Managing temporal data in a comprehensive modeling environment

Norman L. Jones, E. James Nelson and Colby T. Manwaring

ABSTRACT

Numerous graphical pre- and post-processors have been developed for numerical modeling. Many of these systems support numerical models that can be used for both steady state and transient simulations. Dealing with transient data can be particularly difficult in such an environment due to the differing time scales and units associated with transient field data and boundary conditions. A data model and a set of functions are presented in this paper as a simple, yet powerful strategy for managing temporal data in a comprehensive modeling environment. The strategy makes it possible to seamlessly integrate multiple types of transient data and convert the data to the appropriate values at the desired computational time intervals. In addition to simplifying the creation of transient models, the strategy makes it possible to automate much of the data manipulation required for model calibration.

Key words | database management, GIS, temporal data

Norman L. Jones (corresponding author)
E. James Nelson
Colby T. Manwaring
Brigham Young University,
Environmental Modeling Research Laboratory,
242 Clyde Building,
Provo, UT 84602, USA
Telephone +1 801 378 2812;
Fax +1 801 378 2478;
E-mail: njones@et.byu.edu

INTRODUCTION

The Environmental Modeling Research Laboratory of Brigham Young University, in cooperation with the US Army Engineer Waterways Experiment Station, has developed a suite of graphical numerical modeling environments. These systems include the Groundwater Modeling System (GMS), the Surface Water Modeling System (SMS), and the Watershed Modeling System (WMS). As the names imply, these systems are used to develop flow, transport, runoff, and environmental quality models. Each system is a comprehensive environment that includes sophisticated tools for a variety of tasks associated with pre- and post-processing of numerical models. These tasks include generating grids and meshes, assigning boundary conditions and model parameters, building and exchanging GIS data, launching numerical models, plotting of results, animating transient results, and calibrating models. Each of the three systems provides a graphical interface to a family of numerical models. Finite difference, finite volume, and finite element models are supported in multiple dimensions. For example, SMS includes interfaces to one-, two-, and three-dimensional surface water models including WSPRO, DAMBRK, FESWMS, RMA2, RMA4, SED2D, HIVEL2D, RMA10, CH3D, CEQUAL-ICM, CGWAVE, ADCIRC, and STWAVE.

In building a comprehensive modeling environment such as GMS, SMS, or WMS, a number of difficult programming and design challenges must be overcome with respect to data management, automated mesh and grid generation, and visualization. The difficulty is compounded by the fact that a large number of models are supported, in multiple dimensions, with several modeling approaches. One of the more challenging design problems associated with the development of a comprehensive modeling environment is dealing with temporal data. Most of the models supported by the three systems can be run in either steady state or transient mode and some are exclusively transient models.

Transient models are inherently more difficult to set up than steady state models since a time series must be defined for each of the boundary conditions and source terms. In addition, the output is more voluminous since a

solution is computed at each time step. In a comprehensive modeling environment supporting many different models, management of temporal data is complicated by the fact that each of the models may use a different method for defining transient input data and for outputting results. Further, the time scales associated with the models may range from seconds to years and it is often necessary to mix data defined with different time units. Model calibration can also be difficult because the observed field data are often recorded using a different format than the computed results.

In developing tools to manage temporal data, an *ad hoc* approach that focuses on each problem without consideration for an overall temporal data management strategy results in a cumbersome system that is difficult to develop and maintain. In this paper, we present a simple, comprehensive strategy for managing temporal data within a complex modeling environment. This strategy consists of a set of data structures and algorithms that provide an elegant solution for storing, editing, merging, and displaying temporal data.

PREVIOUS WORK

A number of products have been developed to aid in the management of temporal data. Two examples are HECDSS and NetCDF. HECDSS is a database that is customized for storing temporal data (US Army Corps of Engineers 1995). HECDSS was developed primarily for use with the HEC software (HEC-1, HEC-2, etc.) developed by the US Army Corps of Engineers Hydrologic Engineering Center in Davis, California. HECDSS is used to store transient rainfall and runoff data and hydrographs computed by the HEC software. Temporal data stored in HECDSS can be stored in any units, they can be stored in a date/time format, and the time interval can be fixed or variable.

