

## 2.2 Modelling of ageing, development, delays and dispersion

J. Goudriaan and H.J.W. van Roermund

### 2.2.1 Introduction

In a model designed for a population with only one generation, the development stage can be treated as a single state variable (Subsection 2.2.3). However, when there is a distribution of ages, or stages of development, the boxcar train is a suitable method that can be used to simulate the development process of the entire population (Subsection 2.2.4). During the simulation process, some dispersion (variability) may occur within the development rates of different individuals. Three types of boxcar train are possible, differing mainly in this dispersion of development rate. In the escalator boxcar train, dispersion is virtually absent (Subsection 2.2.5). In the fixed boxcar train, dispersion is quite substantial and rigidly determined by the number of boxcars (Subsection 2.2.6). The fractional boxcar train includes a parameter which allows the dispersion to be varied between these two extremes, and to be altered during the simulation process itself (Subsection 2.2.7).

The CSMP statements (see Appendix 5), and FORTRAN subroutines used to implement these methods are given, and the approach is illustrated with a simple application.

### 2.2.2 Development and delay

A good example of a stable and well measurable rate of development can be found in a bird's egg; the time between laying and hatching is rather fixed. Both moments are two clearly marked milestones in the life of a bird. Because the duration of time between these moments is rather stable, the rate of hatching in a population of birds is the same as the rate of laying, delayed over the period of brooding. Such delays are quite common in the description of biological processes. For instance, the well-known equation for exponential growth:

$$\frac{dy}{dt} = r \cdot y \quad \text{Equation 25}$$

can be written more specifically to describe adult birds only as:

$$\frac{dy}{dt} = r \cdot y_{t-p} \quad \text{Equation 26}$$

where  $p$  stands for the duration of egg and juvenile stages combined. The value of the relative growth rate,  $r$ , is not the same in Equations 25 and 26. In the

development of disease for example, latency periods can often cause delays (see Chapter 3).

When the delay period is always the same, a simple CSMP function (see Appendix 5) can be used to implement the delay:

$$\text{OUTFL} = \text{DELAY}(\text{N}, \text{PERIOD}, \text{INFLOW})$$

where the rate *INFLOW* is delayed over a period *PERIOD* to produce the outflow rate *OUTFL*. *N* stands for the number of sampling points describing the shape of the inflow rate during the delay period, and should typically be of the order of *PERIOD/DELTA* (*DELTA* is the time step of integration,  $\Delta t$ ).

---

### Exercise 22

Use this *DELAY* function to simulate hatching 20 days after laying. During days 1 and 2, 100 eggs are laid, and no more afterwards.

---

A major limitation of the *DELAY* function is that it cannot be used with variable delay periods. Another limitation is that operations on the quantities delayed, such as mortality or emigration, are not possible.

### 2.2.3 *Simulation of development of a single generation*

In warm-blooded animals, development and ageing can hardly be distinguished, but in other organisms these rates can be completely different. For instance, plants of the same species may flower at moments that are more determined by temperature and day length than by time since emergence. Usually, at low temperatures the developmental processes run much slower than at high temperatures.

Discernible stages of development have been given names, for instance 'anthesis', 'dough-ripe' (in wheat), 'silking' (in maize), and are largely species specific. Numbers have also been given to these stages, so that they can be more easily quantified. For instance, in the general scale of development, as used in the crop growth model *SUCROS87* (Section 4.1), 'anthesis' was given the value 1, and 'maturation' the value 2. In the literature, more refined scales have been developed, e.g. for wheat (Zadoks et al., 1974; Reinink et al., 1986), maize (Groot et al., 1986) and rape seed (Remmelzwaal & Habekotté, 1986).

Rate of development can be defined as the numerical distance between two stages, divided by the time required to pass from one stage to the other. The problem with empirical scales is that the time intervals between subsequent stages are often not equal, even under constant conditions. Then, the empirical scale must be projected onto a fictive scale that meets this requirement of

a homogeneous rate. Alternatively, the rate of development varies with the stage of development.

In the following example (Remmelzwaal & Habekotté, 1986), a simulation approach will be shown for the development of rape seed from stage 9 (end of flowering) to stage 15 (maturation). Over this range, the development rate (DEVR) is proportional to temperature above 6°C, but it is assumed here that above 25°C no further acceleration occurs. This means that the response between 6°C and 25°C can be quantified in terms of degree-days with a base temperature of 6°C. 490 degree-days are needed to proceed from stage 9 to stage 15.

---

### Exercise 23

How long would this period be at 10°C and how long at 20°C? What would be the rate of development at these temperatures?

---

The state variable 'stage of development' (STAGE) can now be simulated by the following CSMP statements:

```
STAGE = INTGRL(9., DEVR)
DEVR = AFGEN(DEVTB, TA)
FUNCTION DEVTB = (0., 0.), (6., 0.), (25., 0.233), (30., 0.233)
```

where TA stands for air temperature in °C, and DEVTB for DEVelopment TaBle with the development rate as a function of temperature.

---

### Exercise 24

Check the consistency of this model and the manual calculation of Exercise 23 by running this simple model for 10°C and for 20°C.

---

#### 2.2.4 *The boxcar train*

Using the above method, it is only possible to keep track of the stage of development of the entire population, because all individuals are synchronized. This situation is quite common in field crops. In insect populations, especially if they are polyvoltine, several stages of development occur simultaneously. Each stage would then require its own simulation. Also, new generations are continuously being born, which adds to the complexity.

The boxcar train technique provides for all possible development stages simultaneously. Before simulation starts, the developmental axis of one stage is broken up into a number of classes or boxcars, each with identical development

widths. If necessary, several separate boxcar trains may be chained; for instance, one to allow for all egg stages, one for all juvenile stages and one for all adult stages. This separation may be necessary to fulfil the requirement of homogeneity of development rate within a boxcar train. It is then much easier to acquire boxcars of identical development widths, at least within the boxcar train.

After this classification, each boxcar is initialized with the number of individuals contained in it. In principle, a histogram can now be drawn for the development distribution of the population in the stage considered (Figure 13a).

