CHAPTER 1
CONCEPTS AND DEFINITIONS

1.1 INTRODUCTION

TRNSYS is a transient system simulation program with a modular structure. The modular nature of TRNSYS gives the program tremendous flexibility, and facilitates the addition to the program of mathematical models not included in the standard TRNSYS library. TRNSYS is well suited to detailed analyses of systems whose behavior is dependent on the passage of time.

The purpose of this chapter is to provide the user with the concepts and definitions he or she will need to fully understand the use of TRNSYS for modular system simulation. In addition to the information contained in this chapter, the user is expected to have some acquaintance with FORTRAN programming, although a thorough knowledge of programming should not be necessary. The steps which should be followed to prepare a system for simulation using TRNSYS are summarized in Section 1.11.

1.2 A SYSTEM AND ITS COMPONENTS

A system is defined to be a set of components, interconnected in such a manner as to accomplish a specified task. For example, a typical solar water-heating system may consist of a solar collector, an energy storage unit, an auxiliary energy heater, a pump and several temperature sensing controllers. One obvious characteristic of a system is its modularity. Because the system consists of components, it is possible to simulate the performance of the system by collectively simulating the performance of the interconnected components.

The performance of a system component will normally depend upon characteristic fixed parameters, the performance (or outputs) of other components, and time-dependent forcing functions. For example, for a solar water-heating system, knowledge of the weather (i.e., solar radiation, ambient temperature etc.) and the hot water demand as a function of time are necessary in order to determine the transient system performance. It is important to realize that time-dependent forcing functions can be thought of as outputs of specialized system components and they can thus be treated in the same manner as any other component.

The modular simulation technique greatly reduces the complexity of system simulation because it essentially reduces a large problem into a number of smaller problems, each of which can be more easily solved independently. In addition, many components are common to different systems and, provided that the performance of these components is described in a general form, they can be used in many different systems with little or no modification. This feature makes modular simulation most attractive.

With a program such as TRNSYS which has the capability of interconnecting system components in any desired manner, solving differential equations, and facilitating information output, the entire problem of system simulation reduces to a problem of identifying all of the components and formulating a general mathematical description of each.

In some cases, the description of a component may be complex. In Chapter 4, descriptions of many of components of common thermal systems are presented; in addition, components common to all simulations, such as a quantity integrator, printer and plotter are also presented. The user intending to apply TRNSYS to the simulation of systems containing components not described in Chapter 4 must formulate his or her own mathematical description in FORTRAN subroutines consistent with the requirements of TRNSYS as described in Chapter 3.

1.3 INFORMATION FLOW DIAGRAMS

Once all of the components of a system have been identified and a mathematical description of each component is available, it is necessary to construct an information flow diagram for the system. An information flow diagram is a schematic representation of the flow of information into and out of each of the system components. In the diagram, each component is represented as a box. Each piece of information required to completely describe the component is represented as an arrow directed into the box. Each piece of information calculated by equations describing the component can be represented as an arrow directed out of the box.

It is often helpful to think of the arrows connecting component inputs and outputs as information exchanged via pipes and wires in a real system. A collector outlet flowstream temperature and flowrate connected to the inlet of some other piece of hardware is "information" transmitted through a pipe. A controller on-off output connected to a pump is information transmitted through a wire. The analogy between information flow and pipes and wires is, however, not perfect. Information need not always follow the course of pipes and/or wires.

In order to demonstrate the construction of an information flow diagram, consider a very simple solar water-heating system consisting of a solar collector and an auxiliary energy heater as shown in Fig. 1.3.1. Cold water, at a temperature T_i (varying with time), is circulated at a constant rate $\dot{m}$ through the collector. The water is then heated from T_o, the temperature of the water leaving the collector, to a desired temperature T_{set} by the auxiliary heater. The problem is to determine Q_{aux}, the total auxiliary energy required to heat the water to the desired temperature over a specified time period.