NetCDF is also a database for storing temporal data (Rew *et al.* 1996). NetCDF was developed by the Unidata Program Center in Boulder, Colorado. NetCDF is similar to HECDSS in that data can be stored in a variety of formats and it is designed as a tool to support numerical modeling.

Another option for storing temporal data is a geographic information system (GIS). However, storing temporal data in a GIS is typically much more difficult than using a customized database such as HECDSS or NetCDF. A good overview of the advantages and disadvantages of different methods for storing temporal data can be found in Maicchi and Percini (1991).

The data management strategy described in this paper differs from the HECDSS and NetCDF approaches in that the strategy is not a database management system. Rather, it is a component of the model pre-processor that makes it possible to seamlessly integrate temporal data from a variety of sources (including both databases and manual input) in a variety of formats into the proper format and structure needed by the model.

REQUIREMENTS

Before presenting the temporal data management strategy, we will first review the requirements or objectives that the strategy must satisfy.

Conceptual model approach

The temporal data management strategy must be consistent with the conceptual model approach utilized by GMS, SMS, and WMS for building numerical models (Jones and Richards 1996; Nelson and Jones 1996; Richards and Jones 1996). With a traditional modeling approach, a grid or mesh is constructed and the boundary conditions, material properties, and other model parameters are assigned directly to nodes, elements, and grid cells. This process is tedious and labor-intensive. Furthermore, any extensive editing or rebuilding of the grid requires that the model parameters be completely reassigned.

In order to provide a more powerful and flexible modeling environment, a grid independent method has been developed in GMS, SMS, and WMS. This method utilizes GIS objects to define a conceptual model representation of the numerical model. With such an approach, it is possible to define boundary conditions and

source data independently from the numerical grid or mesh. Before a grid is constructed, a conceptual model is constructed using points, lines, and polygons and the model parameters are assigned directly to these objects. For example, with a groundwater model, the problem domain and areal features such as lakes or recharge zones are defined with polygons, linear features such as rivers and cutoff walls are defined with lines, and point features such as injection and extraction wells are defined with points. Values such as conductance, elevation, and flux rate are assigned directly to the objects. Once the conceptual model is defined, a grid or mesh is automatically constructed to fit the problem domain and all boundary conditions and model parameters are automatically assigned to the appropriate cells.

The conceptual model approach has numerous advantages. If a significant change is required in the model, the change can be made to the GIS objects in the conceptual model and the numerical grid can be regenerated or updated in seconds. Furthermore, the conceptual model can be model independent. The same conceptual model can often be used to define a numerical model for several different models. Both finite element and finite difference grids can be generated from the same set of GIS objects.

When building a conceptual model, the GIS objects provide a model-independent method for defining spatially varying input parameters. When the conceptual model is converted to a numerical model, these parameters are discretized to grids, cells, and nodes. However, in addition to the spatial data, much of the data required for a conceptual model such as boundary conditions and internal sources and sinks are temporal in nature. Therefore, the temporal data management strategy must be consistent with the conceptual model approach. It must be possible to enter the temporal data in a model-independent fashion. When the conceptual model is converted, the temporal data are discretized not to grid cells, but to time steps.

Mixing time scales

The temporal data management strategy must be capable of dealing with a variety of time scales (s, m, h, etc.). This is particularly important when using the conceptual model approach. For example, the temporal data on the conceptual model may be initially defined in units of days. If the conceptual model is truly model-independent, the user should be able to set up and run the numerical model using time units of hours or years and the units conversion should take place automatically. Furthermore, when defining the conceptual model, it should be possible to use one time unit for one object and a different time unit for another object and have all temporal data converted to the correct unit when the conceptual model is converted.