If a higher resolution of the development axis is required, the number of boxcars in the boxcar train should be increased (Figure 13b). In Figure 13b, the vertical axis is scaled so that the total area is still the same as in Figure 13a. This can be achieved by plotting vertically not just the number per boxcar, but this number divided by its development width  $\gamma$ . The number obtained in this way is a boxcar-averaged concentration  $c$ , as an approximation of the 'true' concentration which could vary with stage  $g$ , given by the broken line in Figure 13b.

If there is no mortality, this concentration-distribution function simply shifts to the right without any change in shape. Of course, new individuals may enter at  $g = 0$ , and at  $g = g_f$  individuals are removed from the scene. Theoretically, it is possible to store the shape of the graph of  $c(g)$  into a computer with a very high degree of resolution and, accordingly, to simulate the development process. However, computer limitations prevent this procedure, and we have to live with a representation as given by histograms in Figure 13a.

The question now is how to allow for the continuously occurring development drift, which shifts all individuals to a higher stage of development at the same rate. In principle, there are two options available to simulate this process:

1. Continuously shift the entire distribution, shown by the histogram in Figure 13, to higher values of development, including the boxcar boundaries. Only the beginning ( $g = 0$ ) and the end ( $g = g_f$ ) are fixed. This system is called the escalator boxcar train.

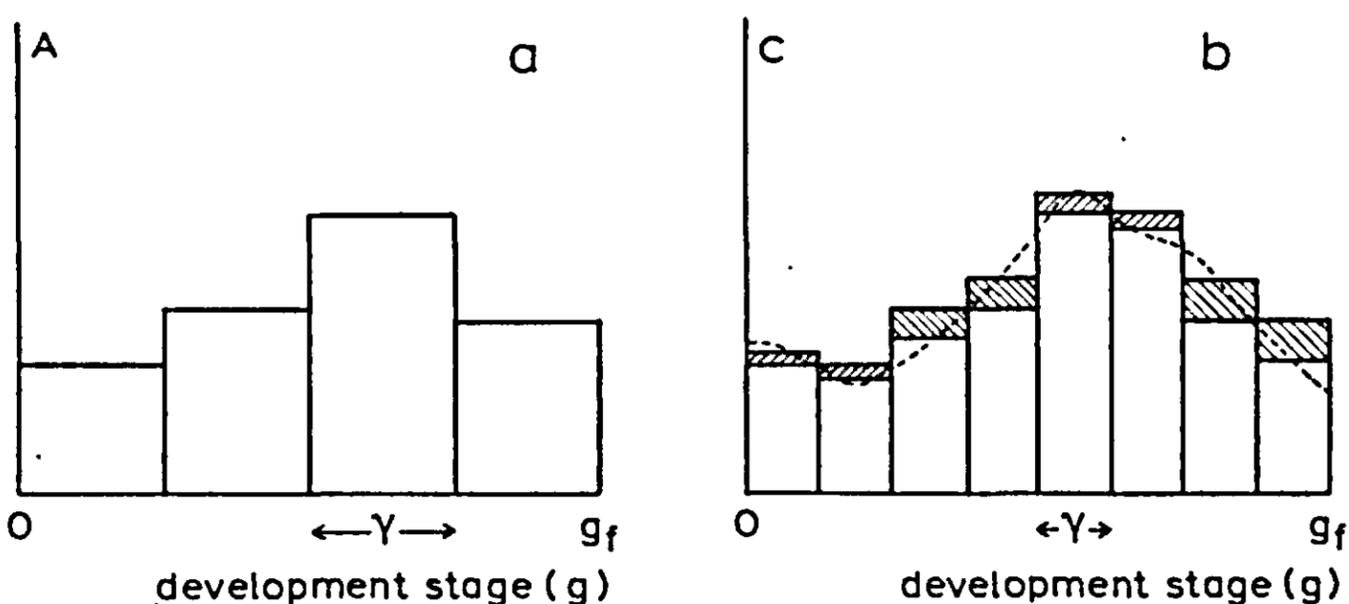


Figure 13. Distribution of numbers of individuals (A) or of concentrations (c) with stage of development (g).  $g_f$  stands for the final value of g, and  $\gamma$  for the width of a single boxcar. For explanation see text.

2. Keep the location of the boundaries of the boxcars (the bars in the histogram) fixed, but allow the individuals to flow from one boxcar to the next. The rate of movement is proportional to the rate of development, and also to the concentration (height of the bar). This system is called the fixed boxcar train.

These two types of boxcar train do not differ in mean delay time. However, in the escalator boxcar train all individuals are about equally delayed, whereas in the fixed boxcar train some are more and others less delayed. The reason for this variability is that for each individual the probability of flowing to the next boxcar is the same, whether that individual has just arrived or has been waiting quite a while. Due to this stochastic process, there is variance in the residence time in each boxcar. The variance of duration of through-flow through the fixed boxcar train, causes a levelling of peaks and dips originally present in the inflow curve (Berger & Jones, 1985). An intensive but brief pulse will be buffered in the relatively long residence time of the boxcars, and will result in dispersion during development. This so-called numerical 'dispersion' is an artefact of the system, but is a useful by-product of the boxcar train. Whenever such variance is observed in nature, the fixed boxcar train may help to simulate this phenomenon, although it does not add any explanatory value about its causes.

It will be explained later that it is possible to hybridize both systems into the fractional boxcar train. Using this hybrid method, the degree of variance of through-flow can be controlled to match the observed variance.

### 2.2.5 *The escalator boxcar train*

In the escalator boxcar train, the developmental process is simulated by a continuous developmental drift of the boxcar boundaries. It is essential that these boundaries are chosen so that each boxcar covers the same developmental width; this means that the duration between adjacent boundaries are made equal for all of them. After the developmental process has completed one such sub-unit of development, boxcar width  $\gamma$  (Figure 14), the entire population will have gradually shifted to the right by exactly one boxcar, and so all boxcar numbers can be reset.

