence of the comparability of the two forecast models can be seen by looking at the Forecast vs. Observed plots, Figures 4.4 through 4.6. The solid curve represents the observed water level values. For each daily forecast, four points are plotted. These points are the initial point, the end point, the max and the min. Water level values at Pleasure Pier are depicted in Figure 4.4. Both the ETSS model and the DGOM model forecast points seem to have some scatter about the observed curve. Naples, Figure 4.5, is a similar story. The DGOM model forecasts have some scatter about the observed curve. The ETSS model seems to have less general scatter, but tends to over-predict major events. At Pensacola, Figure 4.6, the DGOM forecast points seem to hug the observed curve very closely. The ETSS forecast points seem to have a bit more scatter, even though the rms error for the ETSS forecast model is better by about 3/100 of a foot.



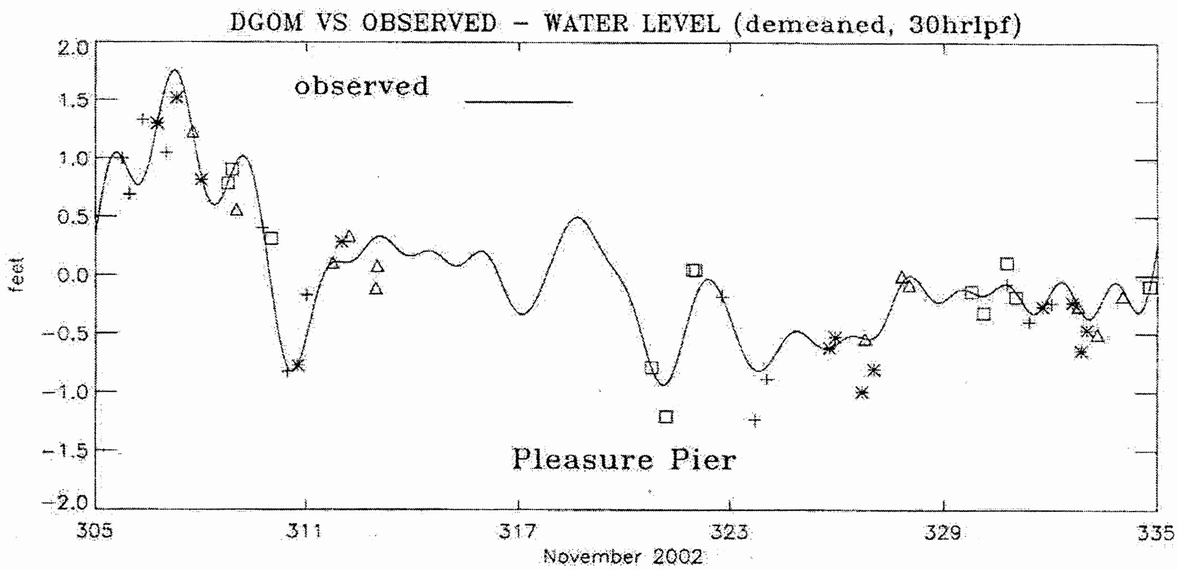


Figure 4.4 Forecast vs. Observed Water Levels at Galveston Pleasure Pier, TX, November 2002
Note : Symbols used to represent forecast points include the plus, square, triangle, and asterisk. The points plotted for each daily forecast include the high, low, start, and end point.

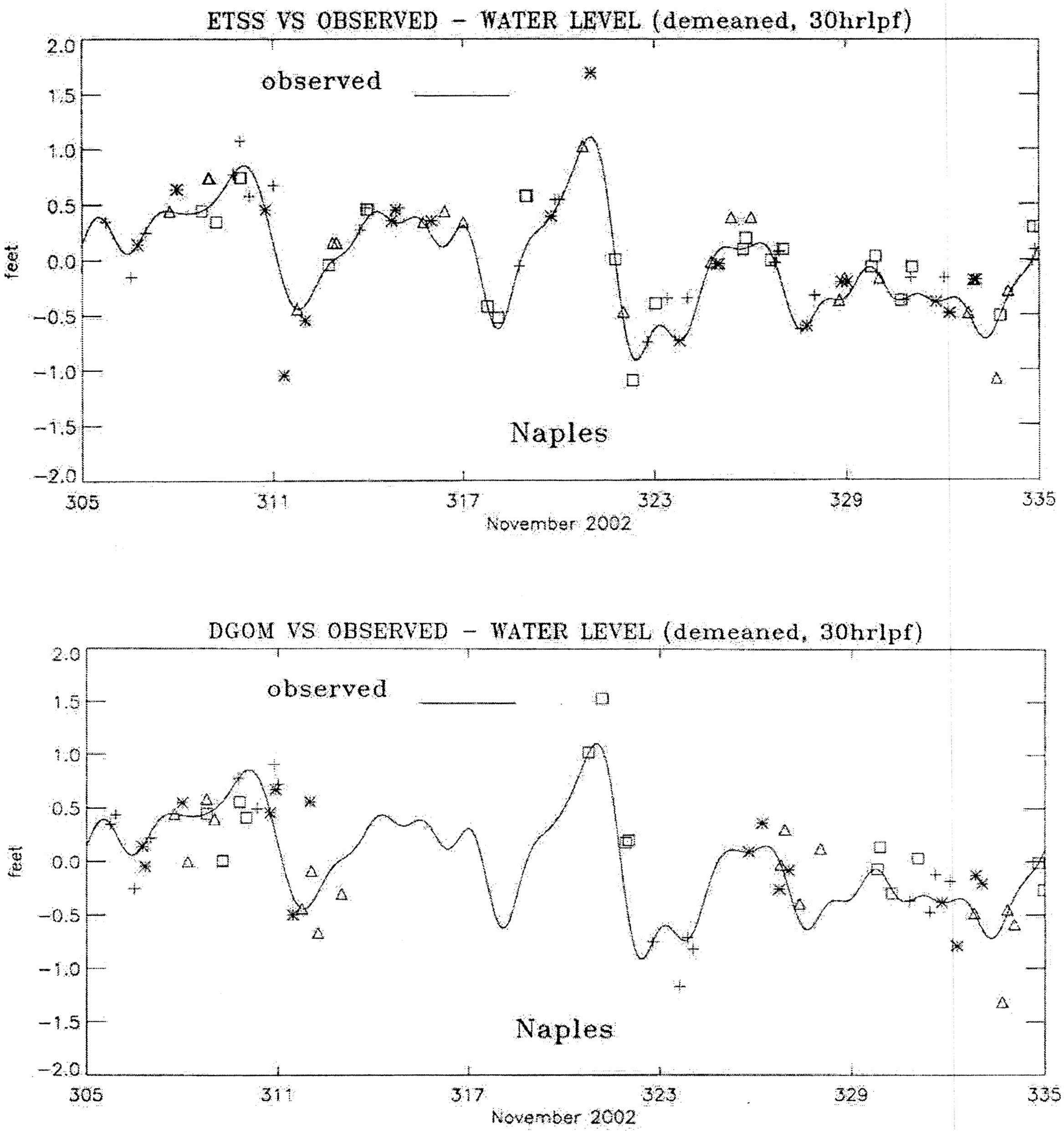
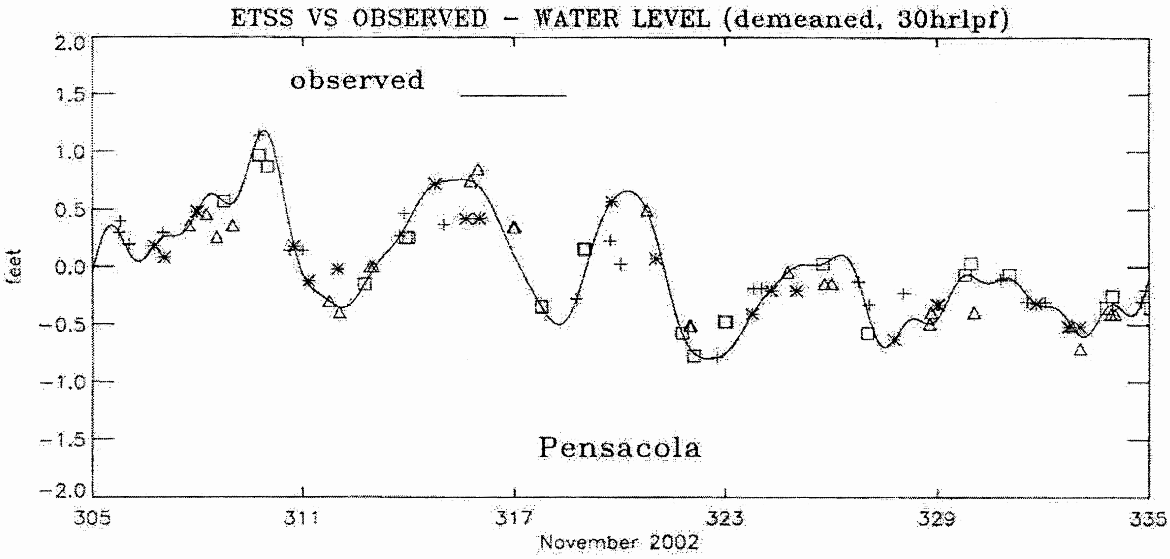


Figure 4.5 Forecast vs. Observed Water Levels at Naples, FL, November 2002



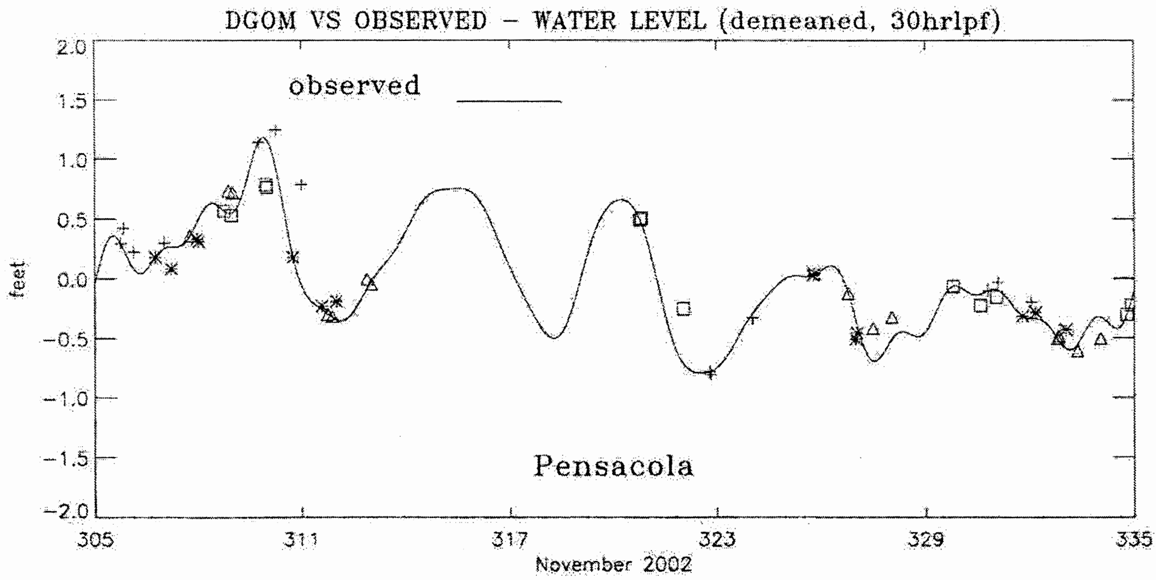


Figure 4.6 Forecast vs Observed Water Levels at Pensacola, FL, November 2002

Residual Analysis

The November analysis was redone using observed data which had been de-tided, as opposed to 30 hour low-pass filtered. The observed data were de-tided using the Program Pred (Zervas, 2003).

Table 4.3 Water Level Analysis Summary for 30 Hour Low-Pass and Water Level Residual, November 2002

ETSS vs. DGOM Model Comparison, November 2002

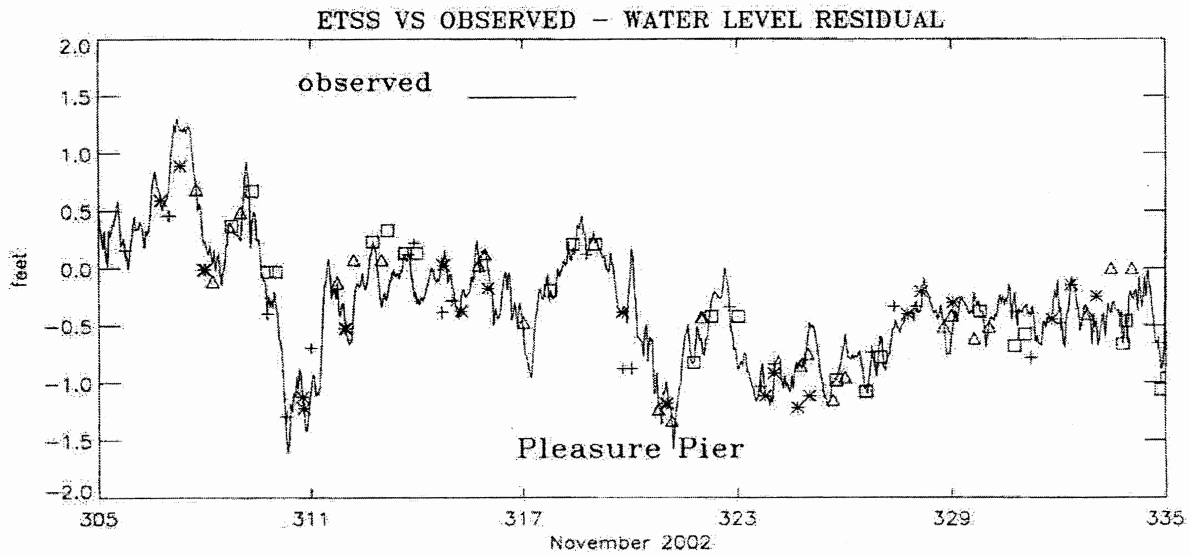
	ETSS rmse (ft)	npf (-)	DGOM rmse (ft)	npf (-)
Naples	0.2465	13	0.2917	3
Pensacola	0.1744	10	0.2030	6
Pleasure Pier	0.2146	9	0.2171	7

ETSS vs. DGOM Comparison, November 2002
(Observed data has been detided, not 30hrlpf)

	ETSS rmse (ft)	npf (-)	DGOM rmse (ft)	npf (-)
Naples	0.3134	10	0.3453	6
Pensacola	0.2224	8	0.2432	8
Pleasure Pier	0.2512	8	0.2611	8

Note : rmse is root mean square error and npf is defined to be the number of preferred forecasts.

The table indicates that the performance of both models degrades when residuals are used. Corresponding station plots are given in Figures 4.7 through 4.9, respectively.