Date/time entry

When entering temporal data, the traditional approach is to define a starting time for the simulation as time zero and have all time values be relative this value (e.g. t_0=0.0 hrs, t_1=1.0 hrs, t_2=2.0 hrs, etc.). However, this often forces the user to do a great deal of data conversion as the model is built. Temporal data such as rainfall rates and river elevations are typically recorded initially in a date/time format (19/11/1997 18:23:19). Furthermore, if a new beginning time for the simulation is chosen or if a different time unit is selected, the conversion process must be repeated. Ideally, the temporal data would be entered to the conceptual model in a date/time format and all of the conversions to model time steps would take place automatically.

Model calibration

Finally, the temporal data management strategy must support model calibration. Model calibration is a process wherein the user iteratively modifies the model parameters until the computed model results match the observed field data within an acceptable tolerance. Managing temporal observed data can be a difficult process since the observations are often recorded in a date/time format as described above. Thus, it should be possible to enter the observed data in a date/time format to avoid conversion problems. Further, the output from the numerical models must be stored in a fashion that facilitates seamless conversion to the date/time format so

Time	Head
0.0	5.5
2.0	6.7
4.0	8.7
6.0	12.4
8.0	14.5
10.0	16.6
12.0	17.1
14.0	16.8
16.0	13.1
18.0	8.5
20.0	7.3
22.0	7.2
24.0	7.2
26.0	7.1
28.0	6.8
30.0	6.5

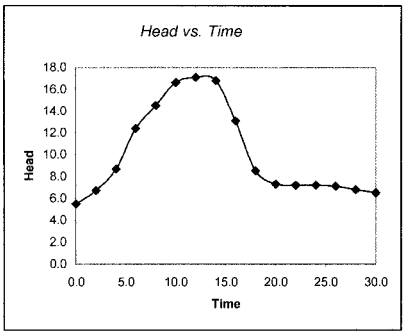

Figure 1 | A set of time/value pairs define a time series.

that comparisons between computed and observed values can be performed automatically.

TEMPORAL DATA MANAGEMENT STRATEGY

We will now introduce a strategy for temporal data management that satisfies each of the requirements listed above. First, we will introduce a data model for temporal data. Then, we will describe a set of algorithms that operate on the data model. Finally, we will illustrate how the data model and algorithms satisfy the design requirements.

TEMPORAL DATA MODEL

The temporal data model consists of a set of objects for storing temporal data. The basic components of the data model are a time series and a reference time.

Time series

The primary object for storing temporal data is a time series. A time series is used to define how a boundary condition, model parameter, or field-observed value varies with time. A time series consists of a set of time/value pairs as shown in Figure 1. The sequence of pairs defines a piecewise linear curve. The time series can start at any time value and end at any time value. Furthermore, any number of points may be used to define the curve and the time interval between points may vary. As a result, the curve is completely independent of the model time steps (the process of converting between a generic time series and the model time steps is described below).

Base unit

Each time series has a 'base unit'. The base unit is years, days, hours, minutes, or seconds (weeks and months are not allowed for the base unit). The base unit is the unit that the time values in the time series are expressed in.

Reference time

Another basic part of the data model is a 'reference time'. A reference time consists of a date and time value in the form:

```
day/month/year hour:minute:second
```

For example:

```
6/12/1995 13:45:30.34
```

All values are stored as integers except for the seconds units, which may have a fractional component. A reference time is stored with each time series. The reference time represents the date and time corresponding to a time value of zero, i.e. the values stored in the time series are relative values with respect to the reference time.

When combined with a time series and a base unit, the reference time serves to 'register' the time series to a common time scale. This makes it possible to merge, compare, and otherwise manage a database of time series within a comprehensive modeling environment.

Relative times vs absolute times

Once a reference time has been associated with a time series, the time values can be displayed using one of two formats: absolute times and relative times. Relative times represent the native time values used to define the time

series. Absolute times represent date/time values. For example, suppose that the base unit is hours and that the following reference time and time series have been specified:

```
6/12/1995 13:00:00.0
time   head
0.0    34.5
2.0    36.7
6.0    38.3
12.0   38.0
```

The preceding times are relative times. Using the reference time, the following absolute times can be computed:

time	head
6/12/1995 13:00:00.0	34.5
6/12/1995 15:00:00.0	36.7
6/12/1995 19:00:00.0	38.3
6/12/1995 01:00:00.0	38.0

While relative times are more efficient for data storage, there are many occasions where it is useful to list each of the values in the time series using the absolute time format. As explained above, observed field data and boundary condition data are often recorded in the date/time format. When entering a time series in GMS, SMS, or WMS, the user first enters a reference time. Since time values are easily converted between the relative and absolute format, the time/value pairs can then be entered using whichever format is most convenient for the particular situation.