This process is schematically given in Figure 14. An escalator boxcar train with 4 boxcars is presented here, so that the total development range covered ( $g_f$ ) is equal to  $4\gamma$ . Although the rate of development is not necessarily constant, it should change simultaneously for all boxcar boundaries.

Immediately following the start of simulation, a gap opens between  $g = 0$  and the lower boundary of boxcar 1. This gap is filled by a new boxcar, with number 0, which will receive the newcomers into the boxcar train. As far as the functioning of the boxcar train is concerned, it does not matter where these newcomers have come from, whether they have been generated as an external driving force, been produced as offspring from the boxcar train itself, or have simply come from a preceding boxcar train.

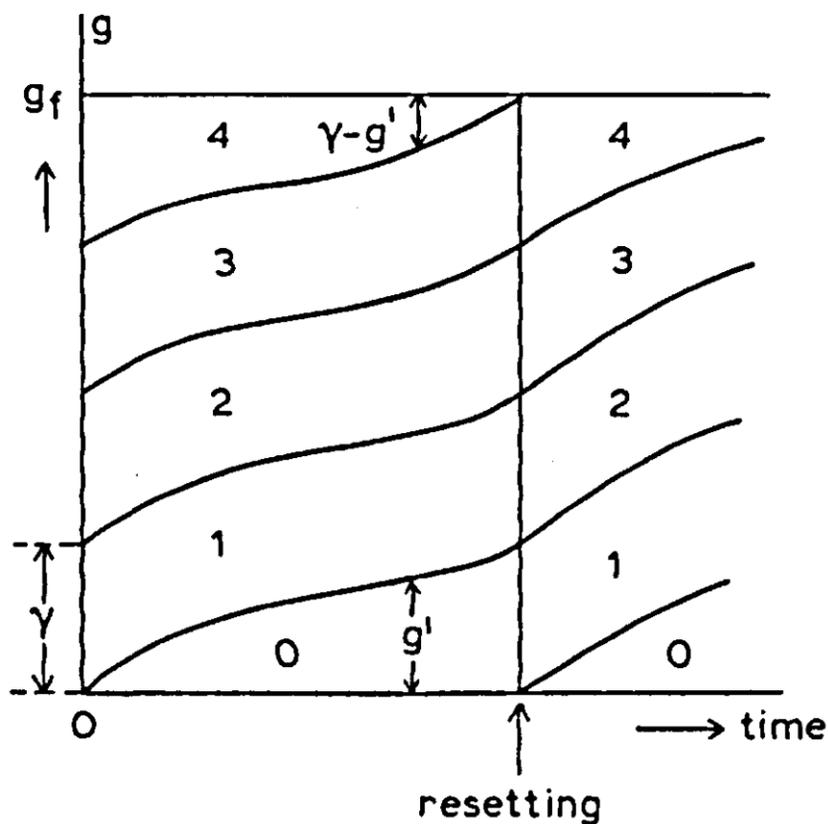


Figure 14. The escalator boxcar train. Time dependence of position of the boxcars and their numbering. For explanation see text.

At the other end of the boxcar train, the last boxcar is contained between a fixed end boundary and an upward-moving lower boundary. The distance between them ( $\gamma - g'$ ) keeps shrinking, and so without outflow the concentration in the last boxcar,  $c_N$ , would grow beyond limit. Such unlimited compression is prevented because there is also a rate of outflow,  $Q_{out}$ , defined as:

$$Q_{out} = v \cdot c_N$$

with

$$c_N = A_N / (\gamma - g')$$

where  $v$  is the rate of development,  $A_N$  is the amount in the last boxcar and  $g'$  the cyclic development stage ( $0 < g' < \gamma$ , Figure 14).

When there is no mortality, the relative rate of decrease of amount  $A_N$  and of the remaining width  $\gamma - g'$  are equal, and so the concentration  $c_N$  does not change. The rate of outflow  $Q_{out}$  is then proportional to the rate of development  $v$ , exactly as we want it to be.

When there is mortality,  $Q_{out}$  decreases within each development cycle. In fact all boxcars, including the zero boxcar and the last one, may or may not lose individuals due to mortality, but they do not exchange them. This lack of exchange preserves the shape of the development distribution curve.

The resetting event occurs when the development process has covered one sub-unit  $\gamma$ , the width of one boxcar. At that moment, each boxcar has reached the position occupied by its successor at the start of the simulation. The boxcar numbers are then reset, the last boxcar is removed entirely and a new zero boxcar is opened.

A calculation of the delay values and residual dispersion in the escalator boxcar train is given in Appendix 3.

---

### Exercise 25

If the escalator boxcar train is used to describe the development of rape seed from stage 9 to stage 15, in 4 boxcars, what is then the value of  $\gamma$ ? How would you formulate the development rate DEVR? If the degree-day simplification is permitted ( $6 < T_a < 25^\circ\text{C}$ ), on how many degree-days does the resetting (or 'shift') event occur?

---

### 2.2.6 The fixed boxcar train

With fixed boundaries between the boxcars, there is a continuous forward flow from each boxcar into the next to allow for the development drift. Water, cascading from tanks can be used as a physical model to visualize how the fixed boxcar train operates. The flow from a boxcar is proportional to the concentration  $c$  in it, and to the development rate,  $v$ :

$$Q_{i+1} = v \cdot c_i \quad \text{Equation 27}$$

where  $Q_{i+1}$  is the flow rate from boxcar  $i$  to boxcar  $i + 1$ . The concentration  $c_i$  is given by:

$$c_i = A_i/\gamma \quad \text{Equation 28}$$

To understand the behaviour of the population contents in the fixed boxcar train, it is best to consider first the simplified situation under a constant development rate  $v$ . Then, each boxcar will act as a first order exponential delay, which means that a single sharp input pulse will give rise to an exponentially declining output flow (Figure 15). Of course, the total area of both pulses must be the same.