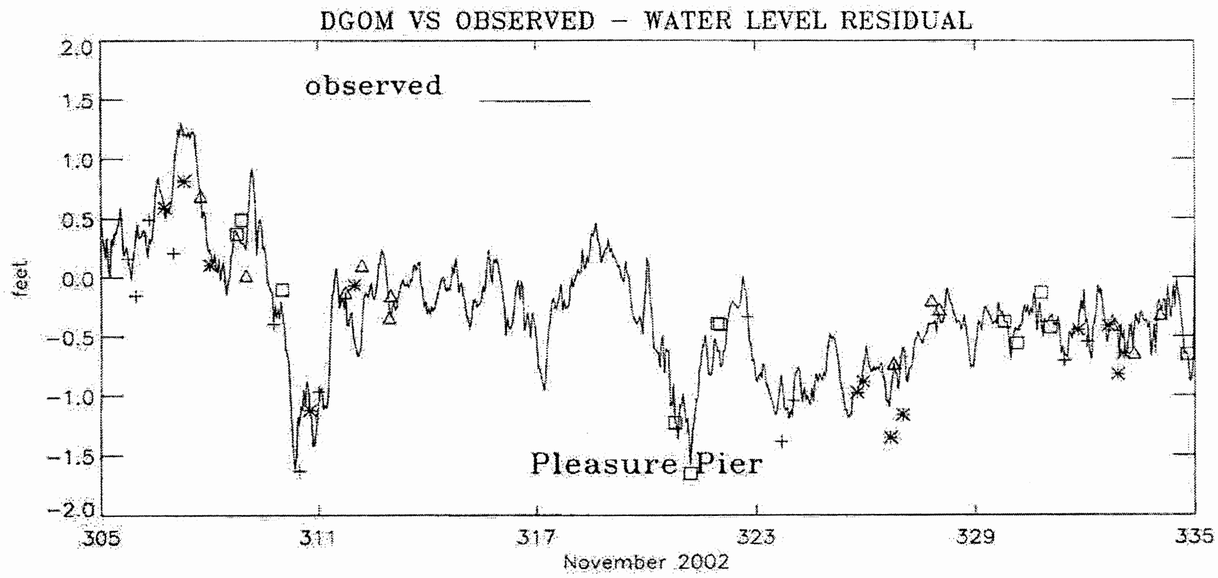
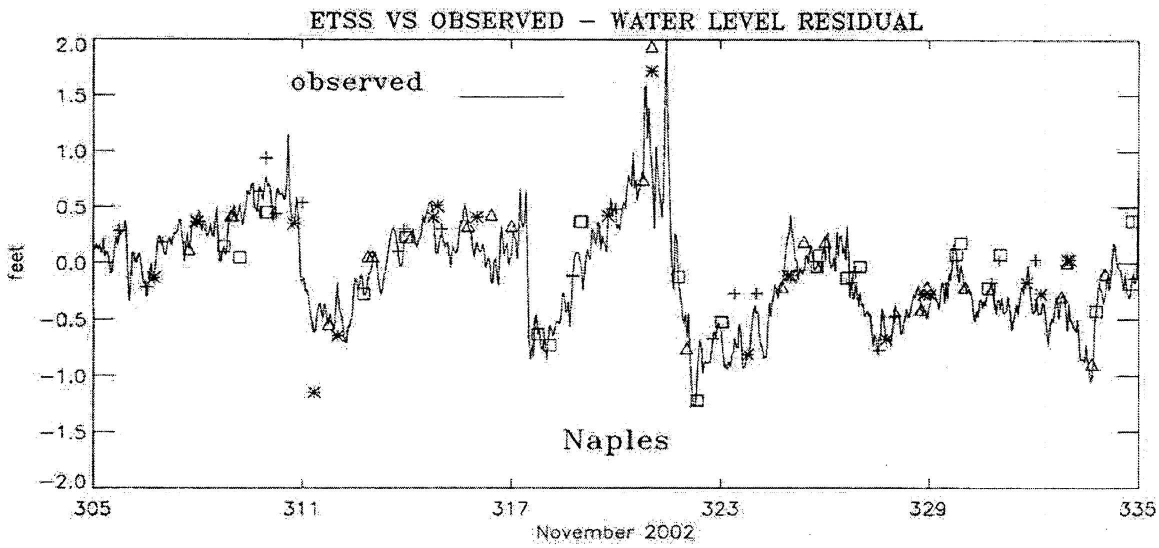


Figure 4.7 Forecast vs. Observed (residual) Water Levels at Galveston Pleasure Pier, TX, November 2002



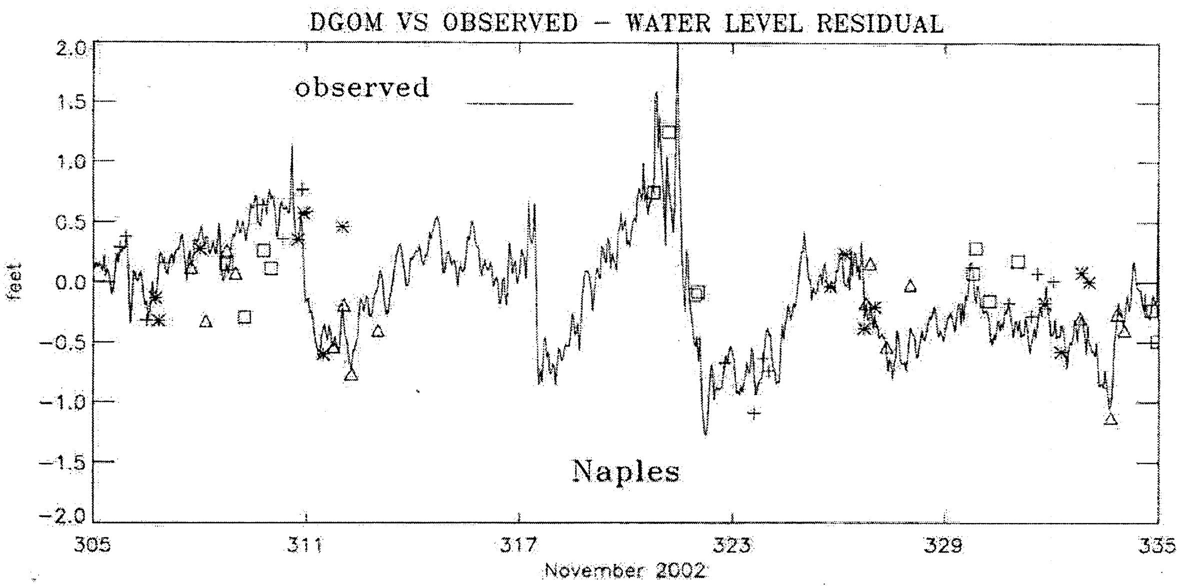
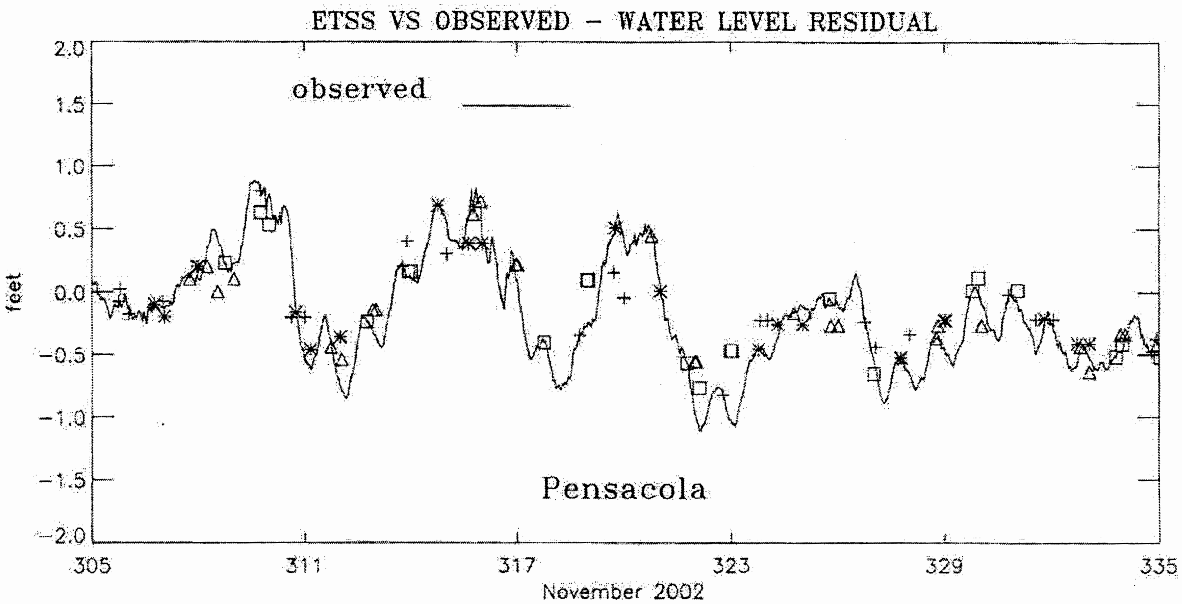


Figure 4.8 Forecast vs. Observed (residual) Water Levels at Naples, FL, November 2002



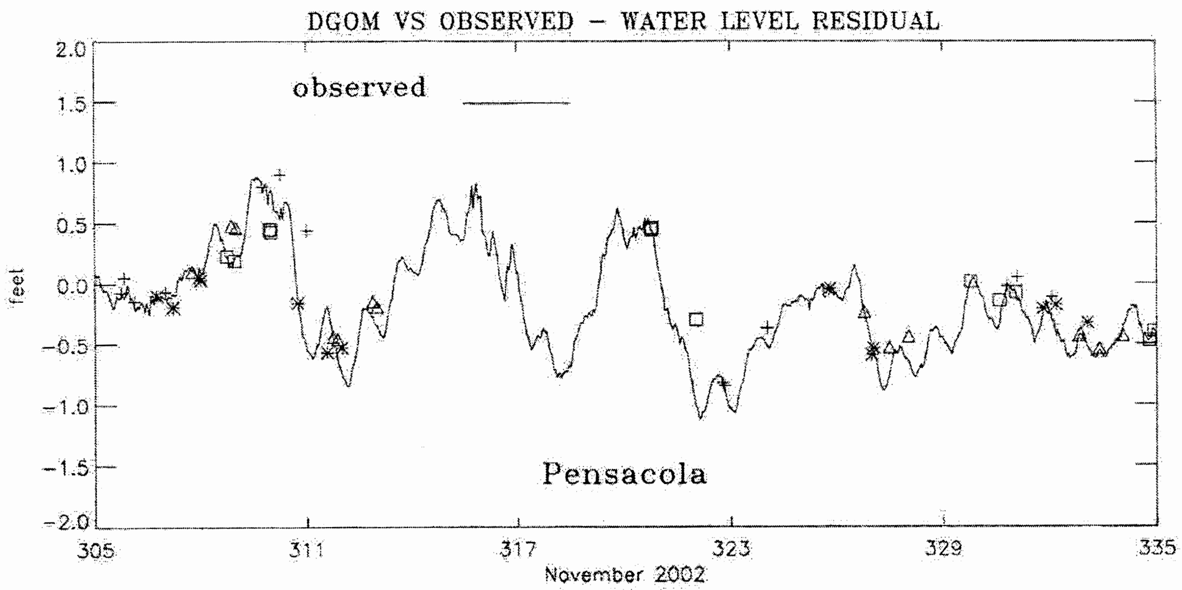


Figure 4.9 Forecast vs. Observed (residual) Water Levels at Pensacola, FL, November 2002

5. JANUARY 2003

The observed subtidal water level plot at Galveston Pleasure Pier for January 2003 depicts significant events. Though events are somewhat less numerous than in November 2002, those occurring during January are more extreme. We chose the low water event occurring between January 16 and January 19 for further examination. The observed subtidal water level at Galveston Pleasure Pier is presented in Figure 5.1. The daily statistical tables, for Julian dates 14.75 through 18.00, are shown in Table 5.1. The statistical tables for Julian dates 17.75 through 22.00 are presented in Table 5.2.

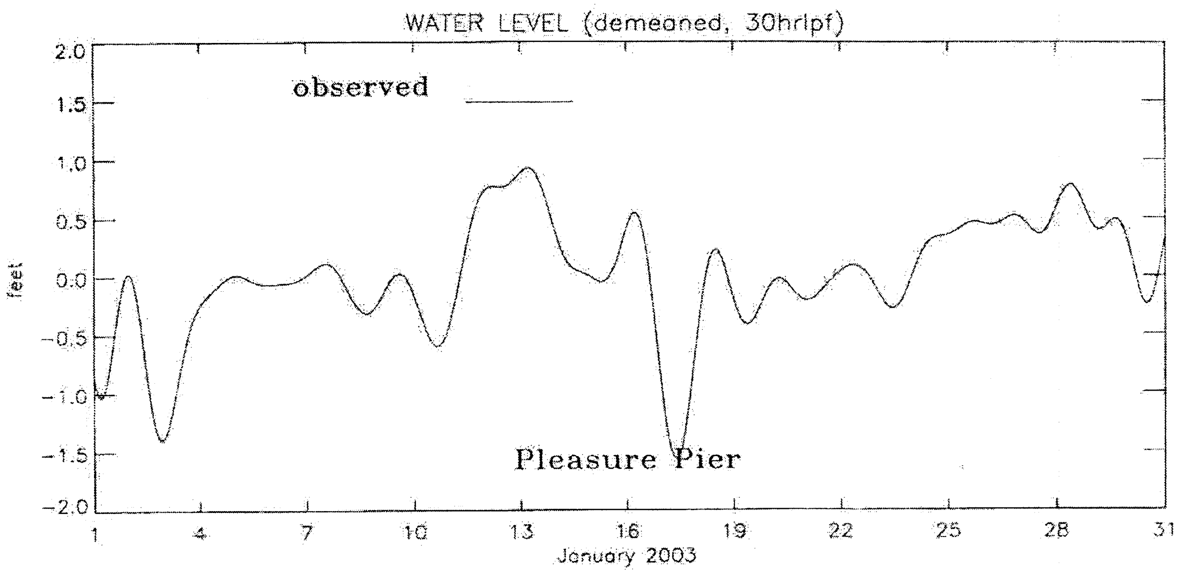


Figure 5.1 Observed Subtidal Water Level at Galveston Pleasure Pier, TX, January 2003

From Table 5.1 we see that the event reaches its low value peak on the day beginning at 16.75 and ending at 18.00. The mean value over this time period at Galveston Pleasure Pier is -1.083 feet. The actual low value peak is about -1.5 feet, making this a fairly extreme event. The DGOM model forecast has an rms error of 0.460 feet over this time period. The rms error of the ETSS model forecast is somewhat higher, 0.659 feet. Looking at Figure 5.2 and Figure 5.3, it is clear that the DGOM model forecasts stay closer to the observed curve.

Table 5.1 Water Level Event Analysis :
January 14 through 18, 2003

Note : all dimensioned quantities are in feet.

start time = 14.7500 stop time = 16.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	-0.027	0.048	31	-0.047	0.156	31	0.148	3.279
pens	31	-0.043	0.110	31	-0.237	0.107	31	0.255	0.970
plea	31	0.066	0.137	31	0.047	0.106	31	0.064	0.776

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	-0.027	0.048	31	-0.277	0.253	31	0.366	5.318
pens	31	-0.043	0.110	31	-0.092	0.061	31	0.136	0.555
plea	31	0.066	0.137	31	0.157	0.119	31	0.134	0.873

start time = 15.7500 stop time = 17.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.058	0.173	31	0.389	0.263	31	0.354	1.522
pens	31	0.058	0.255	31	0.143	0.161	31	0.154	0.634
plea	31	0.126	0.444	31	-0.031	0.649	31	0.270	1.462

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.058	0.173	31	-0.183	0.128	31	0.274	0.738
pens	31	0.058	0.255	31	0.006	0.101	31	0.170	0.398
plea	31	0.126	0.444	31	0.088	0.371	31	0.103	0.835

start time = 16.7500 stop time = 18.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.135	0.194	31	0.333	0.229	31	0.251	1.177
pens	31	-0.002	0.343	31	-0.077	0.298	31	0.267	0.870
plea	31	-1.083	0.400	31	-0.603	0.435	31	0.659	1.089

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.135	0.194	31	0.337	0.188	31	0.277	0.967
pens	31	-0.002	0.343	31	0.188	0.202	31	0.245	0.588
plea	31	-1.083	0.400	31	-0.793	0.470	31	0.460	1.175

Note : napl = Naples,FL, pens = Pensacola, FL, and plea = Galveston Pleasure Pier, TX.