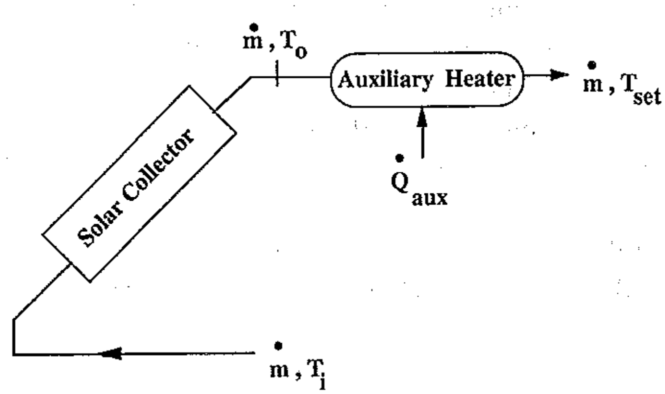


Figure 1.3.1 Simple solar water heating system

Component models which describe the performance of the solar collector and the auxiliary heater have been formulated in a general manner as described in Chapter 4. Ordinarily, general formulations of component models are desirable, since the model can then be used in a variety of simulations without modification. The models in Chapter 4, however, require consideration of details which distract from the purpose at hand. Rather than using these general models to demonstrate the construction of information flow diagrams, let us formulate component models specific to this example.

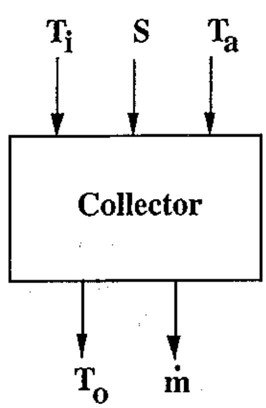
An equation relating the collector outlet water temperature to its inlet temperature can be written

$$T_o = T_i + \frac{AF_R}{\dot{m}C_p}[S - U_L(T_i - T_a)] \qquad (1.3.1)$$

where

T_i	is the inlet water temperature
A	is the area of the collector
C_p	is the heat capacity of water
F_R	is a constant efficiency factor
U_L	is a constant loss coefficient
S	is the time dependent solar radiation absorbed on the collector surface
T_a	is the time dependent ambient temperature

An information flow diagram for this collector model is shown below:



Parameters

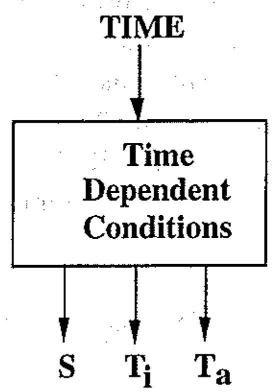
1. **A**
2. **F_R**
3. **U_L**
4. **$\dot{m}$**
5. **C_p**

Note that S, T_i and T_a are only functions of time.

$$S = f_1 \text{ (time)} \qquad (1.3.2)$$
$$T_i = f_2 \text{ (time)} \qquad (1.3.3)$$
$$T_a = f_3 \text{ (time)} \qquad (1.3.4)$$

Time dependent conditions such as S, T_a and T_i are considered to be outputs of a specialized system component, a data reader, which reads the values of S, T_i, and T_a at successive increments of time.

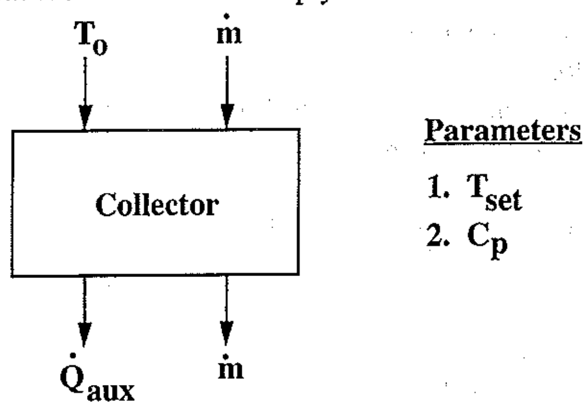
An information flow diagram relating S, Ti and Ta, to time is shown below.



The instantaneous auxiliary energy required, $\dot{Q}_{aux}$, is described by the following equation

$$\dot{Q}_{aux} = \dot{m}C_p\,[T_{set} - T_o] \tag{1.3.5}$$

An information flow diagram for the heater is simply

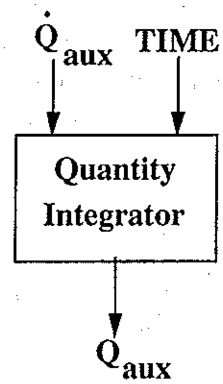


In order to determine the total auxiliary energy required, Q_{aux}, the instantaneous auxiliary energy must be summed or integrated over the period of operation. For this purpose, it is necessary to include a "quantity integrator" as one of the system components. Note that a quantity integrator component is used only to integrate some calculated quantity over a period of time. The integrator is not used to solve first-order differential equations which may be part of the mathematical description of a component because TRNSYS will automatically perform this type of integration. A quantity integrator is treated as any other system component.