Transient model output

The time series object described above is generally used to define transient input data or transient field observations. Another type of temporal data is the output from numerical models. In GMS, SMS, and WMS, this type of output is called a transient data set. A transient data set consists of a sequence of data sets, each of which has a time value associated with it. An individual data set consists of a set of values (vector or scalar), where there is one value defined for each node or cell in the computational grid.

Transient data sets can be managed in a manner similar to that used for time series. The time value associated with each individual data set is defined using a relative time. A reference time is then defined for the entire transient data set. This reference time can then be used to convert the relative times to absolute times. As with the time series, the reference time 'registers' the transient data set to a common time scale and simplifies the management of the transient data sets as part of the overall temporal data management strategy.

BASIC FUNCTIONS

Next, we will define a set of simple functions associated with the temporal data model. The data model and the functions can then be used as a fundamental set of tools for building more complicated algorithms to solve problems associated with temporal data management. The essential functions are as shown in Table 1.

APPLICATIONS

Once the data model and the basic functions described above are implemented, each of the objectives of the overall temporal data management strategy can be achieved. Next, we will summarize solutions to the basic requirements listed above.

Conceptual model

The temporal data model and basic functions make it possible to support the definition of numerical models using the conceptual model approach. Each of the transient sources/sinks and boundary conditions associated with the model can be defined using a time series. This time series can be associated with any of the GIS objects (points, lines, or polygons). When the time series is entered, it can be defined using any base unit, and any reference time, i.e. the base unit and reference time can vary for different objects in the same conceptual model.

Table 1 | The basic functions for managing temporal data

Name	***scale_rel_time***
Input	A relative time, an old base unit, and a new base unit
Output	A relative time
Description	The relative time is converted to a new base unit by multiplying by the appropriate scaling factor
Name	***scale_time_series***
Input	A time series and a new base unit
Output	A time series
Description	The time series is converted to a new base unit by converting the reference time and each of the relative times associated with the time series using the ***scale_rel_time*** function
Name	***abs_time***
Input	A reference (absolute) time and a time value (relative)
Output	An absolute time
Description	Computes an absolute time for a given relative time value
Name	***rel_time***
Input	Two absolute time values and a base unit
Output	A relative time
Description	Computes the relative time difference between the two absolute times in the specified unit
Name	***translate_time_series***
Input	A time series and a new reference time
Output	A time series
Description	The time series is converted to the new reference time by first computing the time difference between the old and new reference times using the ***rel_time*** function. This time difference is then added to each of the time values in the time series

The critical part of the conceptual model approach is how the conceptual model is converted to the grid- or mesh-based numerical model. The numerical model requires time values defined in relative times and at specific computational time intervals. In order to convert the conceptual model, a simulation time must be defined for the numerical model. The simulation time represents the absolute time (date/time) corresponding to a relative time value of zero (i.e. the starting time of the simulation). The computational time steps are then defined in the traditional manner using the simulation time as a base time of zero. When the conceptual model is converted to the numerical model, the temporal data are converted using the following steps:

1. Each of the time series in the conceptual model is modified so that the base unit and the reference time are consistent with the simulation time. This is accomplished using the ***scale_time_series*** and ***translate_time_series*** functions.
2. For each of the computational time intervals, an appropriate value is extracted from each of the time series in the conceptual model and applied to the corresponding boundary condition or source/sink in the numerical model.