Table 5.2 Water Level Event Analysis :
January 18 through 22, 2003

Note : all dimensioned quantities are in feet.

start time = 17.7500 stop time = 19.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	-0.156	0.028	31	-0.814	0.235	31	0.697	8.532
pens	31	-0.279	0.132	31	-0.519	0.078	31	0.288	0.589
plea	31	-0.156	0.386	31	-0.623	0.157	31	0.529	0.407

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	-0.156	0.028	31	-0.338	0.192	31	0.261	6.985
pens	31	-0.279	0.132	31	-0.458	0.061	31	0.245	0.464
plea	31	-0.156	0.386	31	-0.783	0.144	31	0.689	0.374

start time = 18.7500 stop time = 20.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	-0.231	0.052	31	-0.125	0.120	31	0.152	2.295
pens	31	-0.332	0.106	31	-0.243	0.110	31	0.104	1.035
plea	31	-0.246	0.143	31	0.040	0.066	31	0.303	0.463

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	-0.231	0.052	31	-0.151	0.109	31	0.120	2.085
pens	31	-0.332	0.106	31	-0.014	0.062	31	0.348	0.580
plea	31	-0.246	0.143	31	-0.269	0.160	31	0.144	1.112

start time = 20.7500 stop time = 22.0000

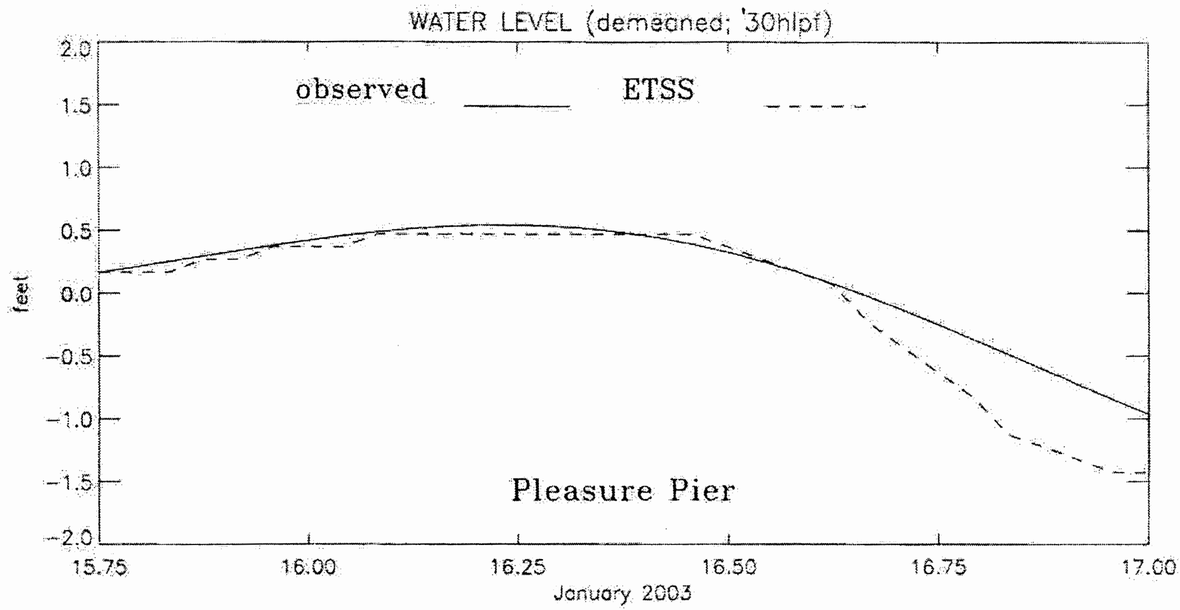
TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	-0.093	0.136	31	-0.006	0.100	31	0.134	0.733
pens	31	-0.037	0.173	31	-0.055	0.117	31	0.075	0.680
plea	31	-0.118	0.081	31	-0.318	0.112	31	0.272	1.372

DYNALYSIS vs. OBSERVED (adjusted)

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	-0.093	0.136	31	-0.079	0.102	31	0.115	0.752
pens	31	-0.037	0.173	31	-0.167	0.044	31	0.195	0.254
plea	31	-0.118	0.081	31	-0.308	0.134	31	0.269	1.642

Note : napl = Naples, Fl, pens = Pensacola, Fl, and plea = Pleasure Pier, TX.



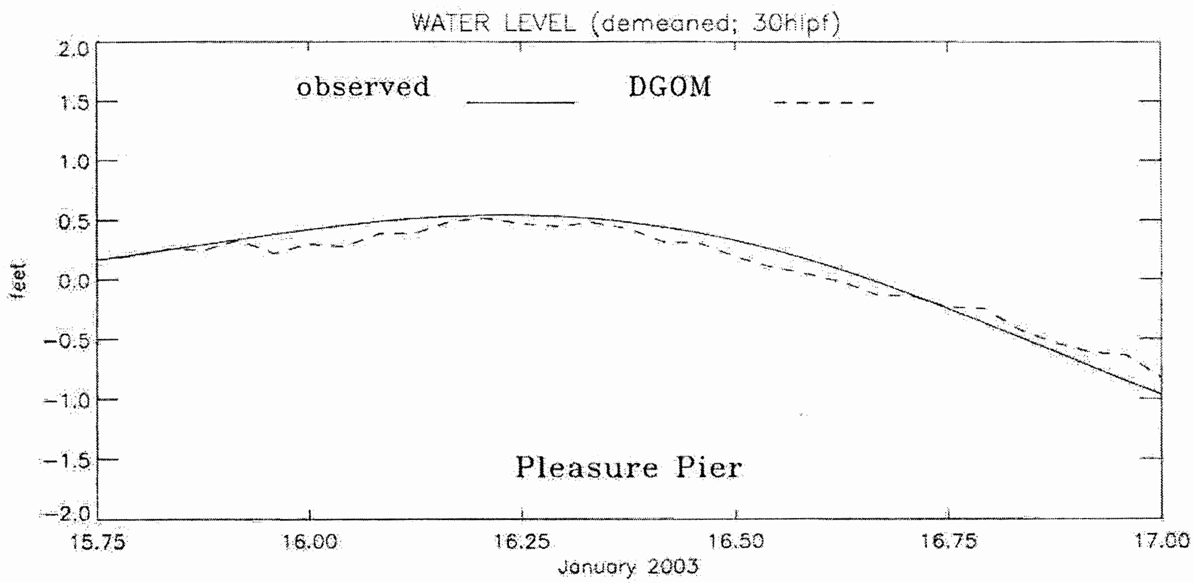
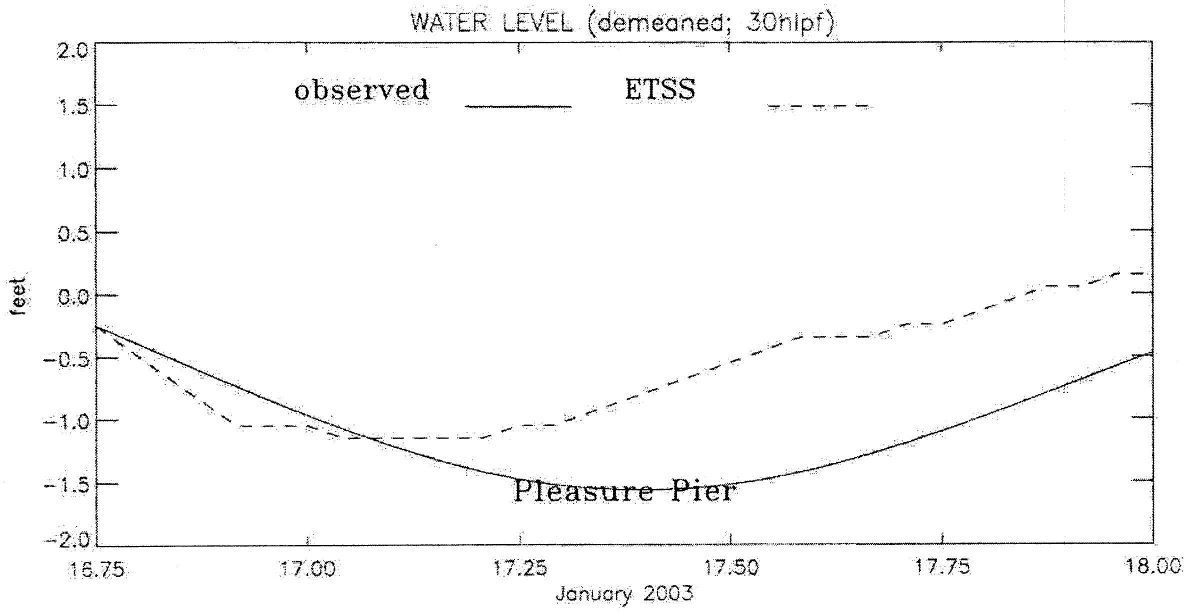


Figure 5.2 Simulated vs. Observed Water Levels from January 2003, Day 1, at Galveston Pleasure Pier, TX



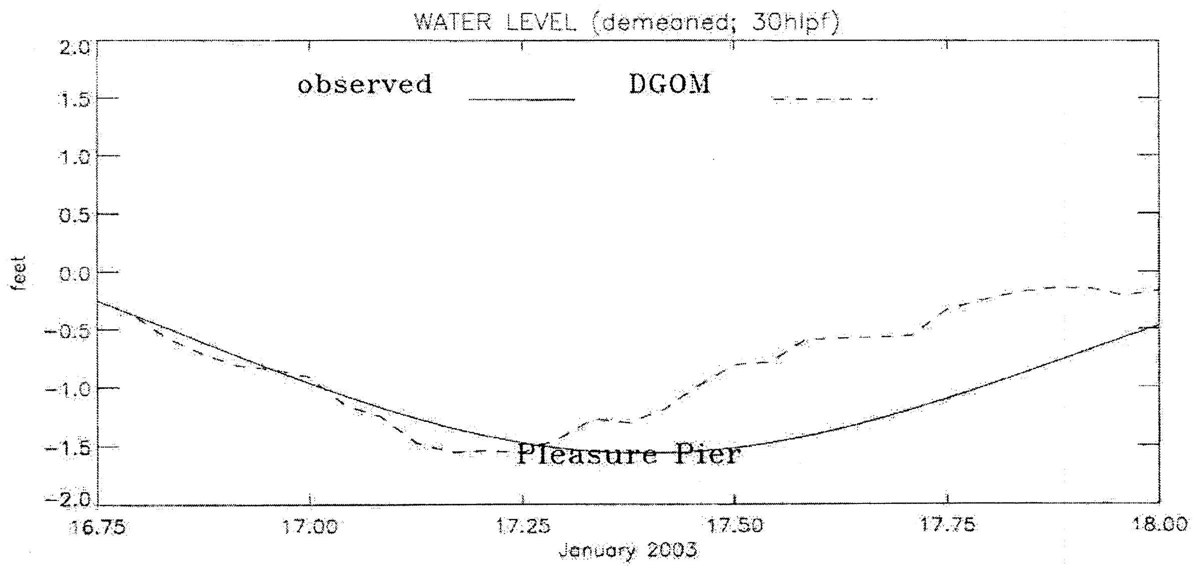
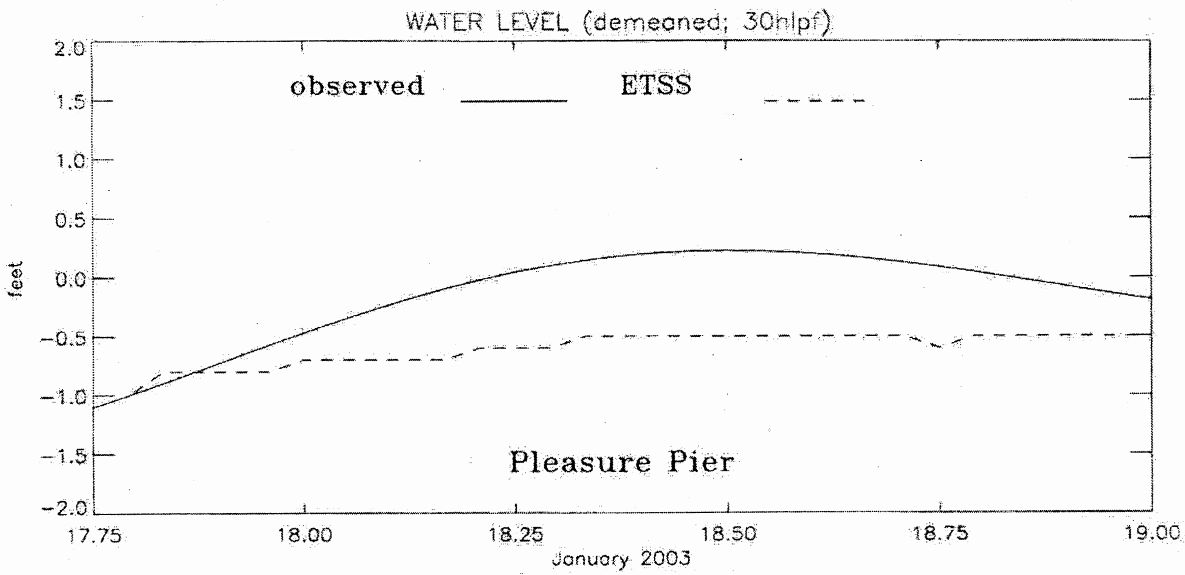


Figure 5.3 Simulated vs. Observed Water Levels from January 2003, Day 2, at Galveston Pleasure Pier, TX



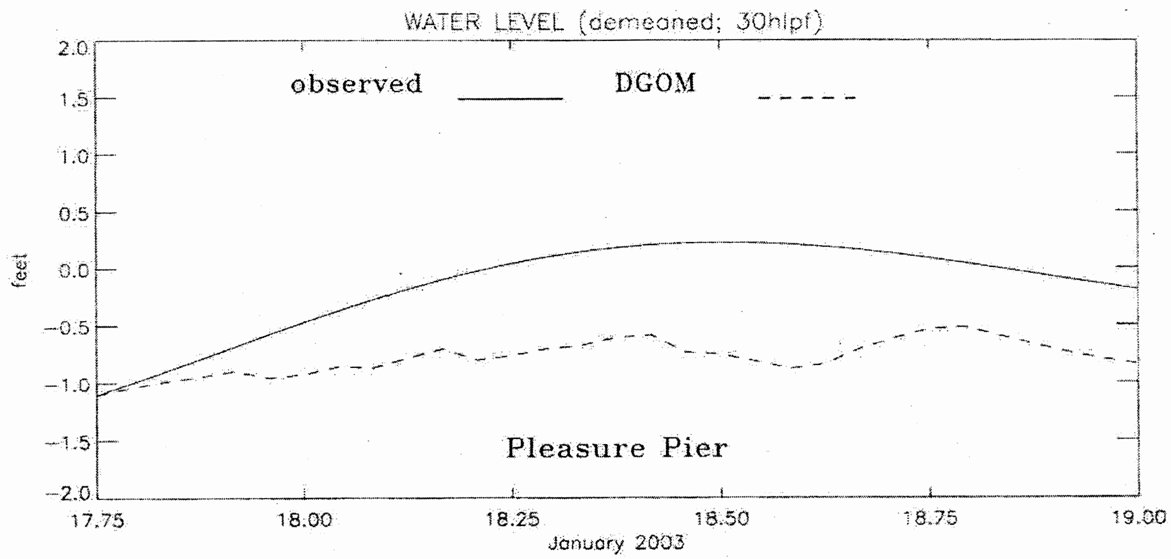


Figure 5.4 Simulated vs. Observed Water Levels from January 2003, Day 3, at Galveston Pleasure Pier, TX

Table 5.3 Water Level Analysis Summary for January 2003

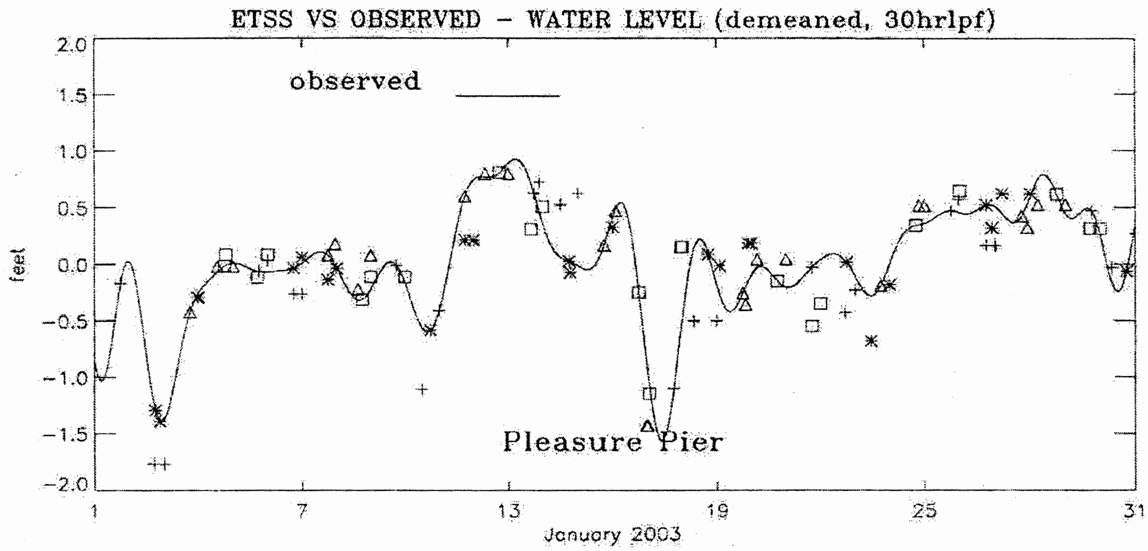
ETSS vs. DGOM Model Water Level Comparison

	ETSS		DGOM	
	rmse (ft)	npf (-)	rmse (ft)	npf (-)
Naples	0.2633	16	0.2692	9
Pensacola	0.1886	12	0.1819	13
Pleasure Pier	0.2707	16	0.2861	9

Note : rmse is root mean square error and npf is defined to be the number of preferred forecasts.