The exponential shape of the decline can be derived as follows. Imagine that at time zero the boxcar considered is empty, and that consequently the output flow is also zero. Suddenly, a very brief, sharply peaked input flow fills the boxcar with an amount  $A_0$ . According to Equations 27 and 28, the output flow  $Q_{\text{out}}$  is directly proportional to the contents  $A$ :

$$Q_{\text{out}} = A \cdot v/\gamma$$

In the situation described, the inflow will be zero immediately after the passage of the brief pulse, and so the differential equation for the contents  $A$  is:

$$\frac{dA}{dt} = -A \cdot v/\gamma \quad \text{Equation 29}$$

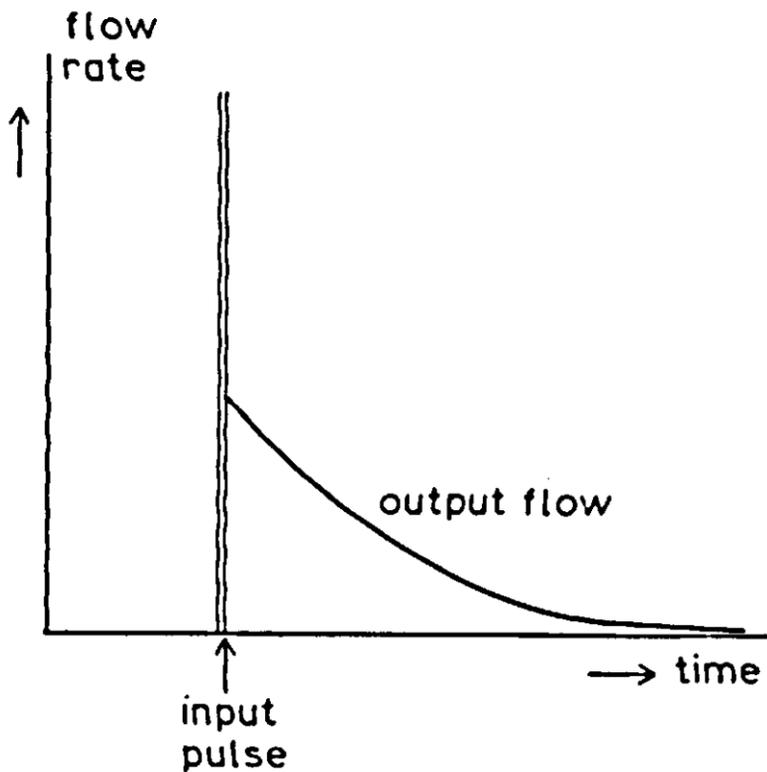


Figure 15. Output flow of a sub-unit in the fixed boxcar train in response to a sharply peaked input pulse.

Since  $v$  and  $\gamma$  are constant in this simplified situation, the solution of this differential equation for  $A$  is an exponentially declining function:

$$A = A_0 \exp(-v \cdot t/\gamma)$$

with  $A_0$  as the initial value of  $A$ , and the outflow  $Q_{out}$  given by:

$$Q_{out} = (v/\gamma) A_0 \exp(-v \cdot t/\gamma) \quad \text{Equation 30}$$

With regard to the relationship between the inflow peak and the outflow function (Figure 15), two observations can be made: (1) on average, the outflow is delayed with respect to the inflow; (2) the shape of the outflow is more dispersed over time than the inflow.

To find the value of the average delay we should remember that the inflow pulse was localized at time zero, and so the average delay is equal to the mean of time  $t$  of outflow in Equation 30. To find this mean time  $\tau$ , time  $t$  should be integrated between zero (start) and infinity, weighted with the value of  $Q_{out}$ :

$$\tau = \int_0^{\infty} t Q_{out} dt / A_0$$

Substitution of  $Q_{out}$  according to Equation 30 gives:

$$\tau = \int_0^{\infty} t \cdot (v/\gamma) \exp(-v \cdot t/\gamma) dt$$

which has the solution:

$$\tau = -\exp(-v \cdot t/\gamma)(t + \gamma/v)]_0^\infty$$

or  $\tau = \gamma/v$ . In fact, this answer is not surprising since  $\gamma/v$  is the time coefficient in the argument of the exponential function in Equation 30, and also in differential Equation 29. This is similar to the derivation in Subsection 2.1.5.

So far, the derivation has only concerned a single infinitely sharp inflow pulse, and one might wonder whether the value of the delay is independent of the shape of the inflow. This simple thought is correct, since any shape of the inflow can be broken up into a series of sharp pulses, each giving rise to its own exponentially delayed outflow. Since the whole system is linear (meaning that outflow is proportional to contents, Equation 29), the contents and outflows due to the subsequent pulses can simply be added together. Therefore, the average delay is also equal to  $\gamma/v$  (or  $\tau$ ), independent of the shape of the inflow. Each boxcar will add this delay to result in the total delay of the entire boxcar train:

$$T_{\text{total}} = N \cdot \tau \quad \text{Equation 31}$$

With regard to the average value of the total delay, the fixed boxcar train and the escalator boxcar train are similar. The difference between these methods appears in the effect on the shape of the outflow. Whereas the outflow is identical in shape to the inflow in the escalator boxcar train, it is much more dispersed in the fixed boxcar train. This means that a considerable amount of variance has been added to the time distribution of the inflow.

Statistically, the variance  $\sigma^2$  is the second order moment of the time of outflow, which can be calculated as the mean value of  $(t - \tau)^2$ , weighted with  $Q_{\text{out}}/A_0$ :

$$\sigma^2 = \int_0^\infty (t - \tau)^2 Q_{\text{out}} dt / A_0$$

With  $Q_{\text{out}}$  given by Equation 30, and using  $\gamma/v = \tau$ ,  $\sigma^2$  can be written as

$$\sigma^2 = \int_0^\infty (t - \tau)^2 (1/\tau) \exp(-t/\tau) dt$$

By using of a table of indefinite integrals, and some algebra, this expression can be shown to lead to (see Appendix 4):

$$\sigma^2 = \tau^2$$

Since each boxcar will add this amount of variance, irrespective of the other boxcars and of its own position in the cascade, the total amount of variance added by the entire boxcar train is given by:

$$\sigma_{\text{total}}^2 = N \cdot \tau^2$$

Combining this expression with the one for total delay (Equation 31) gives the interesting result for relative 'dispersion' RD:

$$\sigma_{\text{total}}/T_{\text{total}} = 1/\sqrt{N}$$

The relative 'dispersion' RD (or 'coefficient of variation CV') of the outflow time, in response to a peaked inflow decreases with the number of boxcars N. This result could be understood more easily if one were to imagine more (but narrower) boxcars in the boxcar train.