The manner in which the value is extracted from the time series in the second step must be consistent with the approach used by the numerical model to define temporal data. In most cases, the value of a boundary condition or source/sink associated with a computational time interval represents a value that is applied at the beginning of the interval and held constant over the interval. Thus, while the time series allows a piecewise linear definition of the value vs time, the numerical model often requires a step function. Since the step function value essentially represents the average value over the duration of the time step, the process illustrated in Figure 2 can be used to extract an appropriate value from the time series. The value for each time step is determined as follows:

1. The relative times at the beginning (t_b) and the end (t_e) of the time step are determined.

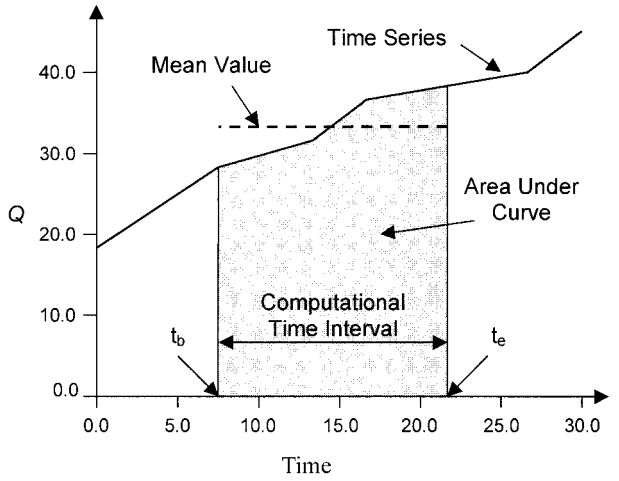

Figure 2 | Extracting an appropriate value for a computational time step from a time series curve.

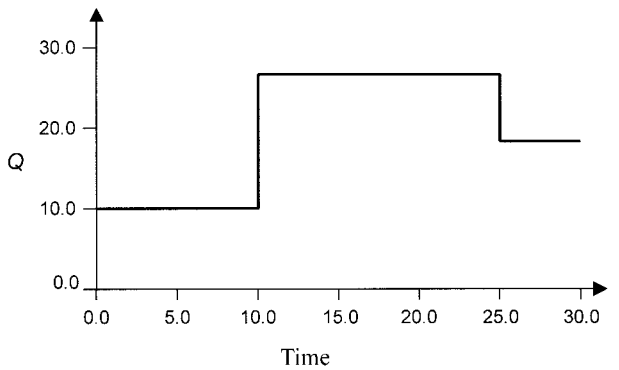

Figure 3 | A step-function time series.

2. The area under the time series curve over the time interval is found by integrating the piecewise linear curve from t_b to t_e.
3. The integral found in step 2 is divided by the time step length to get the mean value over the time step.

In order for the conversion process to work effectively, it is often necessary to ensure that computational time intervals correspond to key points in the time series. For example, a time series for the pumping rate of an extraction well is shown in Figure 3. Since the pumping rate is essentially a step function, a computational time step must begin at $t = 10.0$ and $t = 25.0$ in order for the instantaneous change in the pumping rate to be correctly represented in the numerical model. It is possible to implement the conceptual model conversion process such that the computational time intervals are automatically modified to ensure that a time step begins at each significant discontinuity in the time series curves.

Model output

As mentioned above, model output is typically in the form of a transient data set. The transient data set consists of a series of individual data sets, one for each computational time interval. Each of the individual data sets has a relative time value. If a reference time is stored with the transient data set, an absolute time value can be computed for each individual data set using the ***abs_time*** function.

As mentioned in the previous section, a reference time called the 'simulation time' is associated with each numerical model. This simulation time is the appropriate reference time for all transient data sets output by the model. Thus, a reference time can be easily attached with each transient data set by modifying each analysis code to read the simulation time and write it out with all transient data sets when they are written to disk. The data sets are then properly registered to the global time scale and the data sets can be manipulated independently of the rest of the numerical model data.