The equation describing the quantity integrator is

$$Q_{aux} = \int_{Time} \dot{Q}_{aux}\, dt \tag{1.3.6}$$

The information flow diagram for the quantity integrator is shown below.

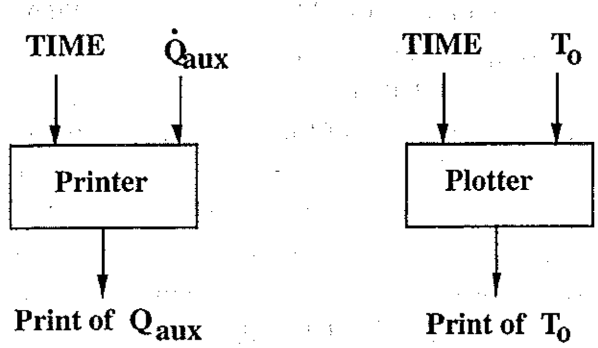


A quantity integrator is needed in the evaluation of many transient systems. Thus, a multivariable quantity integrator component is built into TRNSYS (see Section 4.1.5, TYPE 24, and Section 4.10.4, TYPE 28).

The information flow diagram for the simple solar water heating system of Figure 1.3.1 is now complete except for a component which allows the results of the simulation to be made available to the user. For this purpose, TRNSYS has printers, plotters and other output components built in to print and/or plot desired results (see Section 4.10). Alternatively, the user may write application specific output routines (see Section 3.3.3). The analogous pieces of equipment in a physical system would perhaps be a multi-channel digital display and/or strip chart recorders. In a real situation, these would monitor and display various information streams so that the system performance could be followed.

It is absolutely essential to include an output device (i.e., TYPE 25, 26, 27, 28, 29 or 65 or a user-written output type) in the system information flow diagram. Failure to include one of these components will result in a working simulation with no output. In the example being considered, the user may wish to print Qaux as the integration of $\dot{Q}_{aux}$ progresses and to plot the variation of T_o (collector exit temperature) with time.

Information flow diagrams for the printer and the plotter are



The information flow diagram of a system is constructed by joining all of the diagrams of the system components. The information flow diagram of the solar water-heating system is shown in Figure 1.3.2. Time is not shown explicitly, since it is a built-in input to all components in TRNSYS. Note that the collector outlet temperature, To, serves as an input to both the heater and the plotter. In general, an output of one component can be used as an input to any number of other components.

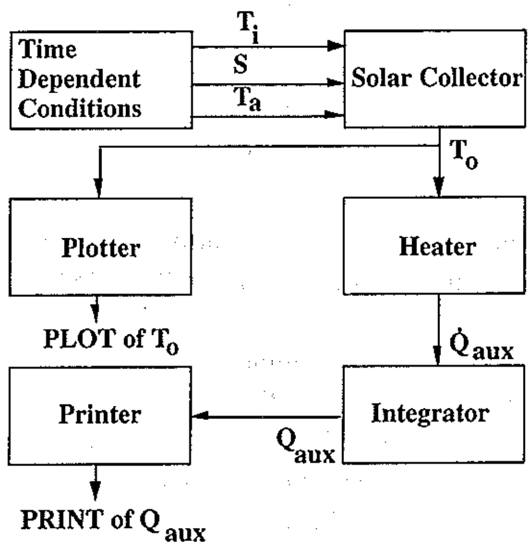


Figure 1.3.2: Example flow diagram for a solar water heating system.

1.4 COMPONENT TYPE NUMBERS

TRNSYS identifies the different kinds of components which appear in the information flow diagram by associating a TYPE number with each kind of component. For example, the solar collector model provided with TRNSYS is designated a TYPE 1 component. The pump model provided with TRNSYS is designated as a TYPE 3 component. The fact that there may be more than one pump in the system is immaterial. Each pump will be referred to as a TYPE 3 component provided that the performance of each is described by the same FORTRAN subroutine.

The relation between a component and its TYPE number is determined by the name of the FORTRAN subroutine coded to model that component. As explained in Chapter 3, every component subroutine must have the name TYPEn, where n is an integer from 1 to 99. Thus, a flat-plate solar collector will be a TYPE 1 component provided that there is a subroutine TYPE1 which models the collector performance. TYPE numbers are already associated with all the component models described in Chapter 4. The apparently arbitrary numbering of component models stems from the historical order in which the models were written. The assignment of TYPE numbers to user supplied component models is discussed in Chapter 3.