### Exercise 26

How many boxcars are needed to simulate a delay of 20 days with a 'dispersion' of 2 days (RD = 0.1)?

A system of N fixed boxcars is often termed an N<sup>th</sup> order delay. The dynamic response of the outflow after a stepwise change in inflow is given in Figure 16 for different values of N (Ferrari, 1978).

#### 2.2.7 The fractional boxcar train

Compared with the DELAY function of CSMP, both the escalator and the fixed boxcar trains are much more flexible, in that they permit variable development rates. However, they still lack flexibility in the relative 'dispersion' (or RD) of outflow. The escalator boxcar train has almost no dispersion (except for the small amount due to the distribution over a single boxcar) and the fixed boxcar train has a fixed RD of  $1/\sqrt{N}$  which, once chosen, cannot be changed during the simulation. But in several experimental data sets there is evidence that delay and dispersion are not equally influenced by e.g. temperature, and so the relative

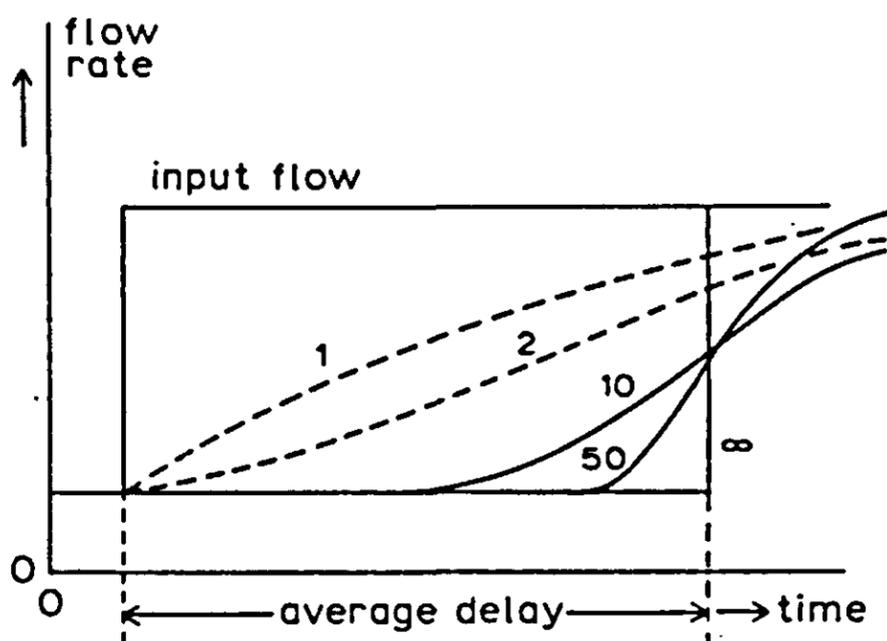


Figure 16. Response of outflow to a stepwise change in inflow for a fixed boxcar train with different numbers of boxcars.

dispersion also varies. To allow for this change during simulation, a more flexible method than that of the fixed boxcar train is needed. Such a flexible method can be obtained by hybridizing the methods of both the fixed and the escalator boxcar trains. This method will be termed the fractional boxcar train, because it is based on a fractional repeated shift.

In the escalator boxcar train, a complete shift to the next boxcar occurs at the moment of resetting. In the fractional boxcar train, it is not the complete contents that are shifted, but only a fraction  $F$  of each boxcar's contents. To compensate for the smaller amount, the shift must occur more frequently. In the escalator boxcar train the renumbering (or shift) occurs upon completion of a full development cycle,  $\gamma$ . In the fractional boxcar train method, the fractional shift occurs upon completion of a fraction  $F$  of the development cycle. This fraction ranges between 0 and 1 and can be changed during simulation. A possible time path of the boundaries between the boxcars is illustrated in Figure 17.

