Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh

16th Annual Progress Report
June 1995

Chapter 6:
Model Input System

Author: Ralph Finch

6 Model Input System

Background

Historically, the majority of effort expended in computer modeling of physical systems has gone into the numerical algorithms used in the computer programs. Other aspects of modeling, such as data quantity, quality, and availability; ease of use (data input and output structure); or the ability to rapidly analyze results, have not received comparable attention.

For the engineers who must use models on a daily basis to solve real-world problems, lack of progress in data availability and user interfaces has prevented them from taking full advantage of the reductions in time and increases in work quality possible by using computer models. However, by viewing models not as simply isolated computer programs or implementations of an algorithm, but rather as entire simulation systems, it is possible to better balance overall development so as to provide significant increases in overall system accuracy and speed, as opposed to only improvements in the numerical engines. This chapter describes a new input system developed at the California Department of Water Resources for flow and mass transport models in estuaries. Efforts to improve data availability to staff engineers were described in the 1991 and 1992 Annual Reports, and a project to improve the graphical output capabilities was described in 1992.

New Input System

The design goals of the new system were to increase ease of use of the models, decrease errors in setting up model runs, and improve the model by allowing more precise and consistent time specifications. Traditional Fortran model input is easy to program, but cumbersome to use. We shifted the effort from the user to the programmer, so that the new system is easier to use but now more difficult to change. We felt this was a reasonable tradeoff to make given past experience with traditional I/O systems. The current system, from a user's point of view, is difficult to learn, sometimes confusing and contradictory, and easy to make blunders.

Fixed Input

Estuary model input may be split into two categories: fixed and time-varying. Fixed input consists of the geometry, run control, input and output control, and so on, which does not varying with time during the model run. Here the concept of orthogonality is important, that is, each logical input section should not be mixed in with other sections. For instance, when the user changes the output control, that should in no way affect the geometry specifications. In the current system, different types of input are often mixed together.

Design goals for the fixed input were clarity, ease of use, and flexibility. Clarity simply means that a user looking at a new input file should be able to understand quickly what the intent of the input file is, without ambiguity. To this end, the input system uses keywords throughout that label different sections of input and also serve as input values. For instance, to specify gate positions, instead of using numbers (0 or 1), we use CLOSE or OPEN. Users can insert comments in the data files for documentation purposes, similarly to computer source code, as well as blank lines for additional ease in reading the input.

```
# Channel geometry information
# Use DSM numbering for channels and junctions

CHANNELS

CHAN    LENGTH  MANNING   DISP    DOWNNODE UPNODE XSECT
1       8400      0.037   0.150       2      1    1   2
2       8400      0.037   0.150       3      2    3   4
3      13600      0.035   0.150       4      3    5   6
END

# Run start and end times
SCALAR
run_start_date          31dec1989
run_start_time          2300
run_end_date            02jan1990
run_end_time            0500
#run_length             26hour      # if used, comment out
                                    #  run end date and time
time_step   5                       # time step, in minutes
END
```

Figure 6-1: Example of Format of Fixed Input to DSM2

For ease of use and flexibility, fixed input is specified in the files in terms of sections, for instance, channel geometry, or gate information. Each section consists of a header line specifying the section keyword; a field header line (except for single-value input), specifying which fields contain what data; the data values; and the END keyword, which marks the end of that section. Thus each section takes the appearance of a simple table (Figure 6-1). The user is never required to tell the computer how much input to expect, because the end of sections are marked. Fixed columns are never used; rather, data fields are separated by spaces so that users can prepare files that are easy to read.

One of the sections is used to specify other files to read, so that different logical sections can be in different files, at the user's option. Each section can be in one to many files, or one file can have several sections. The user can repeat sections, each subsequent section overlaying previous sections. This allows the use of a few main files containing generally stable, permanent information, stored in a central location. For a particular run a user can create new sections which just that information that is changed, not repeating the unchanged information.