1.5 SYSTEM UNIT NUMBERS

TRNSYS recognizes the position of each component in the information flow diagram by a unique but arbitrary user-assigned UNIT number between 1 and n, where n is set in the file param.inc and has a default value of 80. See Section 6.5 for more information. Since a simulation may include more than one component of a particular kind (for example, two TYPE 3 pumps), the TYPE number does not always uniquely identify a component in the information flow diagram. The UNIT number simply provides a reference number for each system component. UNIT numbers are unrelated to TYPE numbers, and need not be in sequential order. The only restrictions imposed on UNIT number selection are that no two system components can have the same UNIT number and each UNIT number must be a positive integer between 1 and n. As an example, the information flow diagram in Figure 1.3.7 has been redrawn in Figure 1.5.1 showing UNIT and TYPE numbers for the system components.

Thus, associated with each component in the information flow diagram is a TYPE number, which defines the component's function, and a UNIT number, which distinguishes the component from all other components in the system. A system which includes two pumps can be said to include two UNITS of TYPE 3; the pumps might (for example) be labeled UNIT 3 and UNIT 43. To minimize confusion between UNIT and TYPE numbers, it may be helpful to assign UNIT numbers corresponding to component TYPE numbers whenever possible.

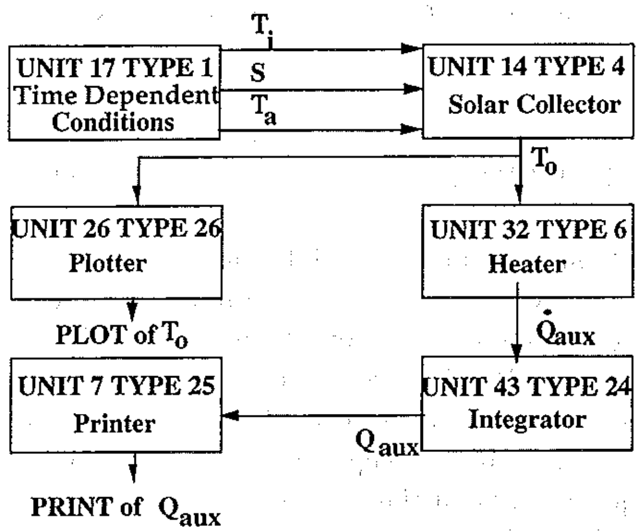


Figure 1.5.1 An information diagram with component TYPE and UNIT numbers.

1.6 TYPES OF INFORMATION FLOW

It is necessary to distinguish among several types of information flow which may occur in the information flow diagram. The most obvious distinction is between information flowing into and that flowing out of a component. The set of information flowing out of a component, represented by outwardly directed arrows, is defined to be the OUTPUT variable set for that component. The OUTPUT variable set of the collector in the information flow diagram shown in Figure 1.5.1 consists of T_o, the calculated collector outlet fluid temperature.

The information flowing into a component, represented by inwardly directed arrows, can be of three types.

1. Those pieces of information flowing into a component which are OUTPUT variables from any other component in the system (or are defined by CONSTANTS or EQUATIONS commands--to be discussed later) constitute what is defined to be the INPUT variable set for the component. The INPUT variables are those variables whose values may vary during the simulation. For a transient system, the INPUT variables may vary with time. The INPUT variable set of the collector in Figure 1.5.1 consists of S, T_i, and T_a. All three are functions of time.
2. Those pieces of information which are always constant throughout the simulation and are of interest to the component are the PARAMETERS of the component. The PARAMETERS of the collector in Figure 1.5.1 are A, F_R, U_L, $\dot{m}$ and C_p.
3. Finally the variable TIME, which is neither an INPUT variable nor a PARAMETER, must be distinguished. TIME is handled internally by TRNSYS and normally need not concern the user.

OUTPUT variables, INPUT variables, PARAMETERS, and TIME represent all the types of information flow which may occur in the information flow diagram of a system. It is important to distinguish among these four types of information flow because TRNSYS treats each type differently.