The water level analysis summary table (Table 5.3) again indicates that the performance of the DGOM forecast model is almost as good as the ETSS model. At all three stations, the rms error for the DGOM model forecasts is not significantly larger than for the ETSS model forecasts. The largest difference in rms error occurs at Pleasure Pier. But the difference here is less than 2/100 of a foot.

Examining the Forecast vs. Observed plots for Pleasure Pier (Figure 5.5) gives more evidence of comparability. Looking at the ETSS vs. Observed plot, we see that the ETSS model has over-predicted the first two low water events. Typically, the ETSS model over-predicts events. On this occasion however, the DGOM model has over-predicted the same two events by almost the same amount. When we look at the DGOM vs. Observed plot for Naples in Figure 5.6, we see a general scatter of forecast points about the observed curve. Looking at the ETSS vs. Observed plot, there is an extreme outlier which occurs just prior to the 19th. Both forecast models over-predict the low water event which occurs just prior to 25th, though the ETSS model over-predicts the event by a greater amount. At Pensacola (Figure 5.7), forecast points from both models seem to conform reasonably well to the observed curve.



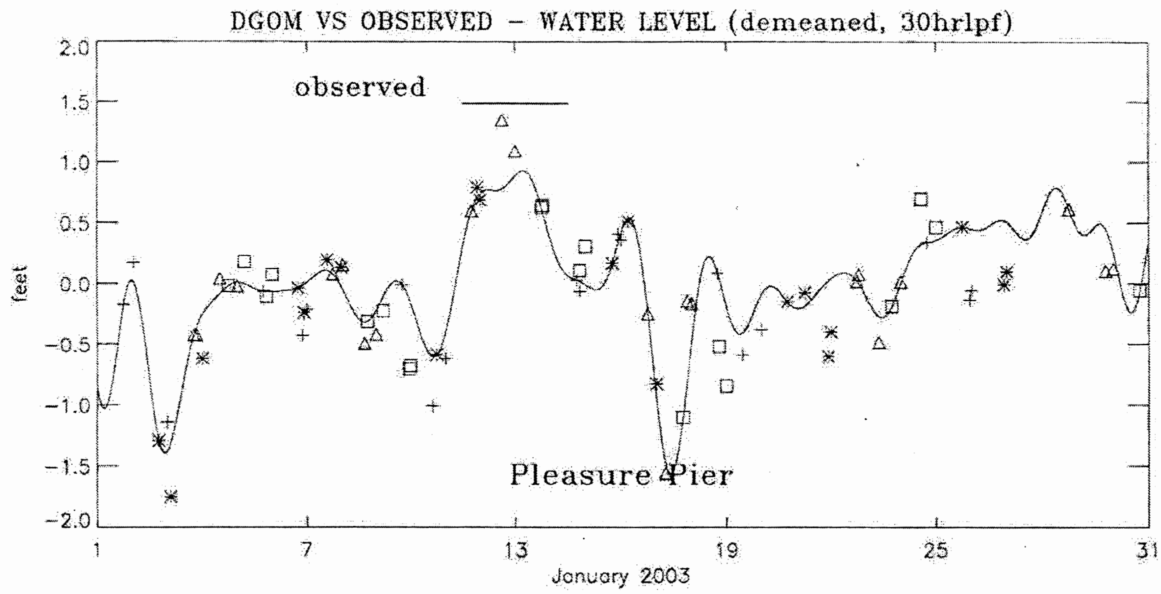
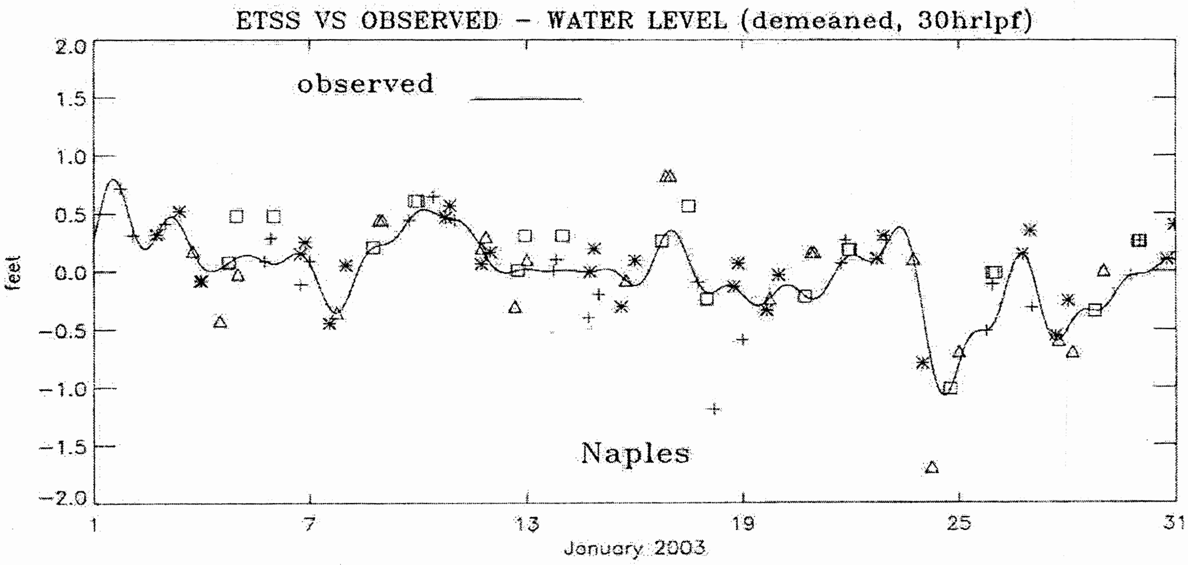


Figure 5.5 Forecast vs. Observed Water Levels at Galveston Pleasure Pier, TX, January 2003



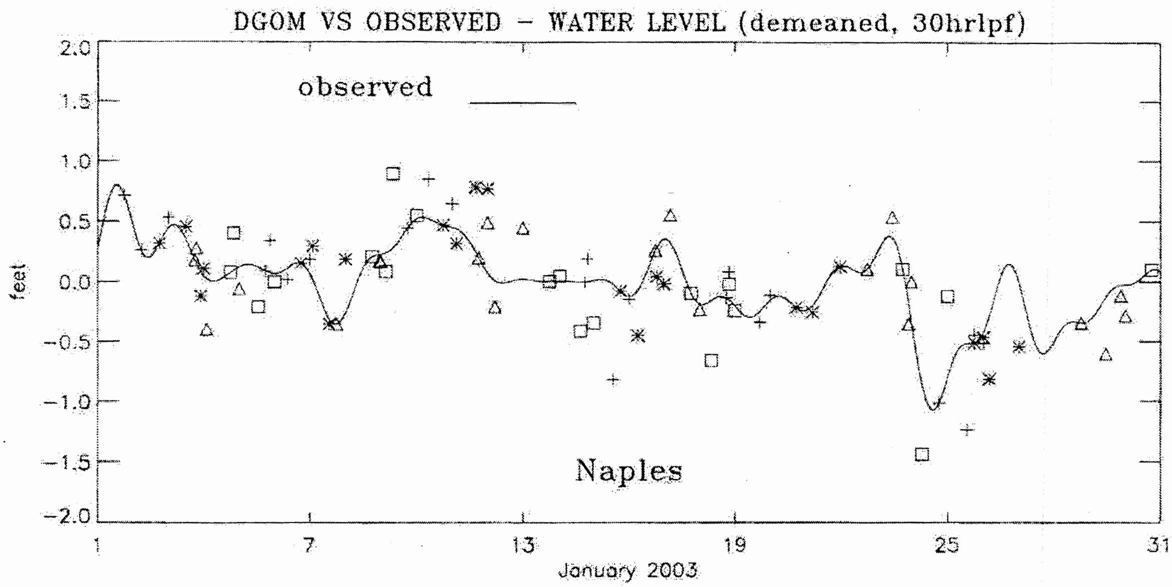
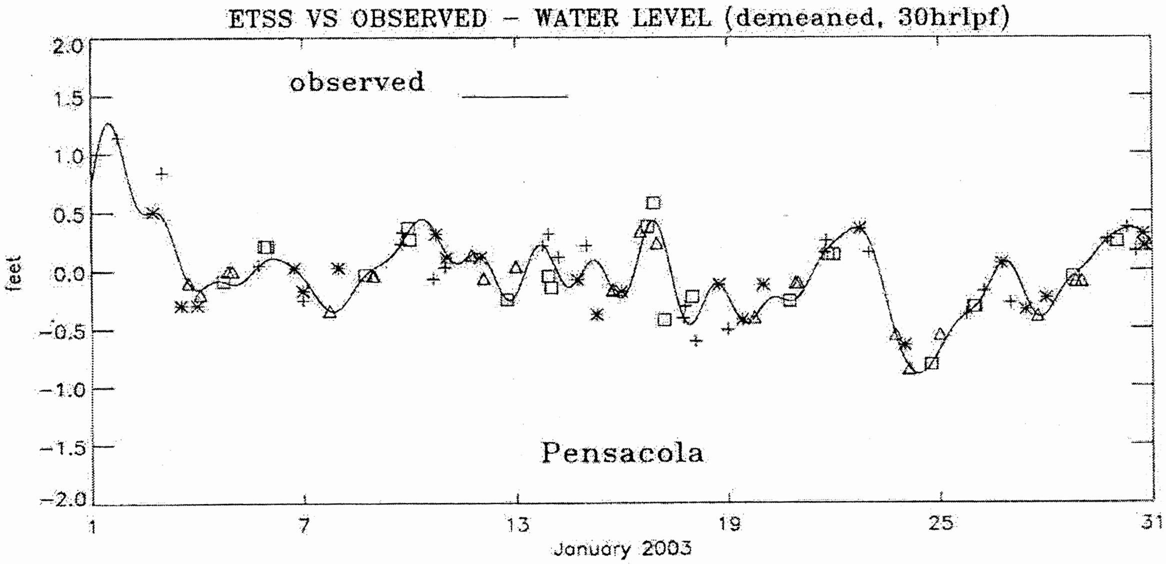


Figure 5.6 Forecast vs. Observed Water Levels at Naples, FL, January 2003



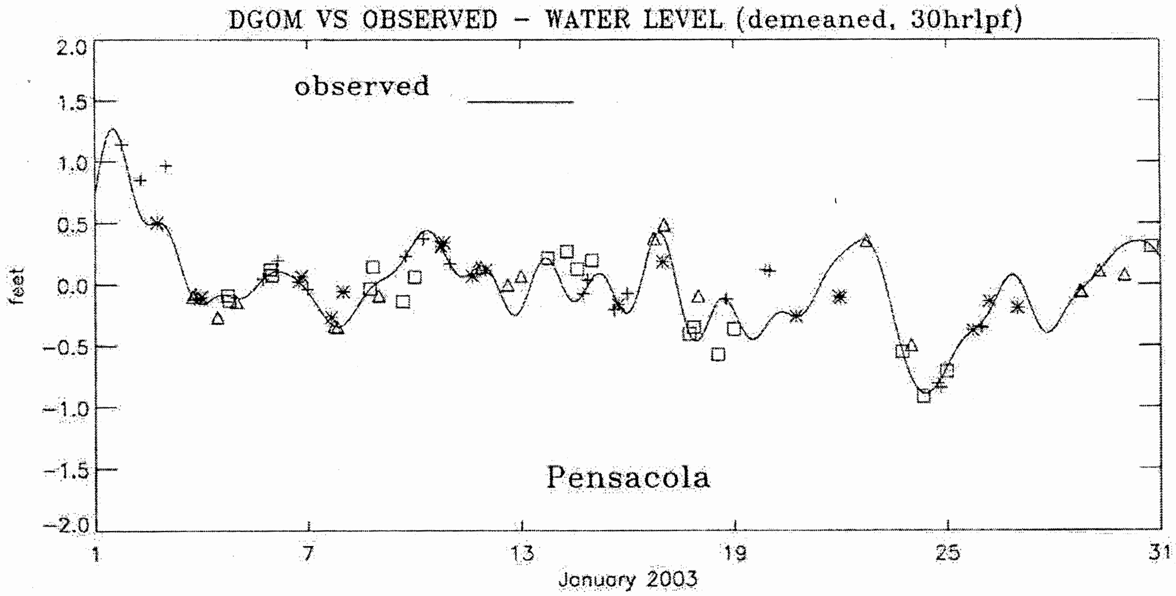


Figure 5.7 Forecast vs. Observed Water Levels at Pensacola, FL, January 2003

6. MAY 2003

The observed subtidal water levels at Galveston Pleasure Pier for May 2003 are somewhat choppy, but there are no significant events. The observed water level plot is presented below in Figure 6.1. The water level analysis summary is presented in Table 6.1.

