

2 Basic elements of dynamic simulation

P.A. Leffelaar

2.1 Introduction

When analysing systems one is usually interested in the status of the system at a certain moment and its behaviour as a function of time. A system, defined in Section 1.1 as a limited part of reality that contains interrelated elements, may be too complex to study in its entirety. However, a model, defined as a simplified representation of a system that contains the elements and their relations that are considered to be of major importance for system behaviour, may be easier to study. The design of such models and the study of the model behaviour in relation to that of the system is called simulation; when the change with time is also included, it is called dynamic simulation.

Dynamic simulation models are based on the assumption that the state of each system at any moment can be quantified, and that changes in the state can be described by mathematical equations: rate or differential equations. This leads to models in which state, rate, and driving variables can be distinguished.

The purpose of this chapter is to introduce the method of constructing models according to the state variable approach by very elementary systems in Section 2.2. The concept of feedback and the possibility of visualizing the available knowledge of a system by means of relational diagrams will be discussed in Section 2.3. Section 2.4 shows how the corresponding differential equations may be integrated analytically to obtain the state variables as a function of time in these simple systems. Slight changes in differential equations make analytical solutions impossible, so solutions must be obtained by numerical integration methods. These solutions are based on the assumption that the rate of change is constant during a small period of time, Δt . The principle of numerical integration and the relation between the time interval of integration, Δt , and the time coefficient of a system are discussed in Section 2.5. In Section 2.6 a more complex system is analysed using the methods presented. The chapter is summarized in Section 2.7.

2.2 State variables, rate variables and driving variables

To introduce the method of constructing models according to the state variable approach, the following examples of elementary systems are used (Figure 2.1):

1. a car driving at a constant speed;
2. a number of animals that increases each year by a certain fraction;
3. a tank that is filled by a flow of water through an adjustable valve until a certain water level is reached.

The state variables in these examples are the distance covered by the car, the num-

ber of animals and the amount of water in the tank, respectively. Generally, state variables have dimensions of length, number, volume, weight, energy or heat content.

The ultimate status of a system is not the only feature of interest; we are also concerned with its behaviour in time. Thus, the rate of change of the state variables in time must be known, as well as the direction of change. If these rates have a clear pattern, they may be formalized by means of rate equations or differential equations. The rate equations and their graphical representation for the three elementary systems are given in Figure 2.1. Rate variables, the left hand sides in Equations 2.1, 2.2 and 2.3 in Figure 2.1, have the dimension of a state variable per unit of time, i.e.

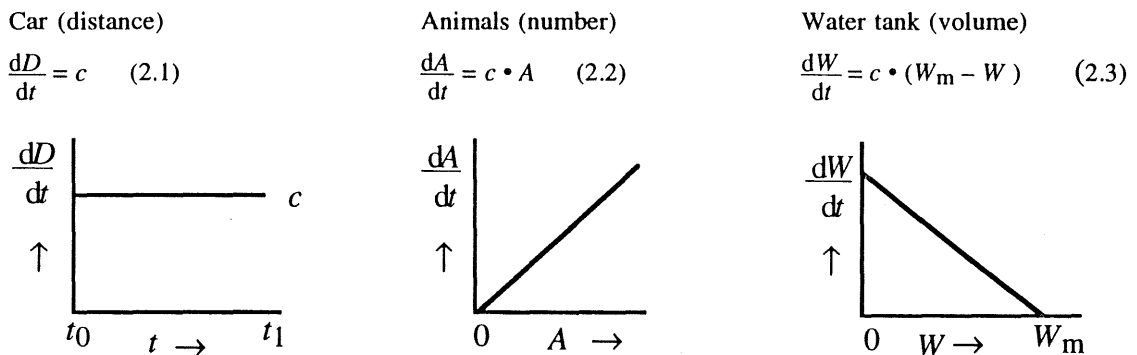


Figure 2.1. Rate or differential equations (Equations 2.1, 2.2 and 2.3) and their graphical representation for three elementary systems. D , A and W stand for the state variables, t for time and c is a constant that may be different in each equation. W_m is the maximum water level that can be reached.

length time^{-1} , number time^{-1} and volume time^{-1} , respectively. In Equations 2.2 and 2.3 the rate variables are functions of state variables, A and W , respectively, whereas in Equation 2.1 the state has no effect on the rate. The influence of a state on its rate of change is called feedback and will be discussed in Section 2.3. The proportionality coefficients, c , in Equations 2.2 and 2.3 are important for the behaviour of the state variables and often have special names. In biological systems c is called the relative growth rate; in technical systems the inverse of c is used, and is called the time coefficient. Time coefficients and their consequences for numerical integration are crucial in dynamic simulation, which is discussed in Section 2.5. The constant c in Equation 2.1 is a driving variable with the dimension speed. Driving variables, or forcing functions, characterize the effect of outside conditions on a system at its limits or boundaries, and their values must be monitored continuously. Driving variables may have the dimension of rate variables, as in Equation 2.1, or state variables, depending on their nature. When the fraction by which the number of animals increases each year depends on temperature, it has the dimension of a state variable. It is imperative to check the dimensions of all variables in any model.

Exercise 2.1

- a. What are the dimensions of c in Equations 2.1, 2.2 and 2.3 in Figure 2.1?
- b. Which general rules form the basis of dimensional analysis?

2.3 Feedback and relational diagrams

The rate variables in Equations 2.2 and 2.3 (Figure 2.1) are functions of the state variables A and W , respectively, whereas the rate variable in Equation 2.1 is independent of the state variable D . When a rate variable, expressed in general terms as the derivative dX/dt , depends on the state variable X , a feedback loop exists, i.e. the state of the variable determines its rate of change, and hence its subsequent state. This feedback takes place continually. Both positive and negative feedback loops exist.

In a negative feedback loop the rate of change may be either positive or negative, but it is a negative function of the value of the state variable. For instance, in the case of the water tank (Equation 2.3) the rate is positive, but it decreases linearly with increasing water level in the tank. An example of a negative rate which decreases as a function of the value of the state variable, is obtained when the sign of the coefficient c in Equation 2.2 is reversed. Then the number of animals decreases each year by a certain fraction, denoting exponential mortality. Negative feedback causes the system to approach some stable equilibrium: if the system is disturbed it returns to its equilibrium state. In the case of the water tank, the equilibrium state is the maximum water level, W_m . In the case of exponential mortality the state variable approaches zero.

In case of a positive feedback loop, there is a positive relation between the values of rate and state, so that both continuously increase. The exponential growth of the animals described by Equation 2.2 is an example of positive feedback. In reality, however, there are limits to growth. Then, the simple Equation 2.2 no longer describes the system and the model needs revision.

Exercise 2.2

Give at least three environmental conditions that should always be satisfied to achieve a situation where the relative growth rate, c in Equation 2.2, is constant.

Relational diagrams are used to visualize feedback loops, rate and state variables and, more generally, the available knowledge about a system. They illustrate the most important elements and relationships of a system and represent qualitative models of systems. Relational diagrams may be especially helpful at the start of research to facilitate the formulation of rate and state variables. They also make the content and characteristics of a model easily accessible. Relational diagrams for the three systems are given in Figure 2.2, following the conventions of Forrester (1961) as shown in Figure 2.3. Figure 2.2 shows that feedback is absent in the case of the car, and that there is positive and negative feedback in the case of the animals and the water tank, respectively. When a parameter turns out to be variable, it must be replaced by a table or an auxiliary equation. For instance, if the coefficient c in Equation 2.2 for the animals, is dependent on temperature, it can be replaced by a so-called auxiliary variable containing information about this temperature dependence which flows to the rate variable.

Relational diagrams of more complex models can often be analysed in terms of the elementary systems of Figure 2.2.

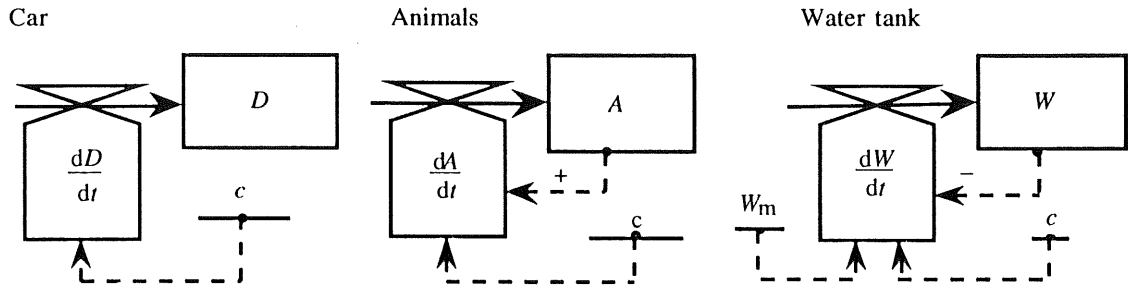
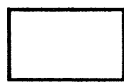


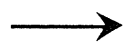
Figure 2.2. Relational diagrams for three elementary systems. Variables and symbols are explained in Figures 2.1 and 2.3, respectively.