1.7 ORDERING OF PARAMETERS, INPUTS AND OUTPUTS

As explained in Section 1.6, four types of information flow may exist in the information flow diagram. A component may receive three types of information; INPUTS, PARAMETERS and TIME. The information flowing from the component is its OUTPUTS. To aid in conveying the information flow diagram to TRNSYS, each piece of information of a given type (for each component) is numbered sequentially. If, for example, a component has 2 INPUTS, they are numbered INPUT 1 and INPUT 2. If the component has PARAMETERS and OUTPUTS, they are likewise numbered sequentially beginning with number 1. The sequential numbering of the INPUTS, OUTPUTS, and PARAMETERS of each component must be consistent with the numbering used in the FORTRAN subroutine modeling the component as explained in Chapter 3. For example, the FORTRAN subroutine modeling a heat exchanger (TYPE 5) described in Section 4.7.1 requires four PARAMETERS. According to this description, the first PARAMETER is the code number indicating the type of heat exchanger. The second PARAMETER is the product of the heat exchanger area and its heat transfer coefficient, UA. The third and fourth PARAMETERS are the heat capacities of the hot and cold streams respectively. These PARAMETERS <u>must</u> be sequentially numbered in the information flow diagram in the order that the heat exchanger subroutine (TYPE 5) expects them. The same is true for the INPUTS, the OUTPUTS, and the PARAMETERS of every system component. In the Component Descriptions manual and in Chapter 4 of this manual, the numbering convention for each of the components modeled will be found with the description of the model.

1.8 TRANSIENT AND STEADY-STATE SIMULATION

If TIME is a variable in the model of any of the components of the system, the simulation of the system is defined to be a transient simulation. In general, a transient simulation will be required if any of the three following situations occur.

1. A component of the system has an OUTPUT variable which is a function of time.
2. The mathematical description of a component of the system involves one or more time-dependent differential equations (with non-zero derivatives).
3. A physical quantity calculated by the simulation model is to be integrated over time by the quantity integrator component.

Although situations 2 and 3 both call for numerical integration, they are conceptually different and TRNSYS treats them differently. If the mathematical description of a component involves one or more differential equations, TRNSYS must know the number of differential equations involved and the values of the dependent variables at the start of the simulation (see Sections 1.10.2, 2.3.4, and 3.3.2). TRNSYS expects that the derivatives of the dependent variables at any time will be evaluated by the component model. Using the supplied values of the derivatives, TRNSYS automatically integrates to evaluate the dependent variables. (The subroutine DIFFEQ provides an alternative method of differential equation solving. See Sections 1.10.1 and 3.4.4.)

If a quantity calculated by the simulation model is to be integrated over time, TRNSYS requires that a quantity integrator component be included as one of the system components in the information flow diagram. As shown in the system of Figure 1.5.1, the quantity to be integrated is supplied to the TYPE 24 integrator component as an INPUT variable. The OUTPUT variable of the integrator is the integrated value of the INPUT variable. Two of the output producing components, TYPE 27 and TYPE 28, include built-in quantity integrators as does the TYPE 55 periodic integrator.

System simulations which are not transient are referred to as steady-state simulations. A steady-state system is one in which none of the system variables change with time. TRNSYS can be used for steady-state or transient simulations.

1.9 ACYCLIC AND RECYCLIC INFORMATION FLOW

An information flow diagram of a system may exhibit either acyclic or recyclic flow of information. Recyclic flow occurs whenever there is a path in the information flow diagram formed by the output arrows which leads from a component to one or more other components of the system and then back to the starting component. The diagram of Figure 1.5.1 is an example of acyclic information flow since no such path exists. If, however, the system represented by Figure 1.5.1 were modified so that the water flow rate $\dot{m}$ is controlled by the difference between temperatures T_o and T_i, the resulting information diagram would exhibit a recyclic flow of information as indicated in Figure 1.9.1. (Note that in Figure 1.9.1, $\dot{m}$ must now be an INPUT variable to the collector and not a PARAMETER as it was in Figure 1.5.1 since its value may change with time.)

When recyclic flow occurs in the information flow diagram of the system, a numerical technique is required to find the values of the OUTPUT variables which satisfy the equations of all of the components involved. This is especially true when none of the OUTPUT variables forming the recyclic loop are the solution of time dependent differential equation. TRNSYS is programmed to identify such problems should they exist, and, if necessary, will use a numerical solution technique, iterating until all of the OUTPUT variables (I_τ) (which are INPUT variables to other components) in the recycle loop converge to within:

$$\left|I_\tau^{\,i} - I_\tau^{\,i-1}\right| < \zeta_A$$

or

$$\left|I_\tau^{\,i} - I_\tau^{\,i-1}\right| < \max\left(\varepsilon_A, \varepsilon_A * I_\tau^{\,i}\right)$$

where ε_A or ζ_A have been specified by the user (see Section 2.2.2). The action taken if this convergence tolerance cannot be met within a reasonable number of iterations is described in Section 2.2.3. A description of the numerical techniques available to solve recyclic information flow situations are discussed in Section 2.2.12. With the release of version 14, TRNSYS now has two different algorithms for solving recyclic problems.