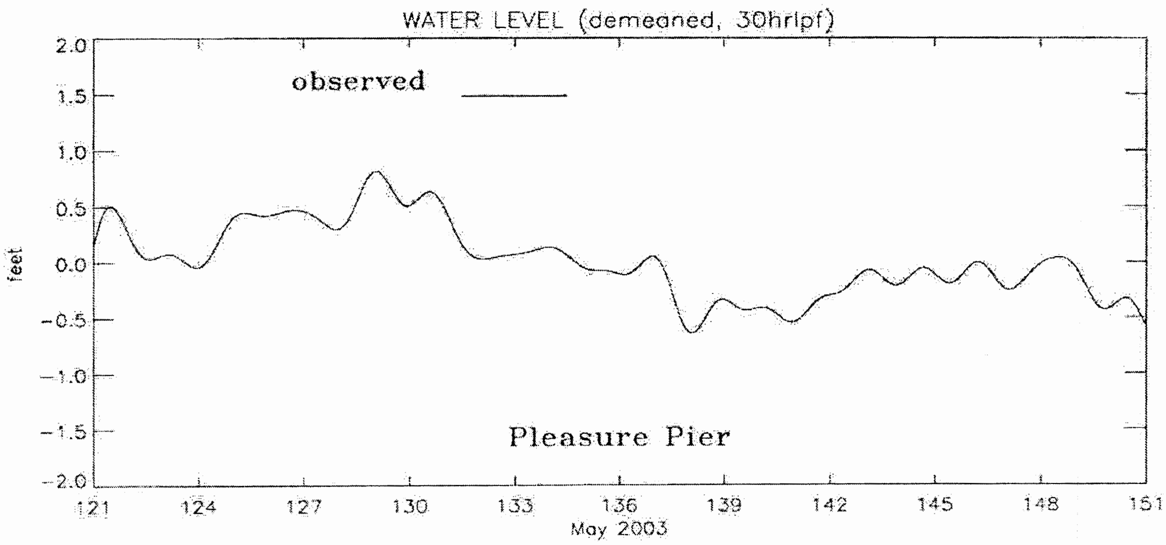


Figure 6.1 Observed Subtidal Water Level at Galveston Pleasure Pier, TX, May 2003

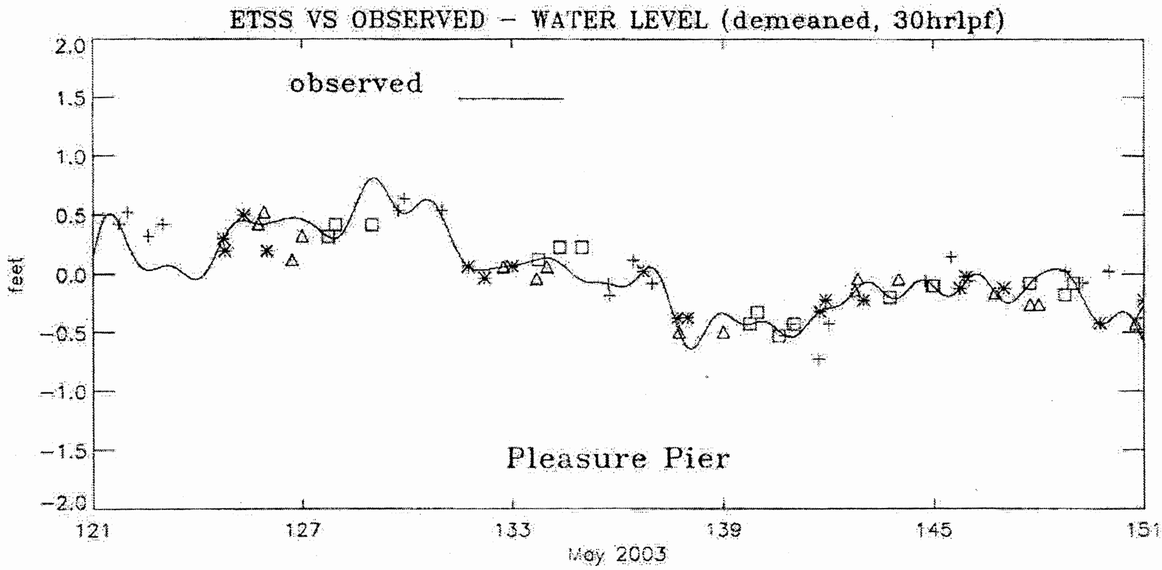
Table 6.1 Water Level Analysis Summary for May 2003

ETSS vs. DGOM Model Comparison, May 2003

	ETSS rms (ft)	npf (-)	DGOM rms (ft)	npf (-)
Naples	0.1207	17	0.1712	7
Pensacola	0.1214	18	0.1526	6
Pleasure Pier	0.1509	20	0.2087	4

Note : rmse is root mean square error and npf is defined to be the number of preferred forecasts.

Table 6.1 indicates that the ETSS model performs better than the DGOM model at each station, but only by a small amount (less than 6 one hundreds of a foot). Viewing the Forecast vs. Observed plots, Figures 6.2 through 6.4, indicates that the ETSS model forecast points hug the observed curve pretty closely. The DGOM model forecast points appear to be scattered just a bit more widely about the curve.



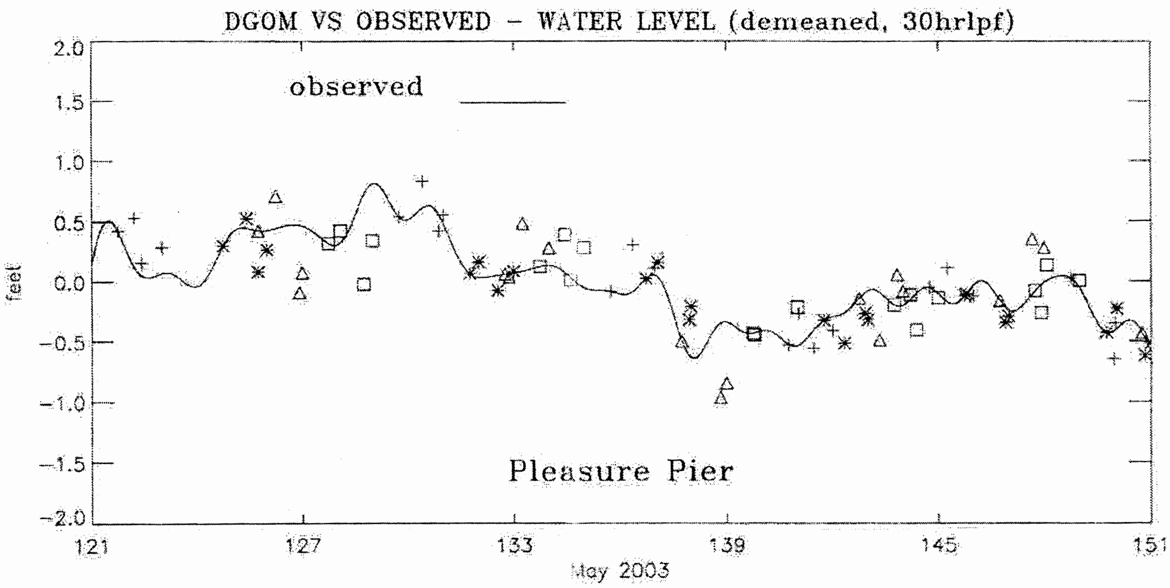
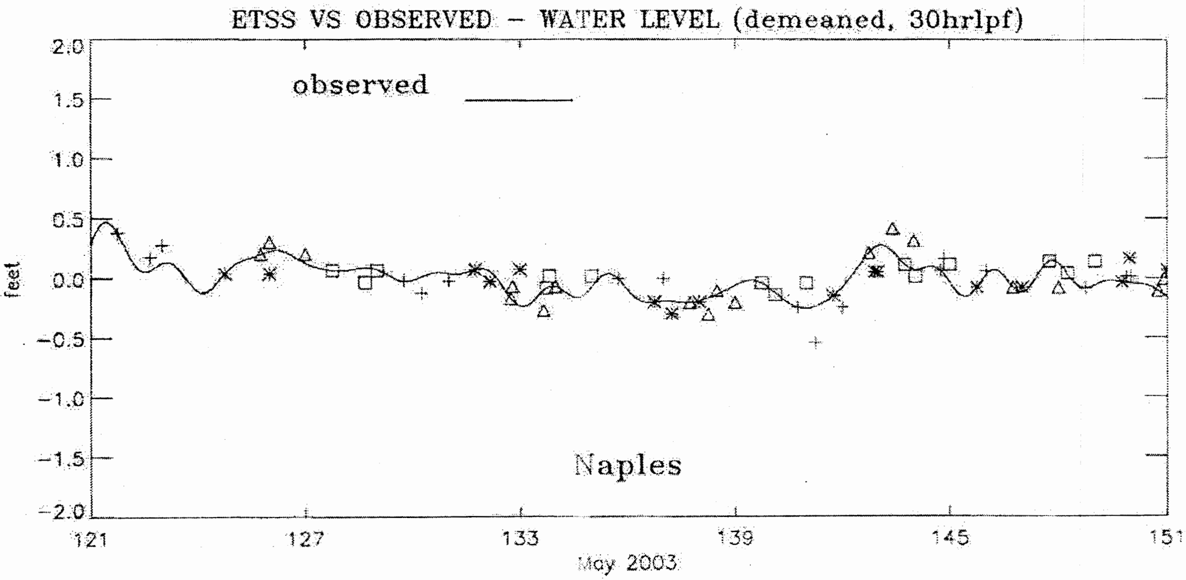


Figure 6.2 Forecast vs. Observed Water Levels at Galveston Pleasure Pier, TX, May 2003



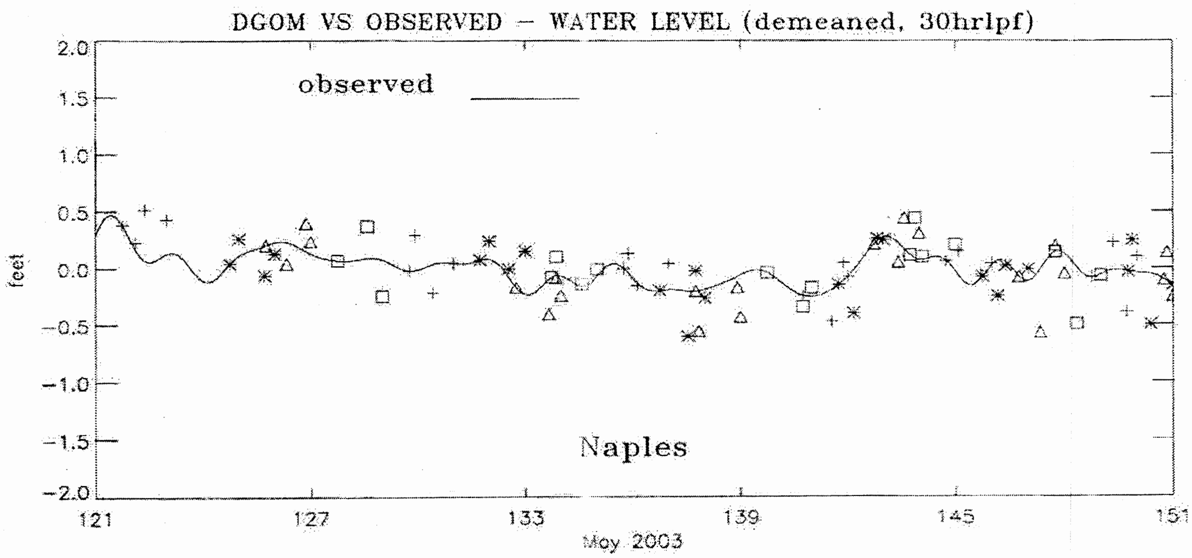
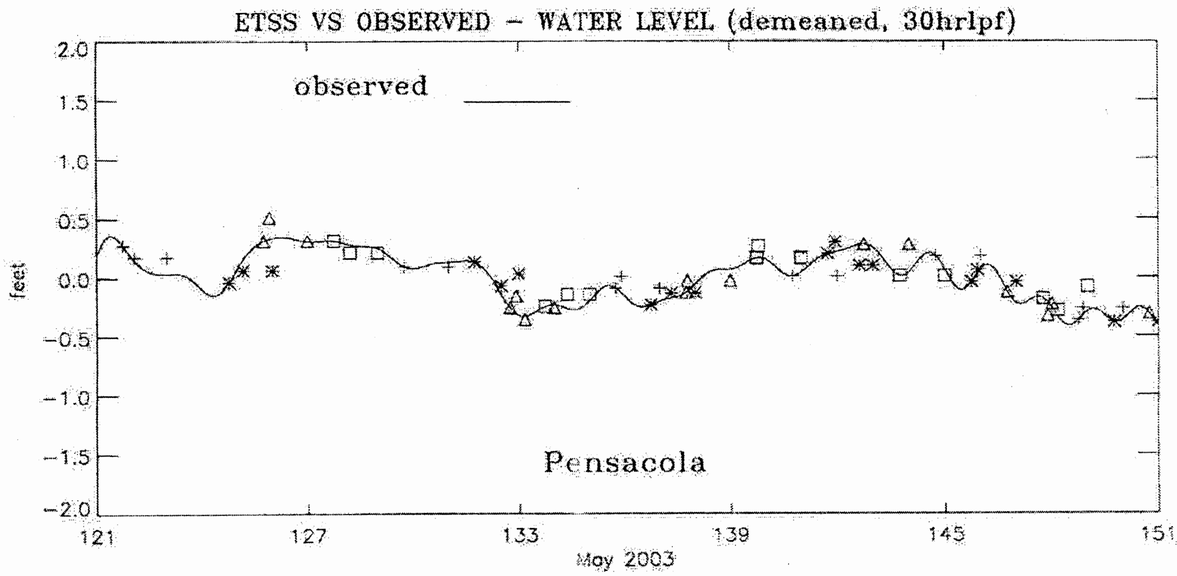


Figure 6.3 Forecast vs. Observed Water Levels at Naples, FL, May 2003



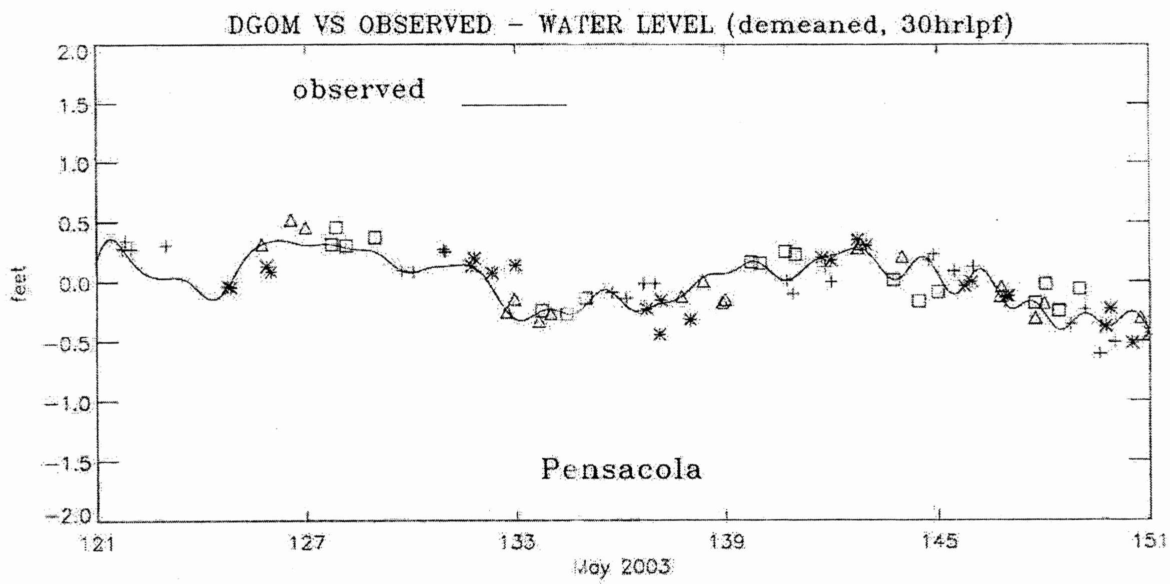


Figure 6.4 Forecast vs. Observed Water Levels at Pensacola, FL, May 2003

7. JULY 2003

Hurricane Claudette hit the Texas coast during mid July of 2003. As can be seen in Figure 7.1, the subtidal water level at Pleasure Pier reached nearly four feet around the fourteenth and fifteenth of July. Figures 7.2 through 7.5 depict this time period in greater detail.