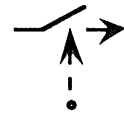
A state variable or integral.



Integral symbol for a boxcar train. Small bars indicate that the boxcar train consists of a series of integrals.



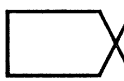
Flow and direction of a substance by which an amount, or state variable, is changed. These flows always begin or end at a state variable, and may connect two state variables.



Flow that is discontinuous: when it is closed, it has some numerical value, otherwise it is zero. This flow is controlled by a flow of information.



Flow and direction of information derived from the state variables of the system. Dotted arrows always point to rate variables, never to state variables. The use of information does not affect the information source itself. Information may be delayed and as such be part of a process itself.



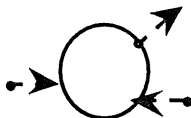
Valve in a flow, indicating that the calculation of a rate variable takes place; the lines of incoming information indicate the factors upon which the rate depends.



Source or sink of quantities in whose content one is not interested. This symbol is often omitted.



A constant or parameter.



Auxiliary or intermediate variable in the flow of information.

+ , -

Sometimes placed next to a flow of information, to indicate whether a loop involves a positive or negative feedback.

Figure 2.3. Basic elements of relational diagrams. Acronyms of variables represented by these elements are usually written inside the symbols. Note that driving variables are often underlined or placed between parentheses. Intermediate variables are often represented by circles.

2.4 Analytical integration and system behaviour in time

Differential equations summarize existing knowledge of a system, i.e. they relate rate variables to state variables, driving variables and parameters. They thus form a model of that system. When the differential equations are formulated, and if the state of the model at a certain moment is known, its state in the future can be calculated. For this purpose the differential equation must be solved with respect to time. This process is called integration and can be visualized for the simplest case of Equation 2.1 by determining the distance covered by the car in a certain period. Here, the speed is multiplied by the length of the period. Thus the increase in the value of the state variable in a period $t_1 - t_0$ time units (Figure 2.1), equals the area delimited by the time axis, the line parallel to this time axis at the value c on the rate axis and the two lines, parallel to the rate axis, at the moments t_0 and t_1 . For Equations 2.2 and 2.3 the situation is different, as the rate variables depend on the state. Then, the formal process to obtain the state variable as a function of time must be applied, i.e. integration of the rate equation. This is shown in Figure 2.4 for all three models. Integration of Equation 2.2 produces the well-known exponential growth curve (Equation 2.5 in Figure 2.4). The relationship between the rate variable, dA/dt , and time is obtained by differentiating Equation 2.5 with respect to time. This yields Equation 2.2a, which has the same form as Equation 2.5. The graph picturing Equation 2.2a may be used to obtain the state variable. This may seem trivial, since the analytical solution is already available (Equation 2.5). The graph may, however, be used to illustrate the errors introduced by numerical integration methods when these are used to solve differential equations. This is discussed in Section 6.4.

In the case of the water tank, the rate of inflow of water decreases linearly with the difference between a maximum water level, W_m , and the actual water level, W (Equation 2.3). Integration yields Equation 2.6, expressing that the water level in the tank approaches W_m exponentially. Equation 2.3a, obtained by differentiating Equation 2.6 with respect to time, shows that the rate of inflow decreases exponentially.

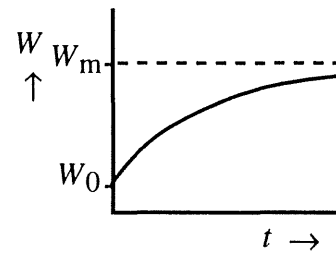
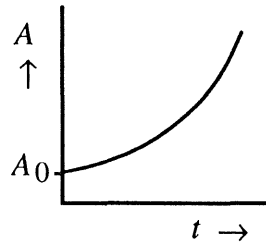
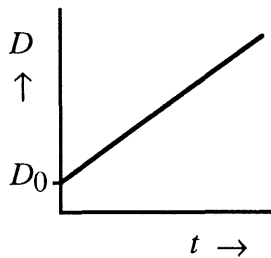
Exercise 2.3

Consider the graphs representing Equations 2.4, 2.5 and 2.6 in Figure 2.4.

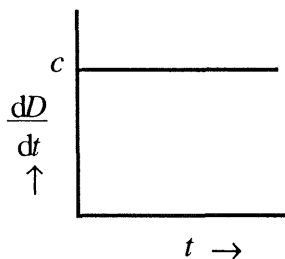
- What do the slopes of the different curves represent?
- What is the dimension of the slope in each case?
- How do the numerical values of the slopes change with time?
- Does your answer in c) agree with the graphical presentation of Equations 2.1, 2.2a and 2.3a?

As long as differential equations are simple, they may be solved analytically to study the behaviour of the models. Minor changes in the equations, however, e.g. if c in Equation 2.2 is a function of temperature, make analytical solutions impossible. This is also the case when model results deviate from the behaviour of the system, and more complex (sub-)models are needed, based on newly gathered knowledge of the system. Then, the resulting set of differential equations should be solved by numerical integration methods.

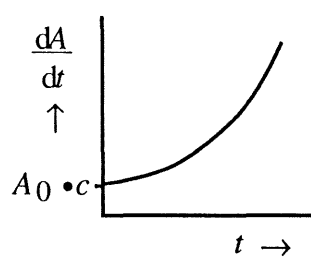
Car	Animals	Water tank
$\frac{dD}{dt} = c$ (2.1)	$\frac{dA}{dt} = c \cdot A$ (2.2)	$\frac{dW}{dt} = c \cdot (W_m - W)$ (2.3)
$\int dD = c \cdot \int dt$	$\int \frac{dA}{A} = c \cdot \int dt$	$\int \frac{d(W - W_m)}{W - W_m} = -c \cdot \int dt$
$D = c \cdot t + Q$	$\ln A = c \cdot t + Q$	$\ln(W - W_m) = -c \cdot t + Q$
initial value of the state variable at $t = 0$:		
$D = D_0,$ so: $Q = D_0$	$A = A_0,$ so: $Q = \ln A_0$	$W = W_0,$ so: $Q = \ln(W_0 - W_m)$
$\ln \frac{A}{A_0} = c \cdot t$	$\ln \left(\frac{W - W_m}{W_0 - W_m} \right) = -c \cdot t$	
$D = c \cdot t + D_0$ (2.4)	$A = A_0 \cdot e^{c \cdot t}$ (2.5)	$W = W_m - (W_m - W_0) \cdot e^{-c \cdot t}$ (2.6)



$$\frac{dD}{dt} = c \quad (2.1)$$



$$\frac{dA}{dt} = A_0 \cdot c \cdot e^{c \cdot t} \quad (2.2a)$$



$$\frac{dW}{dt} = (W_m - W_0) \cdot c \cdot e^{-c \cdot t} \quad (2.3a)$$

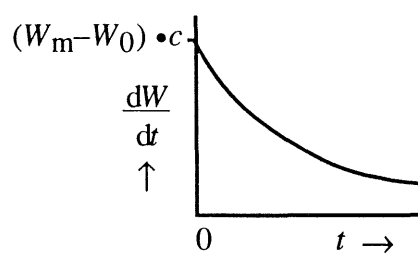


Figure 2.4. Upper half: analytical solutions (Equations 2.4, 2.5 and 2.6) to the differential Equations 2.1, 2.2 and 2.3 respectively, and their graphs.

Lower half: rate variables (Equations 2.1, 2.2a and 2.3a) as a function of time, derived from Equations 2.4, 2.5 and 2.6 respectively, and their graphs. For an explanation of variables see Figure 2.1. Q stands for a general integration constant, and D_0 , A_0 and W_0 are the initial values of the state variables in the particular models.

2.5 Numerical integration and the time coefficient

In numerical integration the assumption is made that the rate of change of a state variable is constant over a short period of time, Δt . To calculate the state of a system after that short period one must know the state of the system at time t , $state_t$, and the value of the rate variable at that time, $rate_t$, calculated from the differential equation. By multiplying the $rate_t$ by Δt and adding this product to the value of the state variable according to

$$state_{t+\Delta t} = state_t + \Delta t \cdot rate_t \quad (2.7)$$

the new state, $state_{t+\Delta t}$, of the system is determined. From this new state, a new rate is calculated which holds for the next time interval Δt , and so on. This scheme is the simplest, so-called rectangular or Euler's integration method. Euler's method will be used to illustrate what is meant by the qualification 'short period', Δt , especially with respect to its relation to the time coefficient of a particular model (see below), and with respect to errors introduced by numerical integration. In part B more elaborate methods will be introduced (Section 6.2).