Model calibration

The temporal data model and basic functions make it possible to develop seamless model calibration tools. During the model calibration stage, output from the model is interpolated to observation points in the domain of the model and compared to field observed values. In the case of transient models, the observed data are entered as a time series. A reference time is associated with the time series and each of the observed values can be directly entered using a date/time format since this is the format the data are typically recorded in. As described in the previous section, each of the transient data sets has an associated simulation time. The ***scale_time_series*** and ***translate_time_series*** functions can be used to modify the observed time series so that it matches the time unit and the reference time of the transient data set. The ***abs_time***

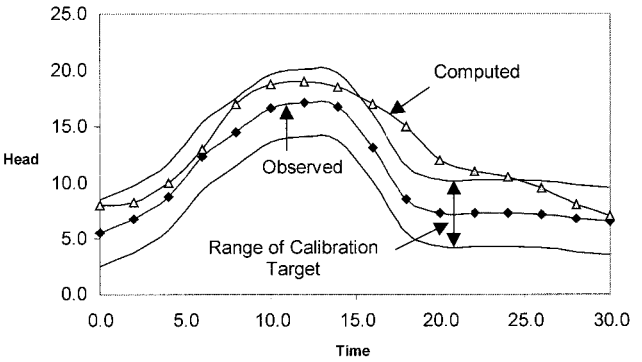

Figure 4 | A transient calibration plot.

function can then be used to compute an observed value at each of the time steps in the transient data set. This makes it possible to automatically generate a transient calibration plot such as the one shown in Figure 4.

The calibration plot shown in Figure 4 is typically used with values observed at a single point such as water surface elevation or concentration. Another important type of observation is a flux observation. The temporal data management strategy described in this paper can greatly simplify flux observations. For example, for groundwater modeling problems, flux observations are typically made at streams or reservoirs. Based on these measurements, estimates are made of the quantity of water gained or lost between the body of water and the aquifer. While these estimates are often made in terms of a flux rate, it is also common to measure a volume of water gained/lost over a specific time interval. With the data model and functions described in this paper, it is possible to enter a flux observation simply as a volume of water with a beginning date/time and an ending date/time corresponding to the period of measurement. When the solution is computed, the reference time associated with the solution can be used to automatically compute the volumetric flux from the model over the observed time period.

CONCLUSIONS

The data model and algorithms described in this paper represent a simple, yet powerful solution to the daunting task of managing temporal data in a comprehensive modeling environment. The management strategy, coupled with a conceptual model approach to model development, makes it possible to directly import transient field observed data and transient boundary condition data with little or no modification. These data can be entered in either a relative or date/time format and units may be arbitrarily mixed. These data can then be automatically converted to the correct reference time and units of the numerical model. Post-processing and model calibration are also supported as part of the overall strategy.

ACKNOWLEDGEMENTS

The research described in this paper was partly funded by the U.S. Army Engineer Waterways Experiment Station (WES) in Vicksburg, Mississippi. The support received from WES is gratefully acknowledged.

REFERENCES

Jones, N. L. & Richards, D. R. 1996 A conceptual model approach to hydroinformatics. In *Proceedings of the Second International Conference on Hydroinformatics*, Zurich, Switzerland, 9–13 September, 1996, pp. 285–291.

Maicchi, R. & Percini, B. 1991 Temporal data management systems: a comparative view. *IEEE Trans. Knowledge Data Engng* **3**(4), 504–524.

Nelson, E. J. & Jones, N. L. 1996 Using the ARC/INFO data model to build conceptual models for environmental/hydraulic/hydrologic simulations. In *Proceedings of the 1996 ESRI User Conference*, 20–24 May, 1996.

Rew, R. K., Davis, G. P., Emmerson, S. & Davies, H. 1996 *NetCDF User's Guide, An Interface for Data Access*. (*http://www.unidata.ucar.edu/packages/NetCDF/docs.html*).

Richards, D. R. & Jones, N. L. 1996 The DoD Groundwater Modeling System: a conceptual model approach. In *Proceedings of the ASCE North American Water and Environment Congress*, Anaheim, California, 22–28 June, 1996, 6 pp.

US Army Corps of Engineers, Hydrologic Engineering Center 1995 *HEC-DSS User's Guide and Utility Manuals*. (*http://elder.water.ca.gov/doc/hec-dss/*).