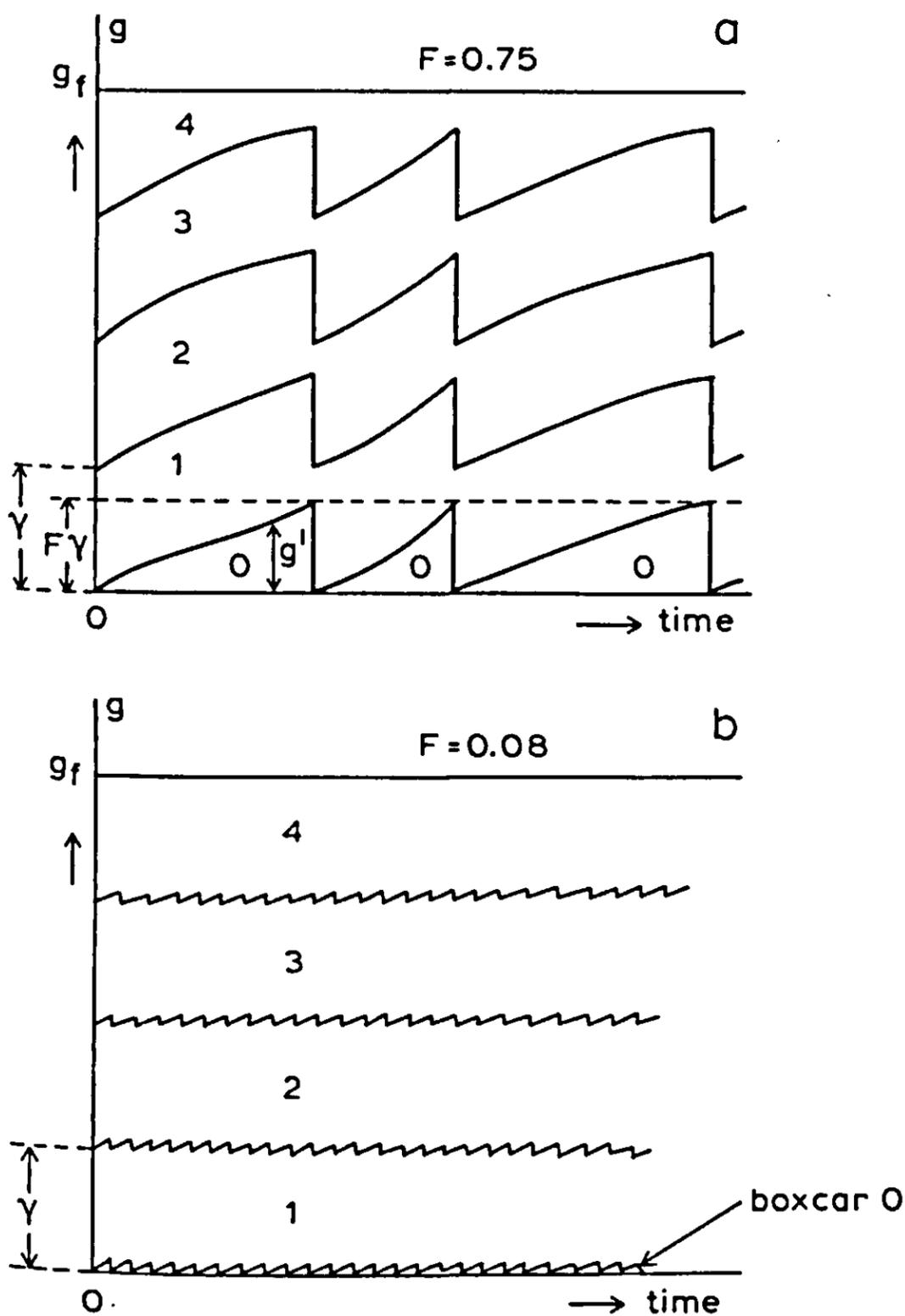


Figure 17. Sawtooth shape of time path of the boxcar boundaries in the fractional boxcar train. For explanation see text.

The value of the fraction  $F$  determines how often resetting and partial transfer of contents occurs. When  $F$  is equal to 1, the escalator boxcar train is effectively restored (Figure 14). On the other hand, when  $F$  approaches zero, the sawtooth shape of the boundaries between the boxcars is practically straight, so that the fixed boxcar train is approached. During simulation, the value of  $F$  can be varied anywhere between these two extremes, so that gradual adaptations can be made in the value of the desired dispersion.

Since movement through the boxcars is pulsewise, the differential equations must be replaced by difference equations. The cyclic development stage,  $g'$ , stands for the development elapsed since the last resetting occurred. In the escalator boxcar train,  $g'$  triggers the renumbering when it exceeds  $\gamma$ . Here, in the fractional boxcar train, the trigger level is set at  $F\gamma$ . When this level is exceeded, fractional shift occurs and  $g'$  is decreased by  $F\gamma$ . The contents of boxcar  $i$  are also reduced:

$$A_{i,j} = A_{i,j-1} - F \cdot A_{i,j-1}$$

where  $j$  counts the number of shifts since the start. Here, as for Equation 29, it is assumed that inflow into  $A_i$  is zero.  $A_{i,j}$  is then given by

$$A_{i,j} = A_{i,0}(1 - F)^j$$

A special situation occurs in the zero boxcar. The contents of this boxcar are entirely transferred to the first one, so that after the shift,

$$A_{0,j} = 0.$$

*The delay in the fractional boxcar train* The first fractional shift does not occur at time zero, but only when  $g'$  equals  $F\gamma$ . When the development rate,  $v$ , is constant, this occurs at time  $F\gamma/v$ , or at time  $F\tau$ . The expression for the average residence time,  $\bar{\tau}$ , is:

$$\bar{\tau} = \frac{1}{A_0} \sum_{j=1}^{\infty} j F \tau A_0 (1 - F)^{j-1} F \quad \text{Equation 32}$$

time    quantity transferred

This expression can be evaluated using the general expression for the sum of the series

$$\sum_{j=1}^{\infty} j r^{j-1} = \frac{1}{(1 - r)^2} \quad 0 < r < 1$$

In this equation,  $r$  can be replaced by  $1 - F$ , and Equation 32 then yields

$$\bar{\tau} = \tau$$

This result shows that the delay per boxcar is independent of the value of  $F$ . Also, the total delay  $T_{\text{total}}$  of the boxcar train is independent of  $F$ , and equal to  $N\tau$ .

*The variance in the fractional boxcar train* The variance can be evaluated from

$$\sigma^2 = \frac{1}{A_0} \sum_{j=1}^{\infty} (j F \tau - \tau)^2 A_0 (1 - F)^{j-1} F \quad \text{Equation 33}$$

deviation      quantity transferred

By using the sum of the series given above, and also the following one:

$$\sum_{j=1}^{\infty} j^2 r^{j-1} = \frac{1+r}{(1-r)^3} \quad 0 < r < 1$$

we find that Equation 33 can be simplified to:

$$\sigma^2 = \tau^2(1 - F)$$

This result shows that the variance is linearly related to the value of the fraction F. This variance occurs in each boxcar, so that the total variance of the whole boxcar train is

$$\sigma_{\text{total}}^2 = N\tau^2(1 - F) \quad \text{Equation 34}$$

### 2.2.8 Implementation of the boxcar train in CSMP

*The fixed boxcar train (see Figure 18)* First, the three types of boxcar train will be applied in a simple example. There will be 5 boxcars with initial content zero:

A = INTGRL(AI, RA, 5)
state            initial rate    number of
state            integrals

In the rest of the program, A and RA can be referred to as indexed variables, just as in FORTRAN: A(I) stands for element I of the array A. The array AI (initial values) is set at zero. It is convenient to use DO loops to calculate rates and concentrations:

```
DO I = 1,5
    C(I) = A(I)/GAMMA (see Equation 28)
ENDDO
```

Here, GAMMA is the name for the mathematical symbol  $\gamma$ , the width of a boxcar in development units, and is defined as  $1/N$ . This implies that the full range of development has the numerical value of unity. This is an arbitrary choice which could have been 100 or any other figure. In the case of another value for the full range of development, the width of the boxcars is adapted proportionally.