Numerical integration is first applied to the example of the water tank, Equation 2.3. Assume that the tank is empty at $t = 0$ s, so $W_0 = 0$ l, coefficient c equals $1/4 \text{ s}^{-1}$, and the maximum volume $W_m = 16$ l. The rate of inflow of water into the tank at $t = 0$ is calculated from the rate equation

$$(dW/dt)_t = c \cdot (W_m - W_t) \quad (2.8)$$

and is 4 l s^{-1} . If the time interval Δt , or the 'short period' equals 2 s, the volume of water at time $t + \Delta t$ is obtained from the state equation

$$W_{t+\Delta t} = W_t + \Delta t \cdot (dW/dt)_t \quad (2.9)$$

as 8 l. During the next time interval of 2 s, the rate is: $1/4 \cdot (16 - 8) = 2 \text{ l s}^{-1}$. Thus, during this time interval 4 l of water will flow into the tank and the total quantity of water after 4 s equals $8 + 4 = 12$ l. The calculations thus proceed according to Equations 2.8 and 2.9, and can be represented by the following diagram:

time	s	0	← Δt	→ 2	4	6	8	10
W	l	0	↓	8				
dW/dt	l s^{-1}	4	→	2				

Exercise 2.4

Complete the numerical calculation and plot the amount of water in the tank against time. Now also calculate analytically the amount of water in the tank using Equation

2.6, and plot the results in the same graph.

- What is the difference between the numerical and the analytical solution and what is the reason of this difference?
- When is the rate of inflow zero?
- What happens if coefficient c is $1/8$ instead of $1/4$?

In the case of the water tank the rate of filling decreases with time (see Figure 2.4, Equation 2.3a), so that the numerical integration, where the rate variable is kept constant during the time interval Δt , overestimates the amount of water in the tank compared to the analytical solution (Exercise 2.4). The difference between the value of the state variable obtained by the numerical method and the analytical value will be smaller when Δt is smaller. The lower limit to Δt is set by the technical possibilities (rounding errors) of performing the calculations over long time spans.

Exercise 2.5

The parameter values in Equations 2.8 and 2.9 are: $W_0 = 0$ l; $W_m = 16$ l; $c = 1/4$ s⁻¹.

- Perform numerical integrations for the filling of the water tank up to about 30 s using the following time intervals: $\Delta t = 1 \cdot c^{-1}$; $\Delta t = 1.5 \cdot c^{-1}$; $\Delta t = 2 \cdot c^{-1}$; $\Delta t = 2.5 \cdot c^{-1}$.
- Plot the results in the graph of Exercise 2.4.
- What is the relation between the results of the calculations and the ratio of the time interval and the value of c^{-1} ?
- What upper limit would you set to this ratio? (Also consider your calculations of Exercise 2.4).

The upper limit to Δt is determined by the inverse value of coefficient c in the differential equation. The inverse of c is called the time coefficient, τ , which has the dimension of time. It is a measure of the reaction rate of a model. In models containing more than one rate variable a first approximation to the time interval is obtained by taking Δt smaller than one tenth of the smallest τ in that model. The time coefficient appears equal to the time that would be needed for the model to reach its equilibrium state, if the rate of change were fixed. This applies to any point on the integrated function, as shown in Figure 2.5.

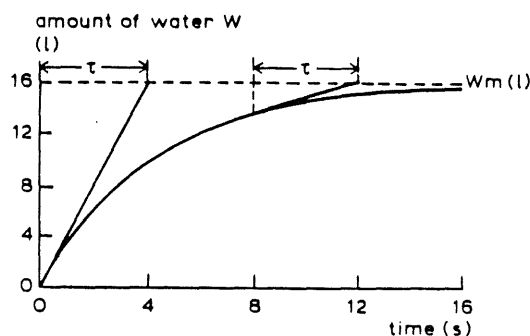


Figure 2.5. The amount of water as a function of time, according to Equation 2.6, with $W_0 = 0$ l, $W_m = 16$ l and $c^{-1} = \tau = 4$ s: $W = W_m \cdot (1 - e^{-t/\tau})$. The time interval over which the tangent must be extended to intersect with the equilibrium line is the time coefficient, τ .

Exercise 2.6

- Check this last statement by applying Equation 2.8.
- The statement also holds for a positive feedback loop, but the formulation is slightly different. Explain this for the case of exponential growth.

Biological models are often characterized by a relative growth rate (rgr), equivalent to the coefficient c in Equation 2.2, with dimension time^{-1} . This name 'relative growth rate' is explained by writing c explicitly as $(dA/dt)/A$.

Exercise 2.7

- Calculate the time coefficients when the relative growth rates are 1.5, 0.2 and 0.05 per year, respectively.
- What time intervals would you use for numerical integration in these cases? Also take into account the practical aspect of numerical calculations.
- Compute the number of animals after 5 years, when $c = 0.2$ and $A_0 = 100$, by
 - the analytical solution to the problem, Equation 2.5;
 - the numerical solution to the problem using Equation 2.2 and $\Delta t = c^{-1}/10$.
- Make a plot of the results.
- Explain the difference between the numerical solution and the analytical solution.

A cautionary remark may be appropriate here. The time coefficient is calculated as the inverse of the relative growth rate (see Exercise 2.7). The growth percentage over a fixed period (i.e. the relative increase in the number of animals after one year or annual relative increase, ari) is often used to calculate τ , but that gives incorrect results. A growth rate of, for example, 20 % per whole year is not equivalent to an rgr of 0.2 yr^{-1} . The relative growth rate is less: when $A_0 = 100$, A equals 120 after one year and Equation 2.5 can be used to calculate the relative growth rate as follows: $A=120=100 \cdot e^{rgr \cdot 1}$, so $rgr = \ln 1.2 = 0.182 \text{ yr}^{-1}$, and $\tau = 5.48 \text{ yr}$ instead of $1/0.2 = 5 \text{ yr}$. The relative growth rate (rgr) may be expressed in the annual relative increase as follows: $A_0 + A_0 \cdot ari \cdot 1 = A_0 \cdot e^{rgr \cdot 1}$ or $rgr \cdot 1 = \ln(1+ari \cdot 1)$, or more generally $rgr \cdot \Delta t = \ln(1+ari \cdot \Delta t)$. For an exponential decline, one can derive $rdr \cdot \Delta t = -\ln(1-ard \cdot \Delta t)$, with rdr and ard the relative death rate and the annual relative decrease, respectively. The difference between ari and rgr , or that between ard and rdr , will be substantial when the annual relative increase or decrease is high.

Other names for the time coefficient and related concepts are time constant, transmission time (in control-system theory), average residence time, delay time, extinction time and relaxation time. This indicates its significance in various disciplines. Doubling time, the time needed to double an amount, is also used to characterize a system, but it is not synonymous with the time coefficient.

Exercise 2.8

The relationship between doubling time, $t(2)$, and the time coefficient in exponential growth is $t(2) \approx 0.7 \cdot \tau$. Derive this relationship.

The relaxation time, a term often used in physics, is the time needed in exponential processes to change the state by a factor e ; it is equivalent to the time coefficient.

For an illustration of average residence time, consider an exponentially decreasing

population of animals (A) without birth or migration, which live in an isolated habitat. The animals thus only die in the course of time. Their rate of decrease (dA/dt) is taken as proportional to their number. The date that these animals entered the habitat is defined as time zero. Figure 2.6 gives the relational diagram that applies to this situation.

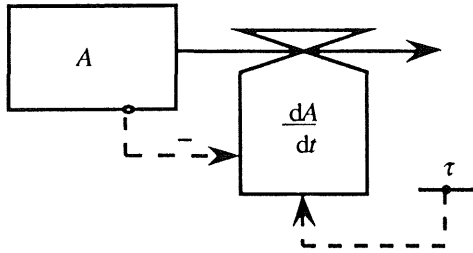


Figure 2.6. Relational diagram for exponential decay or death.

Now we wish to know the average time period that the animals reside in the habitat.

To calculate this average residence time (ART, dimension: time), the residence times of all animals must be summed and divided by the initial number of animals A_0 . This summation process is illustrated in Figure 2.7, where each ΔA may be considered as an individual animal.

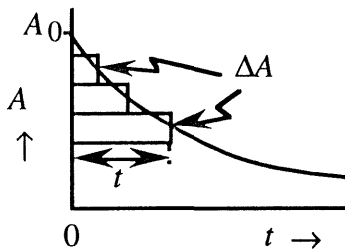


Figure 2.7. Number of animals (A) as a function of time. The surface area under the curve may be found by summation of the rectangles $t \cdot \Delta A$.

By transition from summation over individual animals to continuous integration, the average residence time can be calculated from:

$$\text{ART} = \frac{1}{A_0} \int_0^{A_0} t \cdot dA \quad (2.10)$$

Generally, the $A-t$ curve is not known. However, if the outflow rate of the animals from the area (dA/dt , here in fact the death rate) is *measured*, the curve can be calculated. To do so, dA in Equation 2.10 may be written as the product of the outflow rate and the time step:

$$dA = \frac{dA}{dt} \cdot dt \quad (2.11)$$

and A_0 may be obtained by integrating the outflow rate over time:

$$A_0 = \int_0^{\infty} \frac{dA}{dt} \cdot dt \quad (2.12)$$

Substituting these expressions for dA and A_0 in Equation 2.10 gives:

$$\text{ART} = \frac{\int_0^{\infty} t \cdot \frac{dA}{dt} \cdot dt}{\int_0^{\infty} \frac{dA}{dt} \cdot dt} \quad (2.13)$$

This general definition of ART can be stated in words as follows: the average residence time equals the sum or integral (\int) of the outflow rate at each moment (dA/dt), weighted by the actual time (t), and standardized at a unit number of animals originally present in the area at time zero (A_0 , Equation 2.12). So, the large number of animals that reside a short time in the area contribute much less to the average residence time than do the last animals in the habitat, that have resided a very long period.