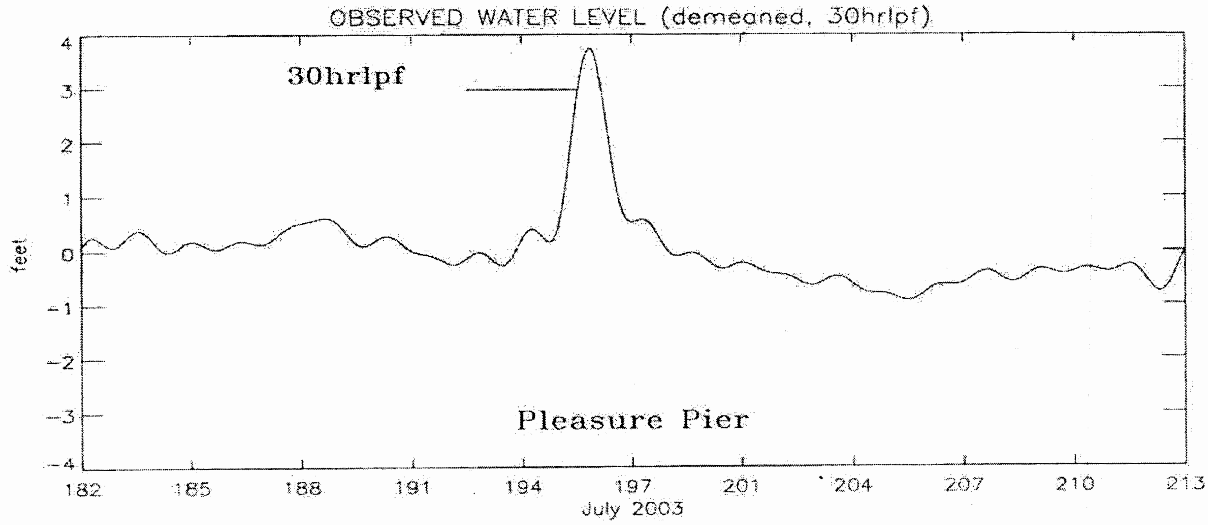


Figure 7.1 Observed Subtidal Water Level at Galveston Pleasure Pier, TX, July 2003

The daily statistical table for Julian days 193.75 through 198.0 is presented in Table 7.1. The peak observed value is reached on the day of 195.75 through 197.0. At Pleasure Pier, the rms error of the ETSS model forecast is 0.804 feet. The rms error of the DGOM model forecast is 1.994 feet.

The pre-hurricane conditions can be seen in Figure 7.2. Figure 7.3 shows the storm "ramping up". The ETSS model misses the initial portion of the high-water event. Unfortunately, there was no DGOM model forecast for July 13. The peak of the high-water event is depicted in Figure 7.4. The ETSS model under predicts the event by about one to one and a half feet. The DGOM model over predicts the event by almost 4 feet. Figure 7.5 shows the period of time immediately following the hurricane. Overall, the ETSS response is somewhat flat compared with the observed signal. This could be due to the lack of resolution in the Aviation forecast winds, used by the ETSS model. It is not known why the DGOM model over predicted the event so dramatically. It is also not known how the DGOM model was re-started following the missing forecast on the 13th.

Table 7.1 Water Level Event Analysis :
July 12 through 17, 2003

Note : all dimensioned quantities are in feet.

start time = 193.7500 stop time = 195.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.019	0.050	31	-0.115	0.062	31	0.160	1.221
pens	31	0.009	0.076	31	-0.057	0.050	31	0.115	0.653
plea	31	0.111	0.247	31	-0.182	0.093	31	0.354	0.376

DYNALYSIS (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.019	0.050	31	-0.002	0.119	31	0.103	2.356
pens	31	0.009	0.076	31	-0.094	0.034	31	0.132	0.444
plea	31	0.111	0.247	31	0.130	0.178	31	0.166	0.721

start time = 195.7500 stop time = 197.0000

TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.051	0.019	31	0.028	0.060	31	0.061	3.110
pens	31	0.127	0.019	31	0.118	0.030	31	0.033	1.533
plea	31	2.837	0.744	31	2.156	0.370	31	0.804	0.497

DYNALYSIS (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.051	0.019	31	0.003	0.119	31	0.124	6.210
pens	31	0.127	0.019	31	0.148	0.042	31	0.046	2.172
plea	31	2.837	0.744	31	4.044	1.752	31	1.994	2.355

start time = 196.7500 stop time = 198.0000

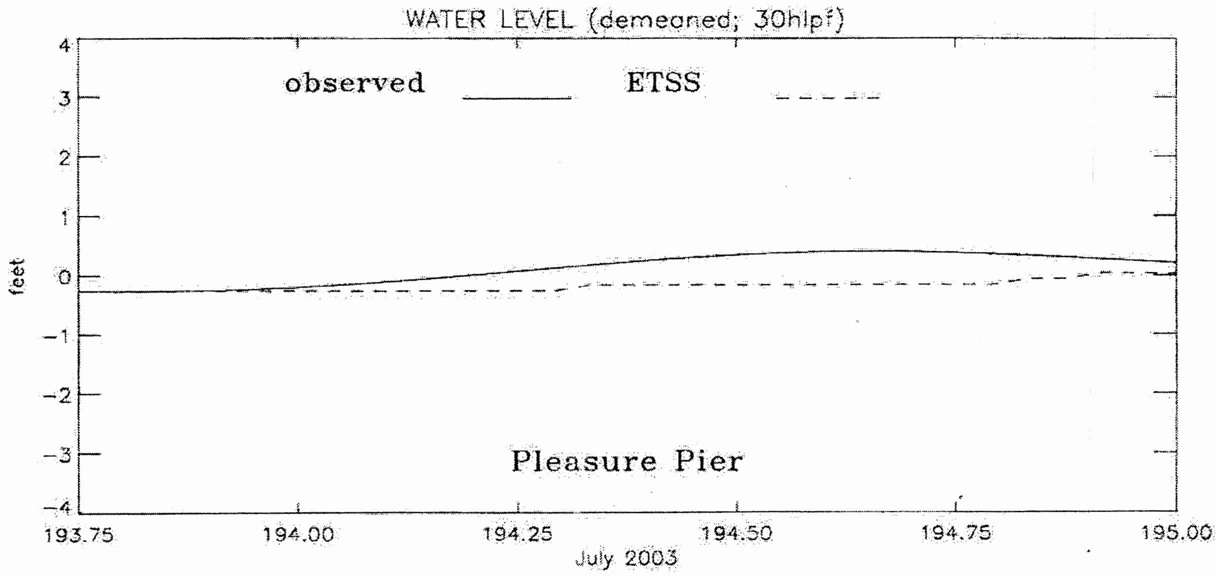
TDL (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.058	0.039	31	0.022	0.049	31	0.074	1.268
pens	31	0.035	0.056	31	0.036	0.058	31	0.045	1.038
plea	31	0.901	0.550	31	0.886	0.544	31	0.156	0.989

DYNALYSIS (adjusted) vs. OBSERVED

stat	num obs	mean	sd	num mod	mean	sd	num	rms diff	variability
napl	31	0.058	0.039	31	0.140	0.124	31	0.165	3.198
pens	31	0.035	0.056	31	0.105	0.055	31	0.100	0.984
plea	31	0.901	0.550	31	1.342	0.396	31	0.530	0.720

Note : napl = Naples, FL, pens = Pensacola, FL, and plea = Galveston Pleasure Pier, TX.



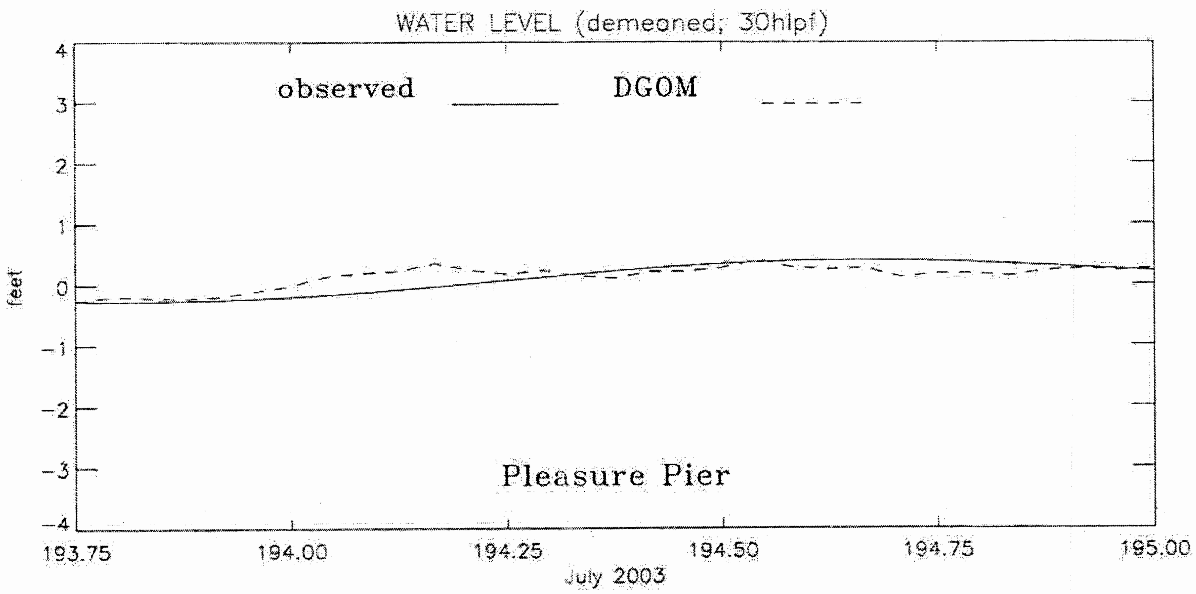
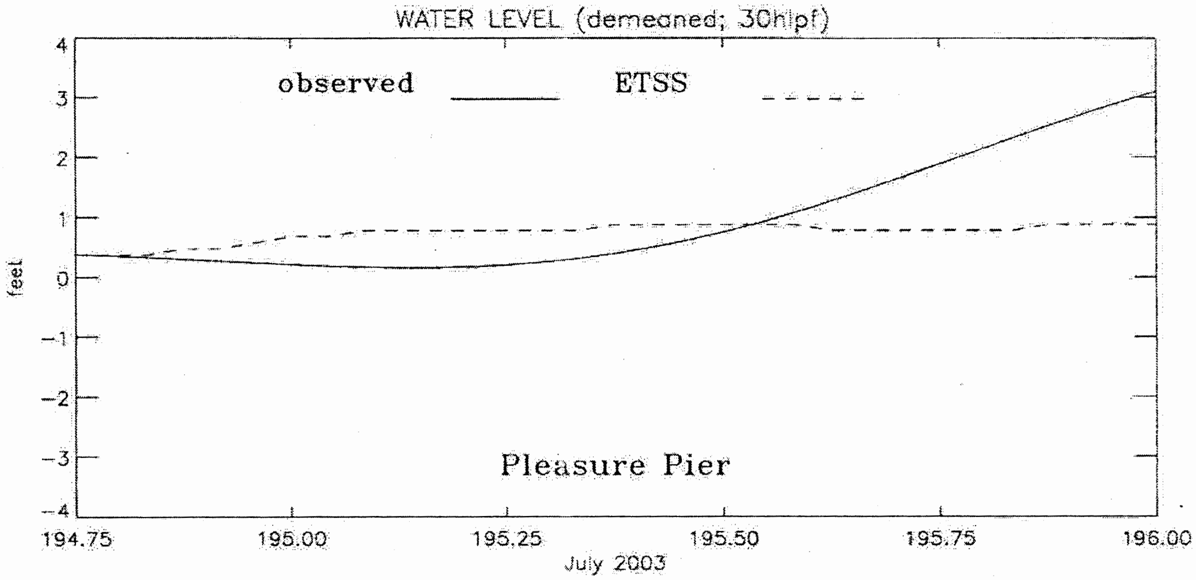
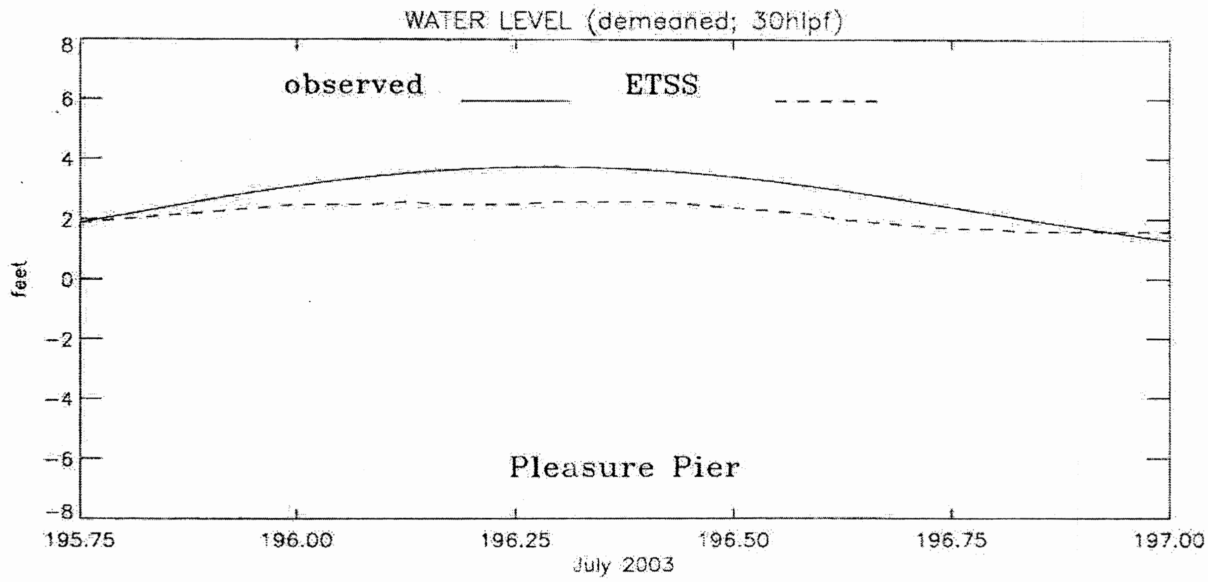


Figure 7.2 Simulated vs. Observed Water Levels from July 2003, Day 1, at Galveston Pleasure Pier, TX



No Dynalysis Forecast for July 13, 2003

Figure 7.3 Simulated vs. Observed Water Levels from July 2003, Day 2, at Galveston Pleasure Pier, TX