The rate of flow from boxcar I to boxcar I + 1 is given by the rate of development, v (DEV), multiplied by the concentration C(I):

$$\text{FLOW}(I + 1) = \text{DEV} * C(I) \quad \text{(see Equation 27)}$$

An inflow pulse of total size UNITY occurring at time PTIME can be obtained by:

$$\text{INFL} = \text{IMPULS}(\text{PTIME}, \text{FINTIM}) * \text{UNITY} / \text{DELT}$$
$$\text{PARAM PTIME} = 0., \text{UNITY} = 1.$$

The rate of development (DEV R), is also integrated to yield the physiological time, G, which can be considered to be the state of development of an immortal individual born at time zero:

$$G = \text{INTGRL}(0., \text{DEV R})$$

It should be noted that G always increases.

The inflow into the boxcar train, generated somewhere else, is used here.

$$\text{FLOW}(1) = \text{INFL}$$

and the outflow is made equal to FLOW(N + 1):

$$\text{OUTFL} = \text{FLOW}(N + 1)$$

The net flow of a boxcar is given by:

$$\text{NETFLO}(I) = \text{FLOW}(I) - \text{FLOW}(I + 1)$$

and, since there is no mortality so far,

$$\text{RA}(I) = \text{NETFLO}(I)$$

The outflow can be collected in a separate integral

$$\text{AOUT} = \text{INTGRL}(0., \text{OUTFL})$$

In this example the inflow was kept at zero, except for a single pulse at time zero with height UNITY/DELT. This discontinuous behaviour of the inflow requires the use of the rectangular integration method:

#### METHOD RECT

The following optional statements for monitoring purposes would normally not be included in a model, but study of their formulation and behaviour improves the understanding.

For this simple situation, the average delay period (ADP) can also be calculated numerically:

$$\text{ADP} = \text{INTGRL}(0., (\text{TIME} - \text{PTIME}) * \text{OUTFL}) / \text{UNITY}$$

The expression for the variance (VAR) is given by:

$$\text{VAR} = \text{INTGRL}(0., (\text{TIME} - \text{PTIME} - \text{ADPG}) ** 2 * \text{OUTFL}) / \text{UNITY}$$

ADPG must be a constant and, in fact, be equal to the final value of ADP at time FINTIM. Therefore, two runs are required, the first one to find ADP and the second one to find VAR. Fortunately, in normal simulations, there is no necessity

to calculate both ADP and VAR. Here, it is done simply to check our model and to compare it with the theoretical results. The timing is defined in a TIMER statement:

```
TIMER FINTIM = 1000., PRDEL = 100., DELT = 1.
```

Here, we require the variables AOUT, ATOT, A(1 – 5), ADP and VAR as printed output on intervals PRDEL. The variables ATOT and BALANC are used only to check the balances. Although they may seem unnecessary, in more complicated models these balance variables are extremely useful to detect omitted or twice-defined flows.

The development rate (DEVR) is given in a PARAMeter statement as 0.005, and N as 5. The variables N and I are declared fixed, and for the arrays NETFLO, FLOW and C, ample space is reserved by a STORAGE statement. For variables in the array integral this reservation is done by default.

---

### Exercise 27

Use this program to generate the curves of Figure 16 for  $N = 1, 2, 5$  and 10.

---

---

### Exercise 28.

Run the simulation model of Figure 18. Compare the simulated (mimicked) delay period (ADP) and its variance (VAR) with the values calculated arithmetically ( $T_{total}$  respectively  $\sigma_{total}^2$ ). When are the simulated values significant?

---

*The escalator boxcar train (Figure 19)* Most statements are the same as for the fixed boxcar train, so here only the differences will be mentioned. The criterion for renumbering (shifting) is when the cyclic development stage,  $g'$ , (GCYCL) has reached the value GAMMA (the width of a boxcar):

```
IF (GCYCL.GE.GAMMA) CALL SHIFT(N, GAMMA, GCYCL, A, A0)
```

The subroutine 'SHIFT' is explained at the end of this Subsection. The use of the IF statement requires a preceding NOSORT label. The shift is formulated in a FORTRAN subroutine with the necessary arguments N (number of boxcars), and the names of the treated integrals. Here, the zero boxcar (A0) appears separately because, unfortunately, the indexing of array integrals begins with one and not with zero. In the main program, A0 must also be formulated separately:

```
A0 = INTGRL(0., RA0)
```

and

```
RA0 = INFL
```

Figure 18. Listing of a program, including the fixed boxcar train.

```
TITLE FIXED BOXCARTRAIN
STORAGE FLOW(6), NETFLO(5), C(5)
FIXED N,I

INITIAL

TABLE AI(1-5)=5*0.
PARAM DEVR=0.005, N=5
GAMMA = 1./N
PARAM PTIME=0.,UNITY=1.

DYNAMIC
NOSORT

*Calculation of the states
G =INTGRL(0.,DEVR)
A =INTGRL(AI,RA,5)
AOUT=INTGRL(0.,OUTFL)

*Calculation of the rates
INFLF =IMPULS(PTIME,FINTIM)
INFL =UNITY*INFLF/DELT
FLOW(1)=INFL
DO I=1,N
    C(I)=A(I)/GAMMA
    FLOW(I+1)=DEVR * C(I)
    NETFLO(I)=FLOW(I) - FLOW(I+1)
*   no mortality
    RA(I) =NETFLO(I)
ENDDO
OUTFL =FLOW(N+1)

* Balance should be zero
ATOT=0.
DO I=1,N
    ATOT=ATOT + A(I)
ENDDO
BALANC=ATOT + AOUT -INTGRL(0.,INFL)

*Optional statements for study and checking purposes:
ADPG=1./DEVR
ADP =INTGRL(0.,(TIME-PTIME)*OUTFL)/ UNITY
VAR =INTGRL(0.,(TIME-PTIME-ADPG)**2 * OUTFL)/ UNITY
S =SQRT(VAR)

TIMER FINTIM=1000.,PRDEL=100.,DELT=1.
METHOD RECT
PRINT AOUT,ATOT,A(1-5),ADP,VAR,S,BALANC

END
STOP
ENDJOB
```