In the special case that the initial number of animals is known and that the outflow of the animals from the area is proportional to their number, the curve in Figure 2.7 would be given by $A = A_0 \cdot e^{-t/\tau}$. Furthermore, the surface area under the $A-t$ curve may also be calculated as indicated in Figure 2.8:

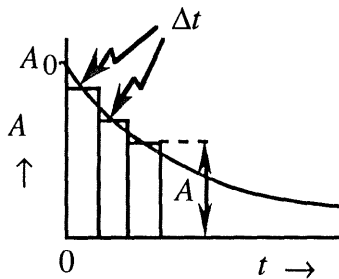


Figure 2.8. Number of animals (A) as a function of time as in Figure 2.7. The surface area under the curve may be found by summation of the rectangles $A \cdot \Delta t$.

Then, the average residence time equals the time coefficient τ .

Exercise 2.9

Check this last statement mathematically by using the definition of the average residence time:

$$\text{ART} = \frac{1}{A_0} \int_0^{\infty} t \cdot \frac{dA}{dt} \cdot dt = \frac{1}{A_0} \int_0^{A_0} t \cdot dA = \frac{1}{A_0} \int_0^{\infty} A \cdot dt$$

and the analytical equation describing exponential decrease: $A = A_0 \cdot e^{-t/\tau}$.

Equation 2.13 for the average residence time is implemented in the boxcar train simulation programs in Chapter 8.

In nature many processes occur simultaneously. In simulation models of such pro-

cesses, however, calculations take place sequentially. But since dynamic simulation is based on the principle that rates of change are mutually independent (i.e. they depend individually on state variables and driving variables), all rates at any one moment can be calculated in series. These rates can then be integrated (sequentially) to obtain the values of the state variables one time interval (Δt) later. In this way the model operates in a semi-parallel fashion, and it simulates simultaneously occurring processes. It is convenient to use special simulation languages to describe parallel processes in a semi-parallel fashion (see Chapter 3). If other computer languages are used, however, this requirement should still be met (see Chapter 10).

2.6 An example

The different steps that may be distinguished in systems analysis of living systems are demonstrated now. A more detailed list of steps is presented in Section 4.5.

Objectives and definition of the system Microbiologists plan to develop a technical system in which yeast can be grown continuously. They wish to use a vessel of constant volume, through which a sugar solution will flow. To gain insight into the proper technical system parameters, such as the volume of the vessel (v , m^3), the required concentration of sugar (c_s , $kg\ kg^{-1}$) and the flow rate of water (q , $m^3\ d^{-1}$), they decide to design a model of the system.

The physiological parameters pertaining to the yeast cannot be adjusted like the technical parameters. Therefore, some experiments are performed which show that the absolute growth rate of the yeast (dy/dt , $kg\ d^{-1}$) is proportional to the amount of yeast present (y , kg), and to the sugar concentration. At a sugar concentration of 10%, c_{s10} , the relative growth rate and the amount of sugar in the vessel are termed μ_{10} and s_{10} , respectively. The rate of sugar consumption per unit yeast (s_y , $kg\ kg^{-1}\ d^{-1}$) is known. The maximum possible quantity of sugar (s_m , kg) in the vessel is determined by the incoming sugar concentration and the volume of the vessel.

Exercise 2.10

The following table gives fictitious data on the amount of yeast as a function of time at different constant sugar concentrations.

sugar concentration in water (c_s , $kg\ kg^{-1}$)	time (h)			
	0	2	4	10
	yeast (kg)			
0	2000	2000	2000	2000
0.02	1950	2119	2304	2958
0.05	1900	2340	2882	5384
0.10	2050	3110	4717	16464

- Derive the relative growth rate of yeast, μ , at these four different sugar concentrations.
- Plot the relative growth rate, in units of day^{-1} , against the sugar concentration, c_s . Express μ in terms of c_s , c_{s10} and μ_{10} .
- Rewrite the expression for μ in terms of the current amount of sugar, s , and s_{10} .

The relational diagram Figure 2.9 shows the relational diagram of the model. Note that this figure is constructed from the elementary systems introduced in Section 2.3. For instance, the lower left rate of inflow with the integral of the sugar (s , kg) is equivalent to the relational diagram of the car from Figure 2.2, and the upper integral of the yeast with the right rate of outflow represents an exponential decrease. The representation of the model by one integral for y and one for s implies that the

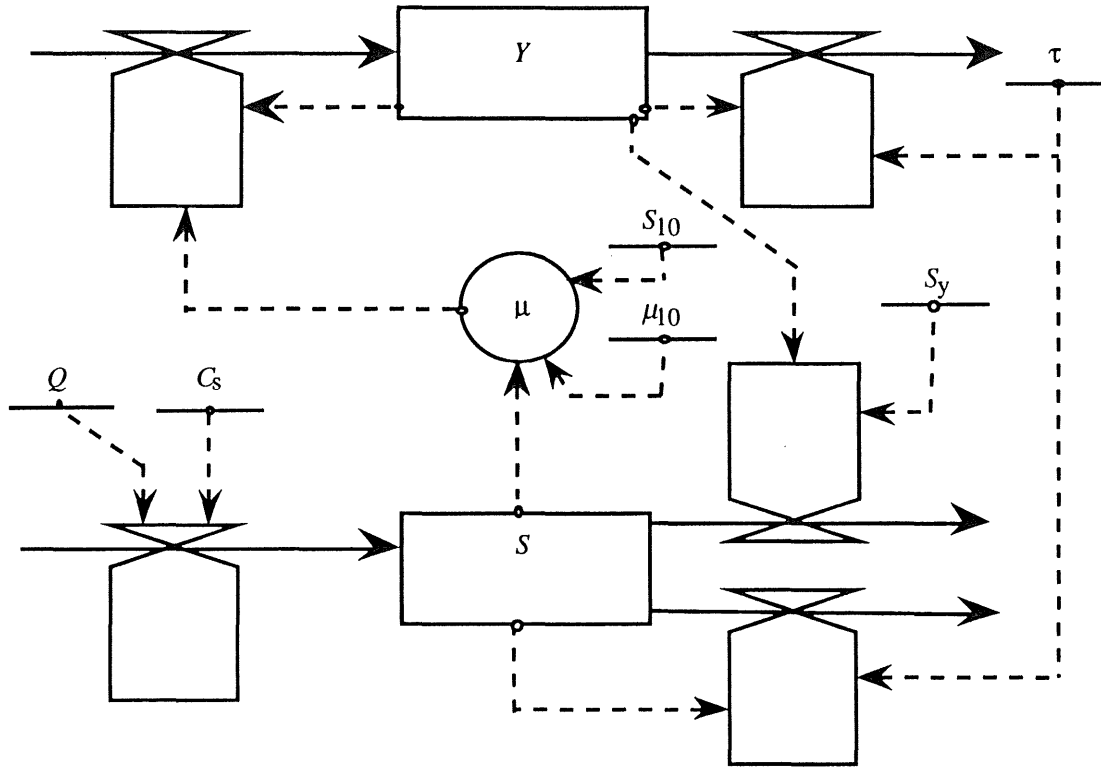


Figure 2.9. Relational diagram of a continuous yeast culture fed by a sugar solution.

yeast and the sugar solution are well mixed throughout the vessel. The relational diagram does not contain yeast mortality, because only the time coefficient of the vessel (τ , d) influences the rate of outflow of yeast. The τ of the vessel has a similar influence on the rate of outflow of the sugar, but here a second rate plays a role, as sugar is consumed by the yeast. The time coefficient of the vessel is defined as the residence time of yeast and sugar in the vessel, when no sugar consumption takes place. In the real system, this characteristic time can be adjusted, as it is defined as $\tau = v/q$ (d). Of course, normally sugar consumption does take place. As a consequence, the residence time of the sugar in the vessel will be shorter. The time coefficient was described as the time that would be needed for the model to reach its equilibrium state, if the rate of change were fixed. This description can be restated as (time coefficient) • (|rate of change|) = |(equilibrium state - current state)|, or

$$\text{time coefficient} = \left| \frac{\text{equilibrium state} - \text{current state}}{\text{rate of change}} \right| \quad (2.14)$$

where 'rate of change' is fixed. The denominator either refers to the rate of change due to inflow or due to outflow. The numerator in Equation 2.14 represents the maximum amount of material that may be added to or removed from the state variable. If we first consider inflow, the 'equilibrium state' is an upper level, as in the example of the water tank. If outflow is considered, the 'equilibrium state' is a minimum level or even zero, as in exponential decay. An equilibrium is eventually reached if the system contains a negative feedback loop. Without feedback, however, numerical solutions will go beyond the boundaries of the real system, and the concept of time coefficient can not be applied. A rate without feedback often characterises the influence of environmental conditions on the system and may be a (measured) time series. Then, we could require that at the beginning of each new integration interval the rate should not be changed more than 5% from the preceding rate.