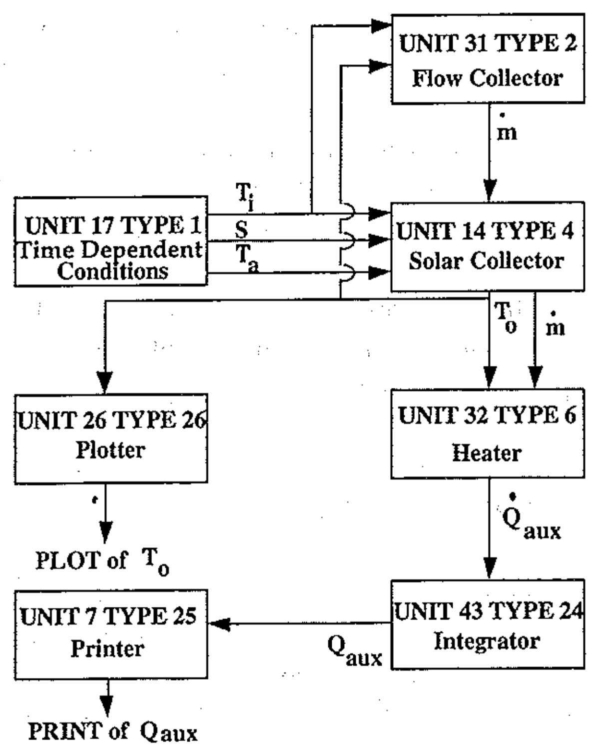


Figure 1.9.1 Information flow diagram with recycle.

1.10 SOLUTION OF DIFFERENTIAL EQUATIONS

Prior to TRNSYS version 11.1, most differential equations were solved numerically. To decrease simulation time, many of these differential equations are now solved analytically or have been rewritten as algebraic equations. The following two sections discuss the analytical and numerical methods used in the TRNSYS program to solve differential equations. The last section discusses convergence and integration.

1.10.1 ANALYTICAL SOLUTIONS

Component models in TRNSYS contain differential equations which can often be formulated as

$$\frac{dT}{dt} = aT + b \tag{1.10.1.1}$$

where T is the dependent variable; t is time; a is a constant; and b may be a function of time and/or the dependent variable. If b is a constant, then the exact solution to this differential equation can easily be determined. If b is not a constant, then a reasonable approximation to the analytical solution is achieved by assuming b is constant over each time step, equal to its average value. At any time, τ, for $a \neq 0$

$$T_\tau = \left(T_{\tau\text{-}\Delta t} + \frac{\bar{b}}{a}\right) e^{a\Delta t} - \frac{\bar{b}}{a} \tag{1.10.1.2}$$

where Δt is the timestep (a constant value specified by the user) and $\bar{b}$ is the average value of b over the interval. If b is a linear function of time, then

$$\bar{b} = b\left(\tau - \frac{\Delta t}{2}\right) \tag{1.10.1.3}$$

If b is expressed as a function of the dependent variable, T, then a good approximation for $\bar{b}$ is

$$\bar{b} = b\left(\bar{T}\,\Big|_{\tau-\Delta t}^{\tau}\right) \tag{1.10.1.4}$$

where,

$$\bar{T}\,\Big|_{\tau-\Delta t}^{\tau} = \frac{1}{a\Delta t}\left(T_{\tau\text{-}\Delta t} + \frac{\bar{b}}{a}\right)\left(e^{a\Delta t} - 1\right) - \frac{\bar{b}}{a} \tag{1.10.1.5}$$

In general, $\bar{b}$ will be expressed as

$$\bar{b} = b\left(\tau - \frac{\Delta t}{2}, \bar{T}\,\Big|_{\tau-\Delta t}^{\tau}\right) \tag{1.10.1-6}$$

For the case where b is actually a constant over an interval, equation (1.10.1-2) represents an exact solution. In the other extreme, where a = 0 and b is an unknown function of time, then

the derivative is considered a constant.
In this case

$$T_\tau = T_{\tau-\Delta t} + b\left(\tau - \frac{\Delta t}{2}, T_{\tau-\Delta t/2}\right)\Delta t \tag{1.10.1.7}$$

This result is equivalent to the modified Euler method of solving differential equations described in the next section.