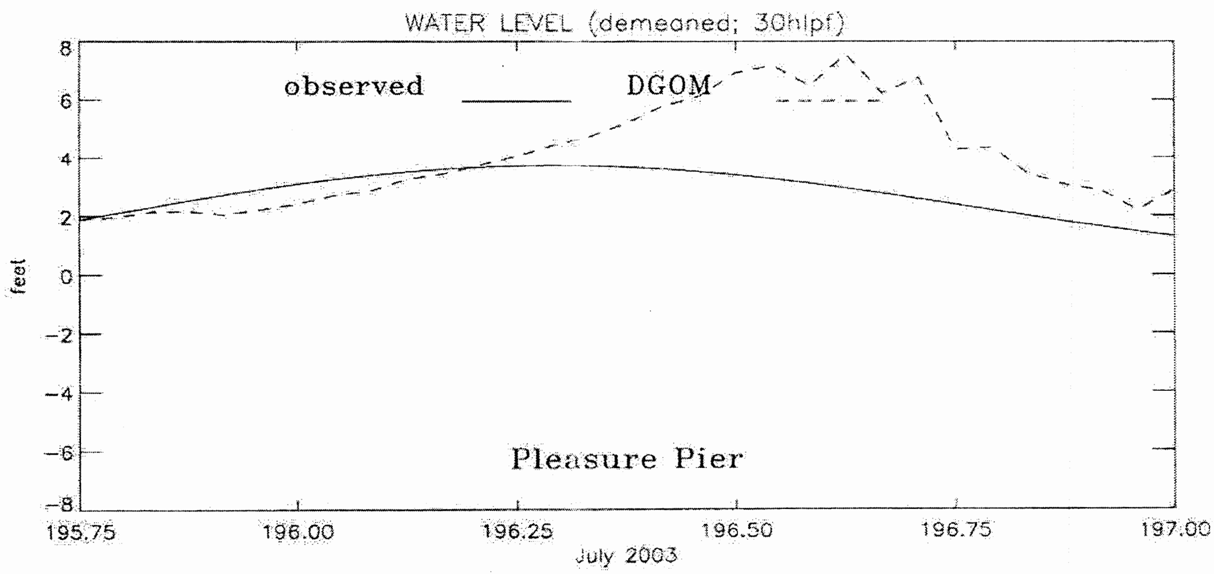
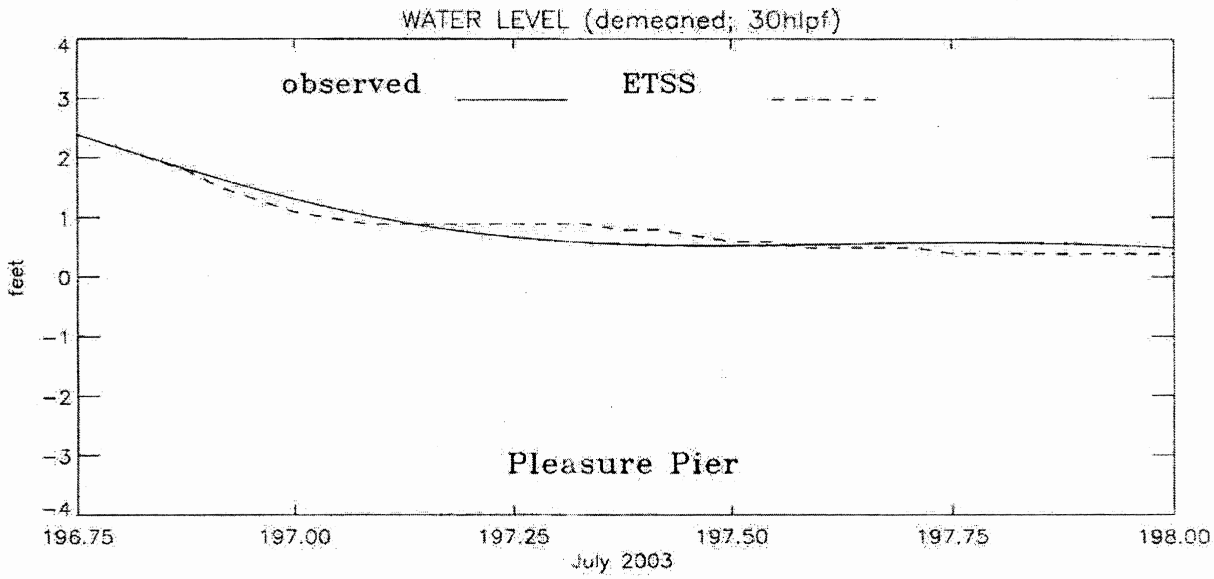


Figure 7.4 Simulated vs. Observed Water Levels from July 2003, Day 3, at Galveston Pleasure Pier, TX



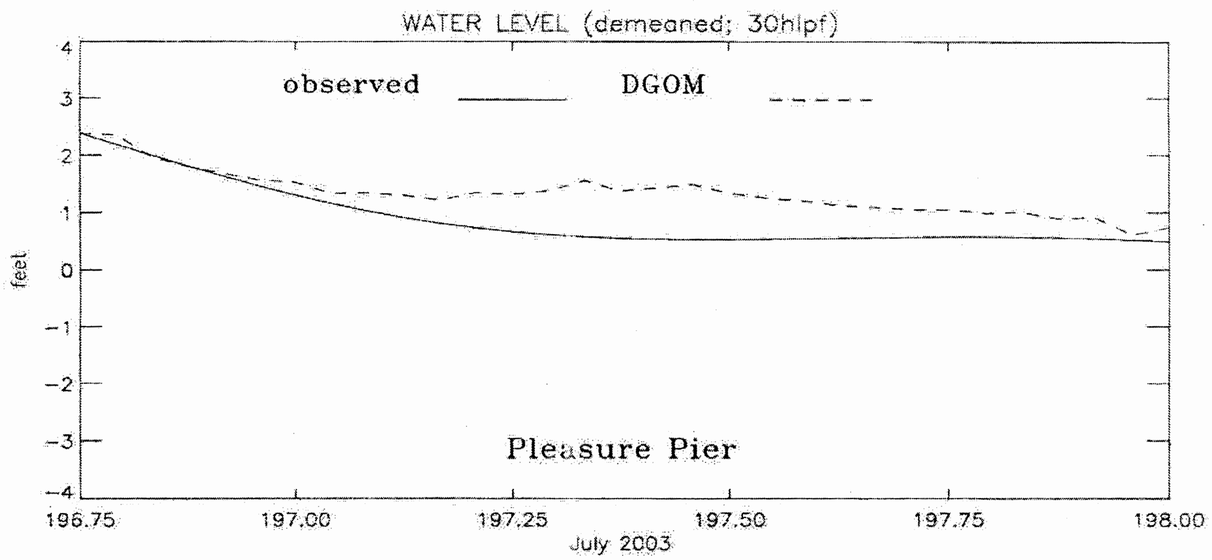


Figure 7.5 Simulated vs. Observed Water Levels from July 2003, Day 4, at Galveston Pleasure Pier, TX

Table 7.2 Water Level Analysis Summary for July 2003

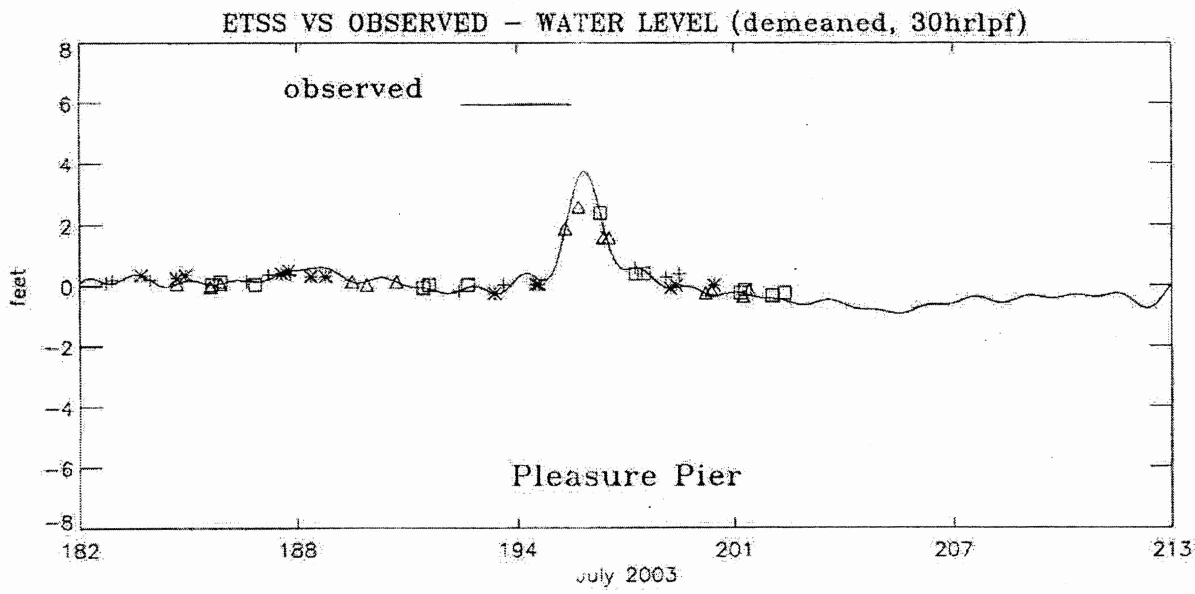
ETSS vs. DGOM Model Comparison, July 2003

	ETSS rms (ft)	npf (-)	DGOM rms (ft)	npf (-)
Naples	0.1028	14	0.1892	2
Pensacola	0.1699	9	0.1500	7
Pleasure Pier	0.2672	10	0.5427	6

Note : rmse is root mean square error and npf is defined to be the number of preferred forecasts.

As noted in Table 7.2, the rms error of the ETSS model forecasts was about half that of the DGOM model forecasts at Pleasure Pier over the first twenty days of July. DGOM model forecasts were available for the first twenty days of July only. Figure 7.6 shows the forecast vs. observed comparison for July at Pleasure Pier. The ETSS model under predicting the high water event can best be seen by looking at the triangles during the event. The DGOM model value peaked at nearly eight feet.

From Table 7.2, the ETSS model appears to out perform the DGOM model at Naples. The rms error for the ETSS forecasts is more than 8/100 less than the rms error for the DGOM forecasts. The ETSS model forecast is preferred 14 times, compared with only two for the DGOM model forecast. A look at the plots in Figure 7.7 would seem to confirm this. Figure 7.8 and Table 7.2 indicate that the DGOM model seems to perform better at Pensacola. The DGOM model has the lower rms error, but is the preferred forecast only seven times, compared with nine for the ETSS model.



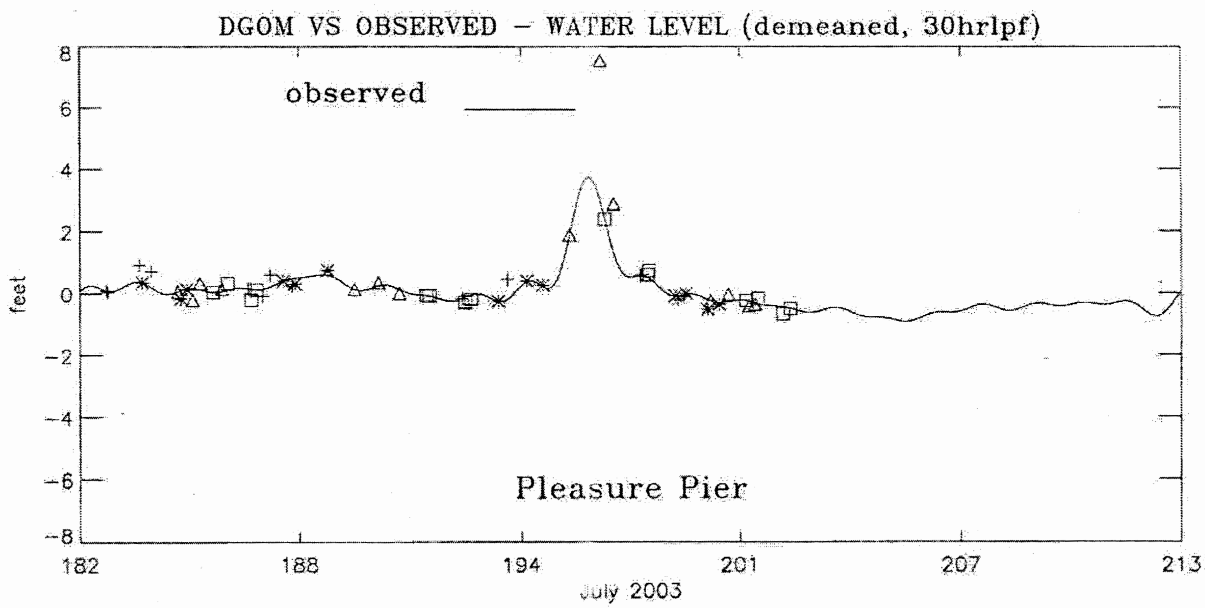
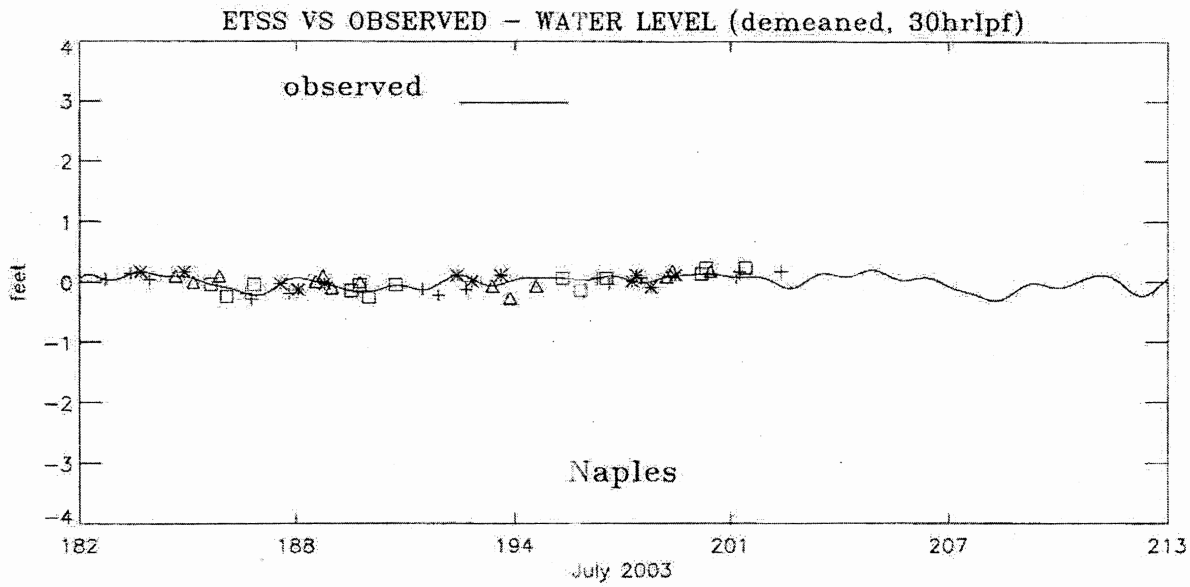


Figure 7.6 Forecast vs. Observed Water Levels at Galveston Pleasure Pier, TX, July 2003



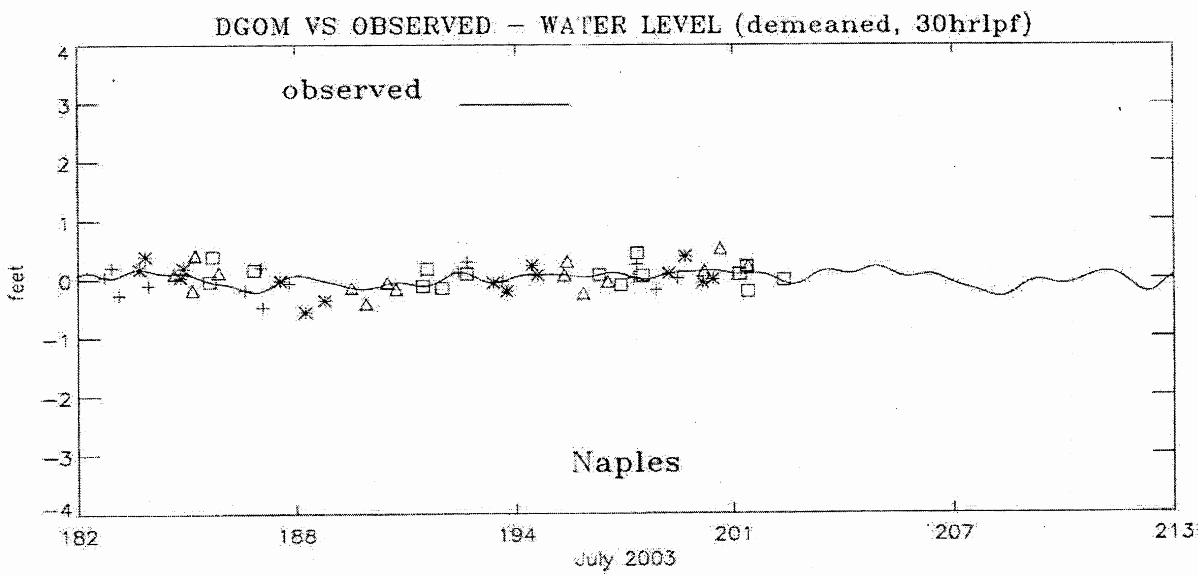
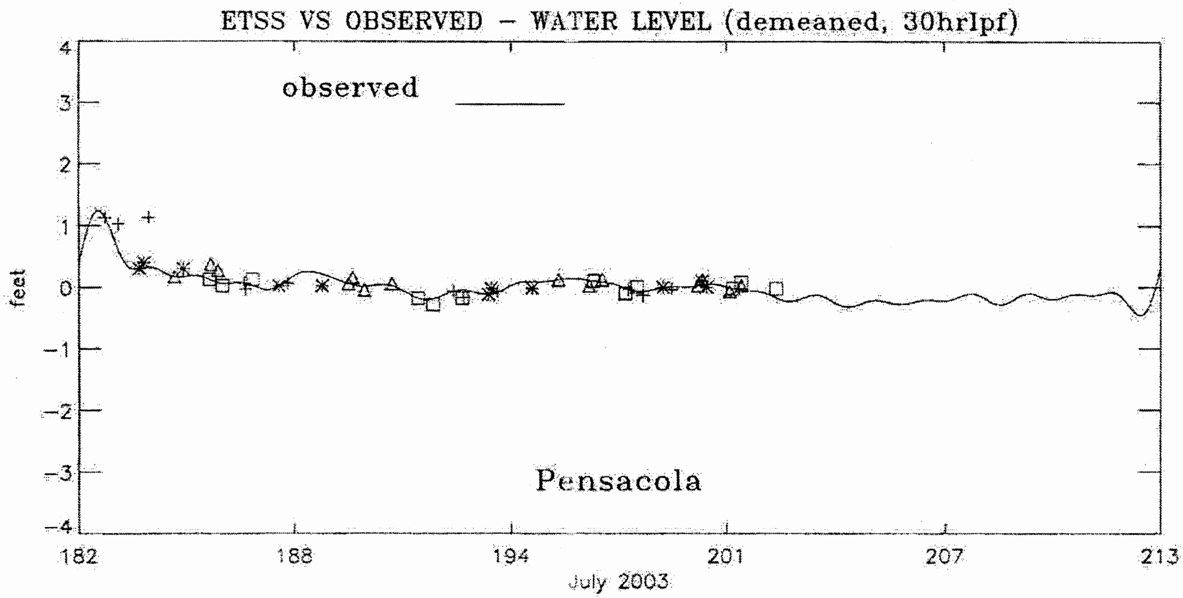