Differential or rate equations The relational diagram in Figure 2.9 can help to derive the differential equations. It is immediately clear which variables will appear in a particular rate or flow. For example, for the inflow of yeast: $dy/dt = f(y, \mu)$, with $\mu = f(\mu_{10}, s_{10}, s)$; and for the outflow: $dy/dt = f(y, \tau)$.

This information, combined with information on the proportionalities and the units of variables, yields the net flow rate for yeast:

$$\frac{dy}{dt} = \mu \cdot y - \frac{y}{\tau} \quad (2.15)$$

where

$$\mu = \frac{s}{s_{10}} \cdot \mu_{10} \quad (2.16)$$

$$s_{10} = c_{s10} \cdot v \cdot 1000 \quad (2.17)$$

$$\tau = \frac{v}{q} \quad (2.18)$$

By analogy, the net flow rate for sugar is

$$\frac{ds}{dt} = c_s \cdot q \cdot 1000 - y \cdot s_y - \frac{s}{\tau} \quad (2.19)$$

Exercise 2.11

- Examine the units of all variables and constants in Equations 2.15 to 2.19.
- What does the number 1000 in Equations 2.17 and 2.19 represent?
- Use Equation 2.14 to formulate the time coefficient for the sugar in the vessel.

Further analysis of Equations 2.15 to 2.19 To study the dynamic behaviour of the yeast-sugar model, Equations 2.15 and 2.19 must be solved by numerical integration. However, a number of model properties can be analysed by studying simplified equations or equilibrium properties.

An example of simplification of equations relates to the time needed to equilibrate a water-filled vessel, which is initially free of sugar, with the sugar solution, in the absence of yeast. The relational diagram for this problem is represented by the lower

half of Figure 2.9 when the outflow of sugar due to consumption by the yeast is omitted. The differential equation for the sugar, when $y=0$, is $ds/dt = c_s \cdot q \cdot 1000 - s/\tau$, which can be solved analytically. For the condition that at $t=0$, $s=0$, this gives:

$$s = s_m \cdot (1 - e^{-t/\tau}) \quad (2.20)$$

where $s_m = c_s \cdot v \cdot 1000$, which is the maximum amount of sugar that can be achieved with given c_s and v .

Equation 2.20 is similar in form to Equation 2.6, when $W_0 = 0$, although their differential equations describe quite different systems and express a dynamic and a static flow model, respectively.

Exercise 2.12

Derive from Equation 2.20, in general terms, the time needed to reach 95 % of the final equilibrium level of sugar in the vessel.

Such equilibrating processes may take a long time when large time coefficients are involved. It is preferable, therefore, to start an experiment by filling an empty vessel with the desired sugar solution.

Exercise 2.13

How much time, expressed in terms of the time coefficient, is needed to reach 100 % of the equilibrium sugar level when an empty vessel is filled at a constant rate with the sugar solution? There is no outflow until the vessel is full.

Equations 2.15 and 2.19 can be analysed to determine whether equilibrium levels of yeast and sugar can be reached, and if so, what these levels are. In a dynamic equilibrium, the values of the state variables are constant, and the sum of the rates of inflow is equal to the sum of the rates of outflow. Thus, the net rate of change of the state variable equals zero. In the case of the continuous culture this means that dy/dt and ds/dt in Equations 2.15 and 2.19, respectively, are equal to zero. The equilibrium levels of sugar and yeast can be calculated from:

$$s = \frac{s_{10}}{\mu_{10} \cdot \tau} \quad (2.21)$$

and

$$y = \frac{1}{\tau \cdot s_y} \cdot (s_m - s) \quad (2.22)$$

A special case of dynamic equilibrium is obtained when y is zero. Such a dynamic equilibrium is established when the yeast culture is washed out because the time coefficient for the vessel is smaller than that for the yeast. This is the case when $q > c_s \cdot v \cdot \mu_{10}/c_{s10}$.

Exercise 2.14

Derive Equations 2.21 and 2.22 from Equations 2.15 to 2.19.

Equation 2.22 shows that the equilibrium level of yeast depends on the time coefficient of the vessel that can be influenced.

The microbiologists will be interested in the combination of manipulable parameters that yields the maximum yeast production with the minimum amount of sugar. The only manipulable variables in Equation 2.22 are s_m and τ . From Equation 2.21 it follows that s is a hyperbolic function of τ , indicating that at very low τ values s will be very large and at high τ values s will be small. So, there is no practical minimum value for the amount of sugar. To investigate whether a maximum exists in the relation of yeast production against the time coefficient, Equation 2.21 is inserted into Equation 2.22, which is then differentiated with respect to τ to obtain

$$\frac{dy}{d\tau} = \frac{1}{\tau^2 \cdot s_y} \cdot \left(\frac{2 \cdot s_{10}}{\tau \cdot \mu_{10}} - s_m \right) \quad (2.23)$$

Exercise 2.15

- Derive Equation 2.23.
- At what value of τ is there a maximum or minimum value of y ?

If the second derivative of y with respect to τ is negative at the τ value found in Exercise 2.15b, y is at a maximum. The second derivative is

$$\frac{d}{d\tau} \left(\frac{dy}{d\tau} \right) = \frac{d^2y}{d\tau^2} = \frac{2}{\tau^3 \cdot s_y} \cdot \left(s_m - \frac{3 \cdot s_{10}}{\tau \cdot \mu_{10}} \right) \quad (2.24)$$

Substitution of the answer from Exercise 2.15b into Equation 2.24 does yield a negative value for the second derivative: hence, y is at its maximum. The maximum yeast level, at that value of τ , can be calculated from Equation 2.21 and 2.22 as

$$y = \frac{1}{4} \cdot \frac{\mu_{10}}{s_{10} \cdot s_y} \cdot s_m^2 \quad (2.25)$$

The amount of sugar is then

$$s = 1/2 \cdot s_m \quad (2.26)$$

Equations 2.25 and 2.26 show that at the optimum value of τ ($= v/q$), the amounts of yeast and sugar can still be changed by adjusting the inflow concentration of sugar which determines s_m .

Exercise 2.16

- Check the units of Equations 2.21 to 2.26.
- Express the water flux q in terms of the other parameters to calculate the inflow

rate resulting in the maximum amount of yeast at a given sugar concentration and volume of the vessel.

2.7 Summary

The terms system, model and simulation were illustrated by introducing three basic elementary systems and their models (car without feedback, exponential growth with positive feedback resulting in an unstable equilibrium, and a water tank with a negative feedback mechanism resulting in a stable equilibrium). The important assumption was that systems and their models are state determined, which means that if the initial conditions and the mathematical description of the rates of change of these state variables are available, the time course of the model can be calculated. In these models state, rate, and driving variables are distinguished. It was shown that the systems could be qualitatively represented by means of relational diagrams where the major state, rate and driving variables and their interrelationships and feedbacks are given.

Analytical integration can sometimes be used to obtain the desired time course of models. These solutions are exact because the time increment dt is infinitely small. However, analytical solutions are restricted to rather simple equations or at least to simple boundary conditions and one needs much skill to apply analytical methods.

Numerical integration can always be used to solve differential equations, however. In this case the time increment of integration, Δt , is finite and the ratio between Δt and the characteristic time in which the system significantly changes, i.e. the time coefficient (τ) of the model, should be chosen so that the solution obtained is sufficiently close to the analytical solution, if it would exist. As a rule of thumb the ratio of $\Delta t/\tau$ is taken as 1/10. Numerical integration methods provide a powerful tool to solve differential equations, also to those who have no strong background in mathematics, provided that one comprehends the calculation procedure given in Equation 2.7 and the important remark in Section 2.5 about mutual independency of rates and semi-parallel integration of simultaneously occurring processes.

An example about yeast growth was used to illustrate a number of aspects introduced in earlier sections. Furthermore, an analysis of a set of differential equations in terms of equilibrium conditions was given. In this context static and dynamic equilibrium were mentioned. Dimensional analysis was found to be very important in developing and understanding equations.

Though the presented techniques may seem very simple, they are extremely powerful in model building of ecosystems. Most models given in the following chapters are composed of the elementary feedback loops, while the above simple mathematical techniques suffice to solve them. Analysis of and solutions to more complex problems require especially more knowledge about the relationships that may characterize system structure, rather than sophistication of mathematical techniques. Such knowledge forms at present the major restriction to systems analysis and simulation of ecological processes, but it forms also the major challenge of future research.

3 A simulation language: Continuous System Modeling Program III

P.A. Leffelaar

3.1 Introduction

CSMP stands for Continuous System Modeling Program, version III. It has been extensively described in the Program Reference Manual by IBM Corporation, 1975. In this chapter only the essentials of CSMP will be discussed. However, these essentials enable us to solve many problems.