When applied, analytical solutions to differential equations are performed each timestep within TRNSYS using subroutine DIFFEQ. A description of the use of DIFFEQ in solving differential equations is given in Section 3.4.4.

1.10.2 NUMERICAL METHODS

It is not always possible to express differential equations in the form of equation (1.10.1.1). In general, they are of the form

$$\frac{dT}{dt} = f(t,T) \tag{1.10.2.1}$$

where $f(t,T)$ is some function of time and/or the dependent variable. The solution to this differential equation at any time, t can be written

$$T_\tau = T_{\tau-\Delta t} + \int_{\tau-\Delta t}^{\tau} f(t,T)dt \tag{1.10.2.2}$$

Various numerical methods have been developed to numerically estimate the value of the above integral. The user is given the choice of three numerical methods: the modified Euler method (1), the original non-self-starting Heun method (2), and a 4th order Adams method (3). By default, the modified Euler method will be used. Users considering changing the default numerical method may wish to consider the following:

Versions of TRNSYS prior to 11.1 utilized the non-self-starting Heun method for solving first-order ordinary differential equations. This is a first-order predictor-correcter algorithm using Euler's method for the predictor step and the trapezoidal rule for the correcting step. The derivative is considered to be constant over each timestep, equal to the average of the derivatives at the beginning and end of the interval.

In order to improve computational efficiency in TRNSYS 12.1, many of the numerical solutions were replaced with approximate analytical solutions (often using the DIFFEQ subroutine). In these cases, the derivatives are considered to vary exponentially over each timestep. To be consistent with the analytical solutions, the non-self-starting Heun method was replaced with a modified Euler method (also known as the improved polygon method). The Euler method is used for the predictor step, while the correcter step uses the derivative at the midpoint of the interval. It is the average values of the dependent variables that are considered to be passed from component to component. In light of this, the integration method used in TYPE 24 (Quantity Integrator) TYPE 27 (Histogram Plotter) and TYPE 28 (Simulation Summary) was changed from triangular to rectangular.

Starting with version 13, the non-self-starting Heun method is again available to users, as is the 4th order Adams method. This latter method uses a 4th order Runge-Kutta method to start, the 4th order Adams-Bashford formula as the predictor and the 4th order Adams-Moulton method as the correcter. Both of these methods pass the dependent variable value at the end of the timestep back to the components. This may cause an inconsistency when used in conjunction with the TYPE 24, TYPE 27, TYPE 28, or TYPE 55 components or with components expecting dependent values to be averaged over the time step.

A description of how to use TRNSYS to numerically solve differential equations is given in Section 3.3.2. Section 2.2.6 describes the method by which the numerical differential equation method can be chosen.

1.10.3 CONVERGENCE AND INTEGRATION

Convergence of the solutions to numerically-solved differential equations is checked by comparing successive guesses of the dependent variables (T_τ). If

$$\frac{2}{\left|T_\tau^{i}\right| + \left|T_\tau^{i-1}\right|} > \varepsilon_D \qquad \text{and} \qquad \frac{2\left|T_\tau^{i} - T_\tau^{i-1}\right|}{\left|T_\tau^{i}\right| + \left|T_\tau^{i-1}\right|} > \varepsilon_D$$

or

$$\left|T_\tau^{i} - T_\tau^{i-1}\right| > \zeta_D$$

where ε_D is a relative (and ζ_D is an absolute) integration error tolerance specified by the user; OR if any of the OUTPUTS of the system components have not converged within the relative error tolerance ε_A (or the absolute error tolerance ζ_A) as discussed in Section 2.2.2, then the iterative procedure continues.

If convergence of either the dependent variables or the OUTPUTS (and hence the INPUTS) has not been obtained within the number of iterations specified on the LIMITS card (see Section 2.2.3), an appropriate WARNING message will be displayed and the simulation will proceed. Should this situation occur more than a specified number of times the simulation will be terminated with an ERROR message.

An important factor in the TRNSYS scheme of solving the simultaneous algebraic and differential equations is that only those component models whose INPUTS (which are OUTPUTS of other components) or dependent variables have not converged within the relative error tolerances ε_A, ε_D (or the absolute error tolerances ζ_A, ζ_D) are called during each iteration. In this manner, TRNSYS requires a minimum of calculation effort to achieve a solution to the equations of the model within the specified error tolerance.