Figure 7.7 Forecast vs. Observed Water Levels at Naples, FL, July 2003



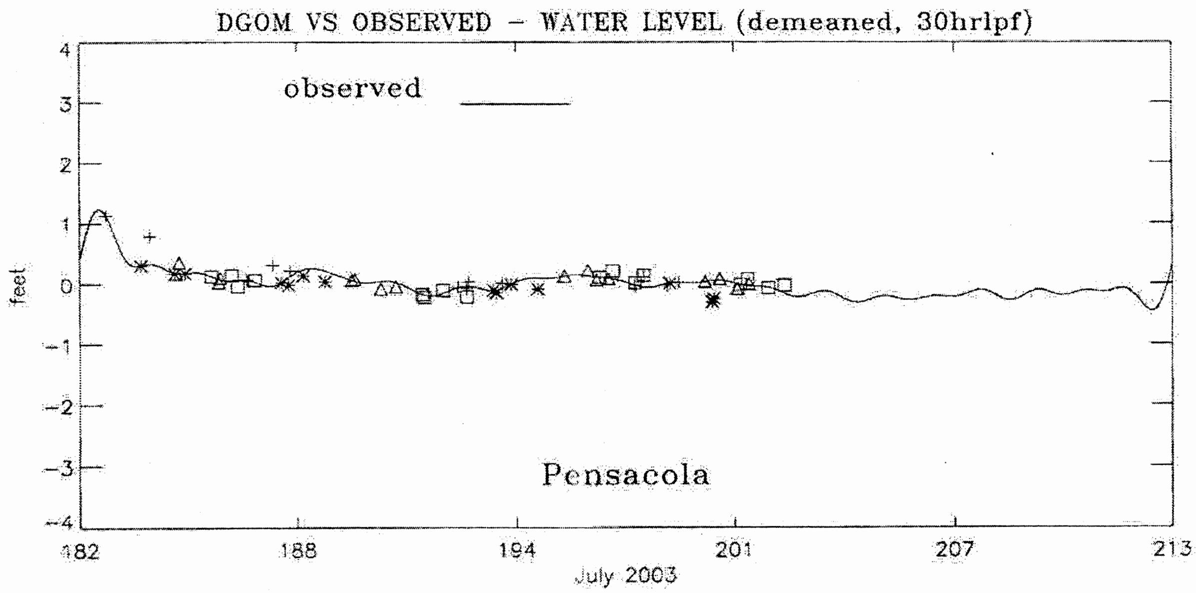


Figure 7.8 Forecast vs. Observed Water Levels at Pensacola, FL, July 2003

CONCLUSIONS

Subtidal water level comparisons of observations with forecast DGOM and ETSS model output were performed for the months of November 2002, January 2003, May 2003, and July 2003. Subtidal water levels were obtained by 30 hour low pass filtering. In the energetic fall and winter months, both DGOM and ETSS model water levels compared favorably with the observations and were of near equal quality in terms of both RMS error and in the number of preferred forecasts at all stations around the Gulf. This was also the case at Galveston Pleasure Pier, TX, for a major cold frontal passage low water event in January 2003. For November 2002 a separate analysis was made using detiding. The results from detiding were slightly degraded for both models with each model response of comparable quality. Thus no further detiding analyses were performed. In the quiescent spring and summer months, the DGOM model water levels exhibited a greater spread around the observed signal than the ETSS model water levels. During Hurricane Claudette in July 2003, the DGOM model over-predicted the peak water level at Galveston Pleasure Pier, TX by about 4 feet, while the ETSS model under predicted the observed 4 foot surge by about 1.5 feet.

The cause of the differences in model water levels is under further investigation. The use of different wind and sea level atmospheric forcing may be a factor. The DGOM model uses the USN COAMPS, while the ETSS model uses the NWS AVN forcings. Efforts are underway to run the DGOM model with COAMPS, AVN, and ETA forcings for a common time period and use the analysis procedures developed here to compare the water level responses.

Future enhancements to the analysis procedure might include the following :

1. For the daily statistics, include the peak water level value. Presently, the table includes the mean water level over the forecast period. Including the peak water level value would give more meaningful information.
2. On the monthly summary tables, npf is defined to be the number of preferred forecasts. If the ETSS model has an rms error of 0.2146 and the DGOM model has an rms error of 0.2171, then ETSS is the preferred forecast. In this instance however, the difference in rms error is negligible. An improvement would be to introduce a threshold value. A difference in rms error less than a specified threshold value, perhaps 0.05 feet, would be considered a tie.
3. Improve the monthly forecast vs. observed plots by plotting the forecast water level points uniformly by day. Presently, the plot program, plot_wlanal.pro, cycles through the four symbols for each model regardless of forecast date. If one model misses a forecast, it becomes difficult to compare forecast points for a given day because the symbols no longer correspond.

Based on the results of this study, NOS is also pursuing the installation of the DGOM model system due to its ability to represent the baroclinic structures and the Loop Current system via data assimilation of SST and AVHRR. The system will be further tested on storm event water level response and on its ability to improve the NOS Galveston Bay Experimental Nowcast/Forecast System (GBEFS) water levels.

REFERENCES

Chen, J., W.A. Shafer, and S.C. Kim, 1993: A Forecast Model for Extratropical Storm Surge, **Advances in Hydro-Science and Engineering**, (ed.) Sam S.Y. Wang, Volume I Part B, University of Mississippi, 1437-1444.

Chen, J., 2003: Personal Communication.

Patchen, R., H.J. Herring, P.P. Niiler, M. Inoue, G.L. Mellor, C.N.K. Mooers, L.Y. Oey, F.M. Vulkovich, and W.J. Wiseman, Jr., 1999: Modeling the Gulf of Mexico. **Dynalysis of Princeton Report No. 115.**

Patchen, R., and J. Herring, 1998: GUMCurrent: Gulf of Mexico Tropical Cyclone Current Hindcast. **Dynalysis of Princeton Report No. 117.**

Patchen, R. and J.P. Blaha, 2002: Implementation of an Infrastructure to Support Operation and Evaluation of Gulf of Mexico Models, **Oceans 2002 MTS/IEEE**, Biloxi, Mississippi. (Preprint available from R. Patchen at www.dynalysis.com)

Patchen, R., W.J. Wiseman Jr., M. Inoue, V. Ransibrahmanakul, S.P. Murray, and S. DiMarco, 1998: Hydrography of the Louisiana Coastal Current: Model-Data Comparison. **Physics of Estuaries and Coastal Seas**, Balkema Publishers, The Netherlands.

Patchen, R., L.L. Lee, and F.J. Kelley, 1999: Preliminary Comparison of Near-surface TABS Current Velocity Observations and Dynalysis Model Simulations Along the Inner Texas Shelf. **AMS 3rd Conference on Coastal and Atmospheric and Oceanic Prediction Processes,** New Orleans, Louisiana. (Preprint available from R. Patchen at www.dynalysis.com)

Patchen, R., L.L. Lee, F.J. Kelley, N.L Guinasso, 2002: Observed and Modeled Near-surface Currents in the Coastal Jet Along the Texas Shelf. AGU 2000 Ocean Sciences Meeting Abstract, January 25-28, San Antonio, Texas.

Richardson, P.H. and R.A. Schmalz, 2002: Water Level Event Analysis: Program Documentation. NOAA, National Ocean Service, Coast Survey Development Laboratory, **CSDL Informal Technical Note No. 1**, Silver Spring, MD.

Zervas, C.E., 1997: User's Guide to the Tidal Prediction and Detiding Program. NOAA, National Ocean Service, Center for Operational Oceanographic Products and Services, Unpublished Internal Report, Silver Spring, MD.

APPENDIX A. SCRIPT AND CONTROL FILES

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

reform.sh

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

# rm *.o

  reformx < reform.n
```

reform.n

```
naple.jul03.raw
naple.jul03.obs
  60.0 time interval of raw data set
 no* - designate start and end time, yes or no
  182.0  start time
  212.96    end time
```

readtdl.sh

```
# f77 read_tdl.f calcjd.f -o readtdl

# rm *.o

  readtdl < read_jul03.n
```

read_nov02.n

```
 22    number of stations to read
 3    number of station output files
/ocean2dir1/adaser/ocean/fcsts/etss/archives/200211/gm/2002113012.gm
 1    station number
 8    lunout
naple.tdl.11302002_12z
10
 9
pensa.tdl.11302002_12z
19
10
pleas.tdl.11302002_12z
 4
11
porta.tdl.11302002_12z
```

readdyn.sh

```
# f77 read_dyn.f -o read_dyn

  read_dyn < readdyn.n > out
```

readdyn.n

```
0    idebug
/dir3/people/philr/dynalysis/dynal/GOM.NOS/2D/jul03/GOM_2D.2003.200.1300.NOS
4    number of Dynalysis stations to read
 1NAPLES
dyn.01naple.200
27PENSAC
dyn.27pensa.200
36GALVEP
dyn.36pleas.200
62PTARAN
dyn.62ptara.200
```

adjust.sh

```
# f77 adjust.f -o adjust


# adjust < adj_tdl.jan03.n
# adjust < adj_tdl.may03.n
  adjust < adj_tdl.jul03.n
# adjust < adj_tdlres.nov02.n
# ls -ll *.tdl.*

# adjust < adj_dynres.nov02.n
# adjust < adj_dyn.may03.n
# adjust < adj_dyn.jul03.n
```

adj_tdl.jan02.n

```
3          number of stations
 31.749    start time (daily)
 9
/dir3/people/philr/dynalysis/tdl/12z/jan03/naple.tdl.01312003_12z
10
/dir3/people/philr/dynalysis/obs/filter/jan03/naple.jan03_0.obs.30
11
naple.tdl.01312003_adj
12
/dir3/people/philr/dynalysis/tdl/12z/jan03/pensa.tdl.01312003_12z
13
/dir3/people/philr/dynalysis/obs/filter/jan03/pensa.jan03_0.obs.30
14
pensa.tdl.01312003_adj
15
/dir3/people/philr/dynalysis/tdl/12z/jan03/pleas.tdl.01312003_12z
16
/dir3/people/philr/dynalysis/obs/filter/jan03/pleas.jan03_0.obs.30
17
pleas.tdl.01312003_adj
```

wl_sa.sh

```
# f77 wl_sa.ph.f -o wl_sa.ph


# wl_sa.ph < wl.dyn.n

# wl_sa.ph < wl.tdl.n

# wl_sa.ph < wl.b2_nov02.n
# wl_sa.ph < wl.nov02_astr.n
# wl_sa.ph < wl.b2_jan03.n > out
# wl_sa.ph < wl.b2_jan03.astr.n > out
# wl_sa.ph < wl.may03.n > out
  wl_sa.ph < wl.jul03.n > out

# rm out
```

wl.jul03.n

```
1               idebug
3               # of stations PAIRS
Naples
Pensacola
Pleasure Pier
wl.out.jul03
TDL (adjusted) vs. OBSERVED
DYNALYSIS (adjusted) vs. OBSERVED
July 2003
16               number of days
 182.750            start time
 184.000            stop time
 183.750
 185.000
 184.750
 186.000
 185.750
 187.000
 186.750
 188.000
 187.750
```

wl.jul03.n (continued)

```
189.000
189.750
191.000
191.750
193.000
192.750
194.000
193.750
195.000
195.750
197.000
196.750
198.000
197.750
199.000
198.750
200.000
199.750
201.000
200.750
202.000
0.02
13
/dir3/people/philr/dynalysis/obs/filter/jul03/naple.jul03_0.obs.30
14
/dir3/people/philr/dynalysis/obs/filter/jul03/pensa.jul03_0.obs.30
15
/dir3/people/philr/dynalysis/obs/filter/jul03/pleas.jul03_0.obs.30
naple
/dir3/people/philr/dynalysis/compar/files.adj/tdl/jul03/naple.tdl.07012003_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/jul03/naple.dyn.07012003_adj
pensa
/dir3/people/philr/dynalysis/compar/files.adj/tdl/jul03/pensa.tdl.07012003_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/jul03/pensa.dyn.07012003_adj
pleas
/dir3/people/philr/dynalysis/compar/files.adj/tdl/jul03/pleas.tdl.07012003_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/jul03/pleas.dyn.07012003_adj
```

Similar structure for days 7/2 - 7/6, 7/8, 7/10 - 7/12, 7/14 - 7/18.

wl.jul03.n (continued)

naple
/dir3/people/philr/dynalysis/compar/files.adj/tdl/jul03/naple.tdl.07192003_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/jul03/naple.dyn.07192003_adj
pensa
/dir3/people/philr/dynalysis/compar/files.adj/tdl/jul03/pensa.tdl.07192003_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/jul03/pensa.dyn.07192003_adj
pleas
/dir3/people/philr/dynalysis/compar/files.adj/tdl/jul03/pleas.tdl.07192003_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/jul03/pleas.dyn.07192003_adj

IDL< .r plot_wlanal.pro

cnt.pleas_dyn.nov02

```
ps
landscape
0      idebug
!17Pleasure Pier!X
!17DYNALYSIS VS OBSERVED - WATER LEVEL (demeaned, 30hrlpf)!X
305.0   start time
335.0   end time
16      number of daily forecast files to read
 5      number of ticks
1       ilegnd
-2.00 2.00 8  yrange, and number of tick marks
juld
November 2002
feet
!17observed!X
/dir3/people/philr/dynalysis/obs/filter/nov02/pleas.nov02_0.obs.30
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11012002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11022002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11032002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11042002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11052002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11062002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11072002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11162002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11182002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11212002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11222002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11252002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11262002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11272002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11282002_adj
/dir3/people/philr/dynalysis/compar/files.adj/dyn/nov02/pleas.dyn.11302002_adj
```