CSMP is a non-procedural language, which means that the user can write programs in a conceptual order, and that CSMP will sort the statements in a computational order. This implies on the one hand that CSMP takes care of the principle of systems analysis that the rates of change are calculated when the states are known and thus that parallel processes are described in a semi-parallel fashion (see Section 2.5), and on the other hand that the user is allowed to start thinking and programming at a high level of integration while descending to the details later on.

CSMP is a dynamic simulation language primarily designed to integrate rate equations to obtain the states of the model as a function of time. CSMP takes care of the integration scheme and keeps track of the independent variable time. CSMP also provides a number of numerical integration routines which are easy to use. In this chapter and in part A of this book, however, only the rectangular method will be dealt with. This approach helps the student to first gain a thorough feeling of the important aspects of the relation between the time step in numerical integration and the different characteristic times or time coefficients in the model.

Furthermore, CSMP provides special functions (e.g. interpolation), data input is simple, and tabular- and/or graphical output can be obtained by just listing the variables on a special label.

As the source program of CSMP is written in FORTRAN (FORMula TRANslation), FORTRAN statements as well as all FORTRAN library functions and libraries written in FORTRAN (e.g. IMSL, 1987; Press et al., 1986) can be used in more advanced models. The possibility of using FORTRAN in combination with CSMP is a major reason to use CSMP as a simulation language. The powerful aspects of using FORTRAN within CSMP models will be postponed to Chapter 7. In Chapter 10 simulation in FORTRAN only is dealt with.

The general design of a CSMP program is given in Listing 3.1. It will be used in the following sections to illustrate the structure of CSMP programs, and to indicate where to place the state, rate and auxiliary equations, the data for input and output, the method of integration and the timer. It is also used to show how reruns may be invoked.

Listing 3.1. General design of a CSMP program. Bold characters are used to highlight the 3 main program segments that may be distinguished, and the beginning and termination of the program; italic characters preceded by an asterisk are used for comments. Regular programs do contain one character type only. Note that an equal-sign means 'is to be replaced by' rather than 'is equal to'. Numbers before program lines are used for reference in the text: they do not form part of CSMP programs.

```

01  TITLE General design of a CSMP program
02  INITIAL
03  FIXED      .....
04  * Data input:
05  INCON      STAT10 =...      , STAT20 =...
06  CONST      PI      =3.1416 , ..... =...
07  PARAM      ..... =...      , ..... =...
08  FUNCTION   DRV1TB = (...,...) , (...,...) , ...
09              (...,...) , (...,...)
10  * Data output:
11  PRINT      STATE1 , STATE2 , NRATE1 , NRATE2
12  OUTPUT     STATE1 , STATE2 , NRATE1 , NRATE2
13  PAGE       NTAB=0 , GROUP=2 , WIDTH=80
14  * The independent variable time:
15  TIMER      TIME =... , DELT =... , PRDEL=... , ...
16              OUTDEL=... , FINTIM=...
17  * The integration method:
18  METHOD      RECT
19  * Initial calculations:
20  DYNAMIC
21  * State variables:
22  STATE1     =INTGRL(STAT10 , NRATE1)
23  STATE2     =INTGRL(STAT20 , NRATE2)
24  * Driving and rate variables:
25  DRV1       =AFGEN(DRV1TB,TIME)
26  NRATE1     =f(STATE1 , DRV1 , TAU...)
27  NRATE2     =f(STATE1 , STATE2 , TAU...)
28  VAR2       =AFGEN(VAR2TB,.....)
29  TERMINAL
30  * Final calculations:
31  * Reruns may be defined between 'END' and 'STOP':
32  END
33  INCON      .... , ...
34  PARAM      .... , ...
35  FUNCTION   DRV1TB =
36  TIMER      .... , ...
37  END
38  STOP
39  ENDJOB

```

3.2 The structure of the model

Listing 3.1 shows that a program should begin with a TITLE label containing a short identification of the program. The first letter of the title should be in upper case. In a CSMP program 3 segments can be distinguished: INITIAL, DYNAMIC and TERMINAL. These statements (labels) indicate that the computations must be performed before, during and after a simulation run, respectively. If one is using these segments, then each segment label closes the former segment. The entire program is terminated by the statements END, STOP and ENDJOB, each on a separate line. END completes the specifications of the model, STOP terminates the simulation run, and ENDJOB terminates the job. The label ENDJOB must start in the first column.

In the INITIAL segment computations are executed only once per run. The use of the segment is optional. The INITIAL segment can be used to give values to the input data and to the time variables, and to define data output and the integration method of the model. Furthermore, the computation of results which are used as input for the dynamic section of the program may be executed here. In the DYNAMIC segment the time course of the state variables is calculated. This segment is therefore executed many times (roughly the simulation period divided by the time step). It is normally the most extensive segment in a model. The segment contains the complete description of the model dynamics, together with any other computation required during the simulation. For some models the program consists of just the DYNAMIC segment. Then, the segment may be indicated explicitly by the label DYNAMIC, but, if no INITIAL and TERMINAL segments are identified, one can omit it.

The TERMINAL segment can be used for computations and specific output that can only be obtained at the end of the simulation run. This could be a computation based on the final values of one or more variables. As in the INITIAL segment, the computations are executed only once. Also the TERMINAL segment is optional.

CSMP is provided with a sorting algorithm to free the user from the task of correctly sequencing the statements. A program is correctly sorted when all input variables to the right hand side of an equation are known. The statements in the INITIAL and DYNAMIC segments are placed in computational order by CSMP. The statements in the TERMINAL segment, however, have to be sorted by the user.

3.3 Integration of rate equations

In Listing 3.1 the state variables are placed directly after the DYNAMIC label according to the assumption that rates depend at each moment on the values of state and driving variables and that thus the state of each system and the driving variables should be quantified before the rates of change can be calculated. A state variable, e.g. STATE1 in line 22 in Listing 3.1, is the output of the structure statement INTGRL, in which the net rate of change NRATE1 is integrated over time. To calculate the rate at time zero, the state should be known at that moment. Therefore, the initial state

should be supplied, i.e. the first argument in the INTGRL statement (STAT10 in line 22). Usually this will be measured data. The mathematical equivalent of line 22 is

$$y = y_0 + \int_{t_0}^t \frac{dy}{dt} \cdot dt \quad (3.1)$$

where y_0 and dy/dt are the initial value and the net rate of change of the state variable: STAT10 and NRATE1 in line 22.

The driving variables that characterize the influence of the outside world on the system should follow the state variables, so that these data are synchronously available as inputs for the rate calculations. Then, the rate equations can be defined.

The INTGRL statement sets up the integration scheme, but it does not specify which numerical integration method should be used to solve the rate equations. This is done by the METHOD statement. The INTGRL and METHOD statements together fully provide for numerical integration. Although CSMP offers a number of integration methods, only the rectangular method is used in Part A of this book. In CSMP this is invoked by: METHOD RECT (line 18). More sophisticated and accurate integration methods and their selection will be introduced in Chapter 6.

Obviously, the state, rate and driving variables vary in time and should be placed in the DYNAMIC segment of the model. The integration method, however, has to be communicated to the program only once, and can thus be placed in the INITIAL.

The sequence of statements discussed here would also be obtained by the sorting algorithm of CSMP. Statement sequencing in CSMP programs is discussed, however, to reinforce the concept of systems analysis that rates of change can be calculated when the state and driving variables are known. Moreover, statement sequencing often improves program comprehensibility and it facilitates the transition to simulation in FORTRAN (Chapter 10).

3.4 The time loop

Time is the independent variable in simulation models. After each integration this variable needs to be incremented with the time step, Δt , used for the numerical integration. CSMP updates time automatically. The user has only to specify a number of so-called timer variables. These include the start time of the simulation (TIME), the time interval of integration (DELT), the time intervals for tabular (PRDEL) or plotted (OUTDEL) output, and the total simulation period (FINTIM). Numerical values of PRDEL and OUTDEL may differ. All of these variables are placed on the TIMER statement (Listing 3.1, line 15 and 16). The timer label has to be communicated to the program only once. Therefore, it is placed in the INITIAL segment.

Exercise 3.1

Reason that time could be kept track of by the statement: T = INTGRL(0.,1.).

3.5 Data input and output

Data input, Listing 3.1 line 4 to 9, concerns the numerical values of the initial

conditions (INCON) of the state variables, the constants (CONST), the parameters (PARAM), and the functions (FUNCTION). The INCON, CONST, and PARAM labels could be used interchangeably, but it is recommended to reserve the label INCON for the values of the state variables at time zero, the label CONST for physical constants, e.g. π , the gas constant, Avogadro's number, etc., and the label PARAM for parameters that may differ for each simulation run. Any relationship between y and some independent variable x , may be communicated to the program by the label FUNCTION. The data pairs are given as numerical values of (x_1, y_1) , (x_2, y_2) , etc., where x -values should increase monotonically. The function name, for instance DRV1TB (DRiving Variable 1-TaBle) in line 8 of Listing 3.1, is later referred to by the selected interpolation method (line 25). Different interpolation methods are discussed in Section 3.6.