Quantity integration, which is provided by TYPE 24 (Integrator), TYPE 27 (Histogram Plotter), TYPE 28 (Simulation Summary), and TYPE 55 (Periodic Integrator) occurs at each time step after all of the algebraic and differential equations have converged. In TRNSYS versions prior to 11.1, trapezoidal integration was used. This was consistent with the Modified Euler method of solving differential equations. Since the current standard algorithms always set average values over each timestep as outputs, rectangular integration is now employed.

1.11 SUMMARY

In this section, the concepts and definitions presented in the previous sections of this chapter have been organized into five steps which represent the manner in which a system should be prepared for simulation by TRNSYS. These five steps summarize the more important points of this chapter.

1. Identify System Components

Modular simulation of a system requires the identification of components whose collective performance describes the performance of the system. A library of component models is provided as part of the TRNSYS program. These models are organized into functional groups in this manual to facilitate the selection of appropriate models.

The level of detail of a simulation is determined by the components selected to describe the system. For example, a very simple solar water-heating system is described in Section 1.3; a more detailed analysis might include a pipe model to calculate thermal losses from pipes, a pressure relief valve model to calculate thermal losses due to boiling of the collector fluid, etc. Furthermore, the pipe model provided with TRNSYS allows the user to choose whether or not to include the effect of the pipe's thermal capacitance. Thus for each system to be simulated, the user must decide how to represent the system as a set of components, depending on the level of detail required for a given application.

Simulations generally require some components which are not ordinarily considered as parts of the system. Such components described in Section 4.1 (utility subroutines) and Section 4.10 (output-producing devices). Familiarity with these components is highly recommended.

2. Associate TYPE numbers with the System Components

The TYPE number of a component relates the component to a FORTRAN subroutine which models that component; a TYPE n component is modeled by subroutine TYPEn where n is an integer from 1 to 99. Each component model provided in the standard TRNSYS library is identified by a TYPE number. The user providing his or her own FORTRAN subroutines must be sure that they are coded to conform with the requirements imposed by TRNSYS as described in Chapter 3.

3. Construct an Information Flow Diagram

Each component can be represented as a box which requires a number of constant PARAMETERS and time-dependent INPUTS, and produces a set of time-dependent OUTPUTS. In general, each INPUT to a component must be matched to an OUTPUT of another component, a CONSTANT (see Section 2.2.4) or a variable defined by an EQUATION (see Section 2.2.5). In some cases, providing appropriate INPUTS to all components may require the use of components which do not have hardware analogs such as the TYPE 14 Forcing Function or the TYPE 15 Algebraic Operator. Information flow representations of component models and an index of PARAMETERS, INPUTS and OUTPUTS are provided in the reference section of this manual. An information flow diagram shows the manner in which all system components are interconnected. A given OUTPUT may be used as an INPUT to any number of other components, including the component itself. Information flow diagrams may contain loops (recyclic information flow).

4. Assign UNIT Numbers to the System Components

Since a system may contain two or more components with the same TYPE number (e.g. two TYPE 3 pumps), TRNSYS uses a UNIT number to identify each component in the information flow diagram. The unit number of a component is an integer from 1 to n, where n is set in the file param.inc and has a default value of 80, which references that component. It will be seen in Chapter 2 that an INPUT variable of a component is recognized in part by the UNIT number of the component from which that variable is an OUTPUT. Although two or more system components can have the same TYPE number, each must have a unique UNIT number.

5. Identify and Initialize Time-Dependent Variables

TRNSYS distinguishes between the integration required to solve first-order differential equations which may describe the performance of a component, and integration of a quantity over time. If a component model involves numerically solved differential equations, TRNSYS must be informed of the number of differential equations and the initial values of the dependent variables. The methods used by TRNSYS to solve differential equations are described in Section 1.10. Integration of a quantity over time is performed by the TYPE 24 Quantity Integrator, the TYPE 27 Histogram Plotter, the TYPE 28 Simulation Summary Routine, and the TYPE 55 Periodic Integrator.

In addition to the data which must be specified for each of the system components, the user must specify the starting and stopping times of the simulation and the simulation timestep. The manner and format in which all of the data are conveyed to TRNSYS are described in the next chapter.

References

1. Steven C. Chapra and Raymond P. Canale, Numerical Methods for Engineers with Personal Computer Application (New York: McGray-Hill, 1985), p. 488.
2. Steven C. Chapra and Raymond P. Canale, p. 483.
3. Steven C. Chapra and Raymond P. Canale, p. 530.