Time-varying Input

Time-varying input are data such as boundary stage, flows, and water quality; internal flows and water quality, pumping rates, and gate operations. For this input, we choose to use a database written by the U.S. Army Corp of Engineers expressly for hydrologic data. The database is the Hydrologic Engineering Center Data Storage System (HECDSS). HECDSS is already in use with the DGUI, and is fast, does not consume excessive storage or computing resources, is available for a wide variety of computers (PCs, Unix machines, and mainframes), and may be used both interactively and as subroutine calls from Fortran. With HECDSS the timing of events are known precisely during the model run, and can be displayed very clearly in both the input and the output, eliminating ambiguity and confusion. This is an important step forward from the situation with our current model, which has a traditional I/O system using ASCII files only. In the current system, time-varying data is entered in fixed, mixed time steps of calendar days, tidal days, and hours, and there is virtually no indication of what time is associated with each data item.

With the new system using HECDSS, users may enter data using several time step intervals (15 minute, hourly, daily, and monthly), as well as irrgularly timed data. Each data stream is clearly labeled as to its interval, and the models automatically keep the different data streams synchronized during a run. Figure 6-2 shows a typical section of the fixed input that describes the time-varying input specifications (the INPUTPATHS section). Several features are worth noting here. First, for consistency and ease of use, the locations may be specified as place names, even though the model needs a channel and distance along the channel. The second section in Figure 6-2 shows how place names are translated to channel/distances. In the INPUTPATHS section, the time intervals of the data are different; the model keeps track of the data for each interval. Modifiers may be used to specify different data streams for different studies, etc. The FILLIN field refers to how values should be specified between exact data values; either use the last (previous) data value, or interpolate between data values. Finally, the DSS filename is given to indicate where each data stream is located. Not shown in Figure 6-2 are fields that allow the data stream to have a different start time than the model run.

```
INPUTPATHS
NAME  TYPE        INTERVAL    MODIFIER    FILLIN      FILENAME
MTZ   STAGE       15MIN       NONE        LAST        TEST.DSS
SJR   FLOW        1MONTH      NONE        INTERP      TEST.DSS
SAC   FLOW        1DAY        STUDY-A     INTERP      TEST2.DSS
CVP   FLOW        1DAY        NONE        LAST        TEST.DSS
END

# Translations from common name to channel-distance
TRANSLATION
NAME  CHAN  DIST
MTZ   441   LENGTH
SAC   410   0
SJR   1     0
CVP   216   0
END
```

Figure 6-2: Typical Section of Fixed Input Describing Time-Varying Specifications

To enter data into HECDSS, two basic methods are available. The first is to use standard HECDSS data input programs called DSSTS (regularly spaced data) or DSSITS (irregularly spaced data). These programs read time series data from an ASCII file (Figure 6-3), and translate it to a HECDSS file. The first few lines contain the HECDSS filename, the pathname of the data stream, units of the data, whether it is average or instaneous values, and the starting date. Subsequent lines contain the data itself, then the END keyword, and either further data streams or FINISH to end the data file.

The second method is designed for repetitive data and was written by Jatinder Singh of DWR. Since for most studies synthetic data with repetitive patterns is used for many of the time-varying inputs, a preprocessor is used to simplify the generation of data and storage into HECDSS. The core of the preprocessor is an ASCII file with one input description per line. A description specifies the start and end date and times, the time interval (hourly, daily, monthly, etc.), the location, study name, parameter (stage, flow, etc.), repeat information (e.g. repeat every hour), and the data value. This file is parsed with a program written in YACC and data written to a HECDSS file. With the repeat information, one line could expand to many different time values in the DSS file, saving the user much effort. To aid users in creating and changing the ASCII file in this method, a template is being written for the text editor GNU Emacs.

```
input.dss
/DELTA/MTZ/STAGE/01JUN1994/1HOUR//
FEET
INST-VAL
01JUN1994 0000
 -0.52
 -0.58
 -0.26
  0.32
  1.00
  1.50
END
/DELTA/SWP/FLOW/01JUN1994/1HOUR//
CFS
PER-AVER
01JUN1994 0000
  1000
  1000
  1000
  1500
  1500
  1500
END
FINISH
```

Figure 6-3: Example of HECDSS Input