Numerical values for the initial conditions of the state variables, constants and parameters could also be calculated in the initial segment.

Data output, Listing 3.1 line 10 to 13, is invoked by listing the variables on the label PRINT or OUTPUT. With the PRINT label the numerical values of up to 55 real variables (see Section 3.8) may be printed in a tabular form at each PRDEL interval. The PRINT label can be used only once in a CSMP program: a second label would override the first. With the OUTPUT label the numerical values of up to 5 real variables may both be printed in tabular form and plotted at each OUTDEL interval. When the number of OUTPUT variables exceeds 5, graphical output is suppressed and printed output (of up to 55 variables) is given alone. More than one OUTPUT label may be used.

The output document can easily be modified by the PAGE label. This is illustrated in line 13 of Listing 3.1: PAGE NTAB=0, GROUP=2, WIDTH=80. This line would cause suppression of the tabular output (NTAB=0) of the variables listed on the OUTPUT label in favor of the largest possible resolution in the graphical output. Furthermore, the first two variables listed would be plotted at the same scale (GROUP=2) while the remaining variables are plotted on their own scales, and the width of the plot is 80 columns (WIDTH=80). The WIDTH label may be used if your computer terminal cannot be set to a width of 132, which is the maximum value of the paper width. Other practical PAGE parameters are NPLOT and LOG. NPLOT can be used to indicate the number of variables that have to be plotted and tabulated (with a maximum of 5), while the remaining variables will be tabulated only. LOG=3 specifies that the first 3 variables in the output group have to be plotted on a logarithmic scale.

Although CSMP sorts the statements, the place of the PAGE label may affect the output result: a PAGE statement following an OUTPUT statement is assigned to that statement; a PAGE statement preceding all the OUTPUT statements applies to the whole group. The OUTPUT statement is clearly more flexible than the PRINT statement. The practical value of the PRINT statement remains, however, because many applications only require tabular output.

Both the input and output data have to be communicated to or calculated in the program only once. Therefore, they are preferably placed in the INITIAL segment.

3.6 Interpolation

The data pairs given on a FUNCTION label (see Section 3.5), may be interpolated in different ways. The simplest way of interpolating between two points is to connect them by a straight line:

$$y = y_1 + (x - x_1) \cdot (y_2 - y_1) / (x_2 - x_1) \quad (3.2)$$

Exercise 3.2

Draw a graph with an x - and y - axis and two data pairs x_1, y_1 and x_2, y_2 , and derive Equation 3.2.

In CSMP, linear interpolation between data pairs is obtained by means of the AFGEN function (Arbitrary Function GENERator):

$$Y = \text{AFGEN}(YTB, X)$$

where Y is the result of interpolating the relationship between the y - and x -values, supplied to the program in function YTB , with the independent variable x as input. Note that in the function definition the independent variable comes first in each data pair, whereas in the interpolation function (AFGEN) the independent variable comes as second argument.

Non-linear interpolation of data pairs is also possible. Then, more than 2 data pairs are required, e.g. three pairs to calculate a parabola: $y = a \cdot x^2 + b \cdot x + c$, or four points to calculate a cubic relation: $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$. In general one can calculate a n^{th} order function through $(n+1)$ data points. The linear, parabolic and cubic interpolation methods are often called first, second and third order interpolation methods, respectively.

Exercise 3.3

Given is the function

$$\text{FUNCTION } YTB = (0., 1.) , (1., 1.) , (1.5, 0.) , (3., 0.) , (5., 0.)$$

Plot the data pairs of the function on a graph, and draw the two parabolae which can be calculated through the pairs $(0., 1.), (1., 1.), (1.5, 0.)$ and $(1., 1.), (1.5, 0.), (3., 0.)$

In CSMP, parabolic interpolation between data pairs is obtained by means of the NLFGEN function (Non Linear Function GENERator):

$$Y = \text{NLFGEN}(YTB, X)$$

Higher order interpolation between data pairs may be obtained by means of the FUNGEN function (FUNction GENERator):

$$Y = \text{FUNGEN}(YTB, \text{ORDER}, X)$$

where the integer variable ORDER (see Section 3.8) should have a value of 1 to 5. When the variable ORDER equals 1 or 2, the AFGEN or NLFGEN functions are

generated, respectively.

Now a number of possibilities of data interpolation are known, the best method should be chosen. Therefore, first plot the data that should be interpolated. The data will often show some scatter and may be approximated by a smooth curve, which could be represented by short straight lines. Linear interpolation is adequate then. Sometimes, a parabolic relationship is expected a priori, e.g. from theoretical considerations. This could be a reason to use parabolic interpolation, but it would also be possible to fit a parabolic or another appropriate mathematical equation to the data. Interpolation is no longer needed then, because the data could be calculated from the auxiliary equation.

Equations are to be preferred to arbitrary functions, since generally they give more insight in the relationship between the data. If interpolation is applied and the data pairs are close enough to be connected by straight lines, especially linear interpolation is recommended, because no unexpected results occur.

Exercise 3.4

The following CSMP program demonstrates the use of the AFGEN and NLFGEN functions:

```
TITLE Linear and parabolic interpolation
INITIAL
FUNCTION YTB = (0.,1.) , (1.,1.) , (1.5,0.) , (3.,0.) , (3.5,0.)
OUTPUT Y1, Y2
PAGE GROUP, WIDTH = 80
TIMER FINTIM = 3.5, OUTDEL = 0.1, DELT = 0.1

DYNAMIC
Y1 = AFGEN (YTB, TIME)
Y2 = NLFGEN (YTB, TIME)

END
STOP
ENDJOB
```

Type the lines of the program on your computer (terminal), urge your computer centre to install CSMP or another simulation language, or install PCSMP on your personal computer. If another simulation language is installed, inform yourself on the deviation in notation between that language and CSMP.

When CSMP is executed, two files are created: FOR03.DAT and FOR06.DAT. The FOR03.DAT lists the user's model together with an indication of the memory space used. Also messages on programming errors are reported here. The FOR06.DAT lists all model parameters and the results of the calculations. Messages on calculation errors during run time are reported in this file. (Since no integration takes place, specification of the integration method is not needed in this interpolation program.)

Then:

- a. Run the program and study its output. Compare the results with your calculations of Exercise 3.3.
- b. Which data pairs are used for parabolic interpolation in the interval $0 \leq t \leq 1$, and which in the subsequent time intervals?

A selection of a number of other useful functions available in CSMP is described in Tables 3.1 and 3.2.

Table 3.1. Some CSMP functions.

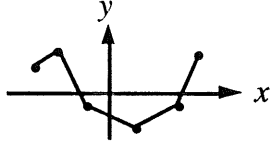
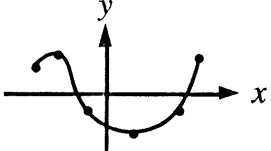
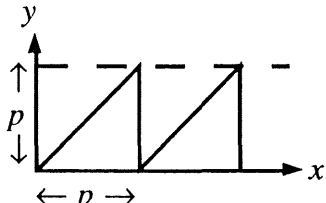
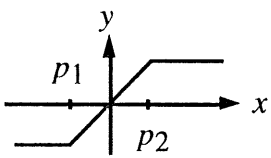
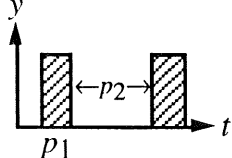
CSMP III Functions	Equivalent Mathematical Expression
Integrator $Y = \text{INTGRL}(IC, X)$ where $IC = y_{t_0}$	$y(t) = y(t_0) + \int_{t_0}^t x \cdot dt$ where $t_0 = \text{start time}$ $t = \text{time}$
Arbitrary function generator (linear interpolation) $Y = \text{AFGEN}(\text{FUNCT}, X)$	$y = f(x)$ 
Arbitrary function generator (quadratic interpolation) $Y = \text{NLFGEN}(\text{FUNCT}, X)$	$y = f(x)$ 
Modulo function $Y = \text{AMOD}(X, P)$	$y = x - n \cdot p$ n is an integer value such that $0 \leq y < P$ 
Limiter $Y = \text{LIMIT}(P1, P2, X)$	$y = p_1 ; x < p_1$ $y = p_2 ; x > p_2$ $y = x ; p_1 \leq x \leq p_2$ 
Not $Y = \text{NOT}(X)$	$y = 1$ if $x \leq 0$ $y = 0$ if $x > 0$
Input Switch Relay $Y = \text{INSW}(X1, X2, X3)$	$y = x_2$ if $x_1 < 0$ $y = x_3$ if $x_1 \geq 0$
Dead time (DELAY) $Y = \text{DELAY}(N, P, X)$ where $N = \text{number of points sampled in interval } P$ (integer constant) and must be ≥ 3 and ≤ 16378 , $P = \text{delay time}$, and X the function or variable that is delayed	$y = x \cdot (t - p) ; t \geq p$ $y = 0 ; t < p$
Impulse generator $Y = \text{IMPULS}(P1, P2)$ where $P1 = \text{time of first pulse}$ $P2 = \text{interval between pulses}$	$y = 0 ; t < p_1$ $y = 1 ; (t - p_1) = k \cdot p_2$ $y = 0 ; (t - p_1) \neq k \cdot p_2$ $k = 0, 1, 2, 3, \dots$ 

Table 3.2. Some FORTRAN functions, which can be used in CSMP statements.

FORTRAN Functions	Equivalent Mathematical Expression
Exponential Y = EXP (X)	$y = e^x$
Trigonometric sine (argument in radians) Y = SIN (X)	$y = \sin (x)$
Trigonometric cosine (argument in radians) Y = COS(X)	$y = \cos (x)$
Square root Y = SQRT(X)	$y = \sqrt{x}$
Largest value (Real arguments and output) Y = AMAX1(X1,X2)	$y = \max (x_1, x_2)$
Smallest value (Real arguments and output) Y = AMIN1(X1,X2)	$y = \min (x_1, x_2)$
Integer part of the argument value, by truncation (Real argument and integer output) IX = INT(X)	$ix = \text{int} (x)$
Real conversion of the argument (Integer argument and real output) X = REAL(IX)	$x = \text{real} (ix)$

3.7 Reruns

If a simulation run is to be repeated with new input data and/or execution control statements (e.g. TIMER data), the new data could be given between two END-statements (Listing 3.1, lines 32 to 37). This feature enables the user to do a number of simulation runs, without having to load the program structure in the computer again and again. All data enumerated in Listing 3.1 under the headings *Data input*, *Data output*, *The independent variable time*, and *The integration method* may be adapted in a rerun. Data that are not changed in a rerun specification will retain their original values. Only data and not structure (e.g. rate equations) can be changed in a rerun.

3.8 Some elements of CSMP

Numeric constants There are two types of constants: real and integer. Real (floating-point) constants are numbers written with a decimal point, with a maximum of 7 digits. A real constant may be followed by a decimal exponent written as the letter E, followed by a signed or unsigned one- or two- digit integer constant. The decimal exponent E format forms a real constant that is the product of the real constant portion multiplied by 10 raised to the desired power. E.g. $213.15 = 2.1315E2 = 2131.5E-01$. Real constants are restricted to a total of 12 characters. Integers are whole numbers with a maximum of 10 digits without a decimal point.

Variables The name of a variable can contain one to six characters and the first character must be a letter. No blanks or special characters (e.g. + - * / = : () \$ ' , .) are allowed in a name. For so-called 'reserved words' one is referred to the Program Reference Manual (IBM Corporation, 1975). All names of labels, functions and data statements are reserved words. In CSMP programs, from the label TITLE to the label STOP, all variables are declared real automatically. When using integer variables in a program, these variables should be explicitly declared at the beginning of the program by the label FIXED (Listing 3.1 line 3). For example, the integer variable ORDER in the FUNGEN function in Section 3.6 should appear on the FIXED label.

Units With respect to the use of units of measurement in software, it is advised to use SI units (m, s, g, mol, Pa). If required, separate conversion routines can be written to provide other commonly-used units of measurement as the program's output.

Operators The operators and the hierarchy of the operations are similar to FORTRAN, on which CSMP is based:

()	grouping of variables and/or constants	1 st
**	exponentiation	2 nd
*	multiplication	} 3 rd
/	division	
+	addition	} 4 th
-	subtraction	
=	replacement	5 th

Functions and expressions within parentheses are always evaluated first. For operators of the same hierarchy, the component operations of the expression are performed from left to right. There is an exception for exponentiation, where the evaluation is performed from right to left. Thus, the expression $A**B**C$ is evaluated as $A**(B**C)$.

Number of positions on a line All executable statements should be positioned in the first 72 positions of a line. Positions 73 to 80 may be used for comments.

Statement continuation To continue statements or functions on the next line, it should be followed by typing three dots (...) on the line to be continued (Listing 3.1, lines 8-9 and 15-16). The last dot may appear in the 72th position.

Comments Explanatory remarks in programs may be given following an asterisk (*) (Listing 3.1, lines 4, 10, 14, 17, etc.).

3.9 Syntax

Some syntactic rules are proposed to help increase readability of programs. These are:

- split up your program into an INITIAL, a DYNAMIC and if necessary, a TERMINAL segment;
- lump all parameter specifications at the beginning of the INITIAL, to have an easy overview of them;
- place all INTGRL statements together at the beginning of the DYNAMIC;
- use blanks (spaces) in your equations to line up e.g. equal-signs (=) and to distinguish between data pairs in FUNCTION statements;
- use blank lines and ***-lines to distinguish between different program parts;
- make short comments in your program at the appropriate place;

Most of these syntactic rules have been applied in Listing 3.1.

3.10 Concluding remarks

Although the essentials of CSMP that were described in this chapter look simple, they enable us to solve complicated rate equations without a thorough knowledge of numerical mathematics. Even when one would not know what time coefficients are involved, it is possible to get trustworthy solutions by the trial and error method described in the answer to Exercise 3.8b. The CSMP algorithm is a so-called explicit scheme. This means that new values of the state variables can be calculated from the current values, the driving variables, and the parameters involved. (This complies with the basic assumption in systems analysis as mentioned in the introduction of this chapter.) Contrary to explicit schemes are the implicit schemes, where a set of n equations with n unknowns need be solved simultaneously at each time step, often with iterative methods. Explicit schemes are more easily developed than implicit schemes, and one can thus focus on the representation of ecological problems in terms of rate equations, rather than on the mathematical difficulties involved in implicit schemes.

In the rectangular integration method the time step, Δt , is fixed and based on the smallest time coefficient, τ , in the model. This implies that slow processes (with large time coefficients) are integrated with a higher accuracy than fast processes (with small time coefficients). Moreover, since the time coefficient that determines the time step often varies in time (see e.g. Exercise 3.8), the accuracy of the calculations in the model also varies, because this is a function of the ratio $\Delta t/\tau$. It would be more appropriate when the accuracy of the calculations would be constant, by adapting the time step of integration. Integration methods that achieve this are discussed in Chapter 6.

Exercise 3.5

- Write a CSMP program for the 'animals' example from Chapter 2 (Figure 2.1, Equation 2.2). Take the relative growth rate (RGR) equal to 0.1, and FINTIM=10.0.

- b. What is the time coefficient of the model, and what time increment of integration (DELTA) would you select?
- c. Run the program and study its output.

Exercise 3.6

Write a CSMP program for the 'reservoir' example from Chapter 2 (Figure 2.1, Equation 2.3). Take $TC=4.0$, the maximum water level (WATMAX) as 16 and choose FINTIM approximately 3 times larger than TC: in 3 times TC, approximately 95 percent of the maximum is reached in this proportional process (compare the result of Exercise 2.12).

Exercise 3.7

- a. Write a CSMP program for the 'car' example in Chapter 2 (Figure 2.1, Equation 2.1). Use instead of a constant rate, the following function of speed (RDIST, km h^{-1}) against the independent variable of time (h):

```
FUNCTION RDIST = 0.0, 80.0 , 2.0,80.0 , 3.0,120.0 , ...
                5.0,120.0 , 6.0,40.0 , 10.0, 40.0
```

- b. Plot the function. How would you select the size of the integration interval to properly read the rate function?
- c. How would you use the graph to manually calculate the path travelled?

Exercise 3.8

- a. Write a CSMP program for the 'yeast-sugar' example from Section 2.6. Try to find reasonable parameter values. Initialize the state variable yeast at 1/10 of the equilibrium value that will eventually be attained, and initialize the state variable sugar at its equilibrium value before inoculation of the vessel with yeast.
- b. Estimate the time interval for rectangular integration to solve this set of rate equations.
- c. Run the program and study its results. Does the model reach an equilibrium, as can be calculated from Equations 2.21 and 2.22?

Exercise 3.9

- a. Write a CSMP program for the 'animals' example from Chapter 2 (Figure 2.1, Equation 2.2), in which the relative growth rate is now a function of temperature:

TEMP ($^{\circ}\text{C}$)	0	10	20	30	40	50
RGR (h^{-1})	0	0.08	0.16	0.21	0.24	0.25

The temperature is given by a sine function that depends on time:

$$\text{TEMP} = \text{AVTMP} + \text{AMPTMP} * \text{SIN}((2 * \text{PI} / 24.0) * \text{TIME})$$

where time (TIME) is expressed in hours, and the mean (AVTMP) and amplitude (AMPTMP) of the temperature amount to 20°C and 10°C , respectively. The angular frequency of the temperature wave, often denoted by ω , is defined as $2\pi / t_c$, with t_c the time required for the wave to make a complete cycle of 2π radians. Omega (ω) is given here as the ratio $2 * \text{PI} / 24.0$.

- b. What is the maximum and the minimum time coefficient in this model for the given parameters?
- c. What is the dimension of the argument of the sinus function?
- d. Estimate from b) the time interval for rectangular integration to solve the rate equation. Does this time interval nicely follow the sinus function?
- e. Run the program for two days and study its output.