Figure 19. Listing of a program, including the escalator boxcar train.

```
TITLE ESCALATOR BOXCARTRAIN
STORAGE C(5)
FIXED N,I
```

```
INITIAL
```

```
TABLE AI(1-5)=5*0.
PARAM DEVR=0.005, N=5
GAMMA =1./N
*The cyclic development starts halfway
GCYCLI=0.5*GAMMA
PARAM PTIME=0.,UNITY=1.
```

```
DYNAMIC
NOSORT
```

```
*Calculation of the states
GCYCL =INTGRL(GCYCLI,DEVR)
G      =INTGRL(0.,DEVR)
AO     =INTGRL(0.,RA0)
A      =INTGRL(AI,RA,5)
AOUT   =INTGRL(0.,OUTFL)
```

```
IF(GCYCL.GE.GAMMA) CALL SHIFT(N,GAMMA,GCYCL,A,A0)
```

```
*Calculations of the rates
INFLF  =IMPULS(PTIME,FINTIM)
INFL   =INFLF*UNITY/DELT
RA0    =INFL
```

```
DO I=1,N-1
*   no mortality
   RA(I)=0.
```

```
ENDDO
C(N)   =A(N)/(GAMMA-GCYCL)
```

```
*To prevent a negative value of A(N), the AMIN1-function is used
OUTFL  =AMIN1(DEVR * C(N),A(N)/DELT)
RA(N)  =-OUTFL
```

```
* Balance should be zero
```

```
ATOT   =A0
```

```
DO I=1,N
   ATOT=ATOT + A(I).
```

```
ENDDO
```

```
BALANC=ATOT + AOUT - INTGRL(0.,INFL)
```

\*Optional statements for study and checking purposes:

ADPG=1./DEV

ADP =INTGRL(0.,(TIME-PTIME)\*OUTFL) / UNITY

VAR =INTGRL(0.,(TIME-PTIME-ADPG)\*\*2 \*OUTFL) / UNITY

S =SQRT(VAR)

TIMER FINTIM=200.,PRDEL=20.,DELT=1.

METHOD RECT

PRINT A0,A(1-5),AOUT,ATOT,BALANC,G,GCYCL,ADP,VAR,S

END

STOP

SUBROUTINE SHIFT(N,GAMMA,GCYCL,A,A0)

DIMENSION A(N)

DO I =N,2,-1

A(I)=A(I-1)

ENDDO

A(1) =A0

A0 =0.

GCYCL=GCYCL-GAMMA

RETURN

END

ENDJOB

Since the flows between boxcars are zero (by definition in the escalator boxcar train), the names FLOW and NETFLO are not required. Only the outflow from the last boxcar must be defined (see Subsection 2.2.5):

$OUTFL = AMIN1(DEV * C(N), A(N)/DELT)$

and

$RA(N) = -OUTFL$

The other rates of change in the boxcars are zero (no mortality yet):

DO I = 1,N - 1

RA(I) = 0

ENDDO

which can be stated in the INITIAL segment. The concentration C(N) of the last boxcar must be calculated in a different way than in the fixed boxcar train:

$C(N) = A(N)/(GAMMA - GCYCL)$

The cyclic development stage, GCYCL, is formulated in exactly the same way as G:

$GCYCL = INTGRL(GCYCLI, DEV)$

It is not equal to G, however, because GCYCL is modified in the subroutine SHIFT.

*The subroutine 'SHIFT'* This subroutine definition is placed between the CSMP labels STOP and ENDJOB.

The list of arguments transfers the values from the main program to the subroutine and vice versa. A separate DIMENSION declaration of A is necessary in the subroutine. With ample memory space available, over-dimensioning is convenient. The DO loop must be a backward one, because a forward loop would fill all array elements with the first one.

```
DO I = N, 2, -1
  A(I) = A(I - 1)
ENDDO
A(1) = A0
A0 = 0.
GCYCL = GCYCL - GAMMA
```

It is important that GCYCL is decreased by GAMMA, and not reset to zero. The shift is triggered when  $g'$  reaches  $\gamma$ , but this does not necessarily occur at an exact multiple of DELT. The excess value of  $g'$  above  $\gamma$  before the shift must be retained.

*The escalator boxcar train, applied to a demographic problem* The best method for solving demographic problems is the escalator boxcar train, because age, which is used as a characteristic, does not disperse. To illustrate its use, the same example will be given for the growth of the Dutch population, described by de Wit & Goudriaan (1978). For reasons of clarity, only the female proportion of the population is simulated, the male proportion being taken for granted. The ratio of boys to girls at birth (FRBOY), is used to calculate the fraction of girls (FRGIRL). The age dependence relating to relative death rate and relative birth rate is given in Figure 20. The corresponding fraction of survival (FS) is found by simulating a single cohort from birth onwards. Mathematically, the relative death rate (RDR) and the fraction survival (FS) are related by:

$$\text{RDR} = -(\text{d}(\text{FS})/\text{d}A)/\text{FS}$$

where A stands for age.

The listing of the CSMP program used is given in Figure 21. Data concerning the initial age distribution of the population are supplied. This is done by a TABLE specifying the contents of the twenty 5-year classes of the population array AI. Then, two FUNCTIONS with a list of coordinate points of the relationship between relative death and birth rates and age are supplied. In the INITIAL segment some computations are done for the discretization of age and development scale. This is necessary before the actual simulation in the DYNAMIC segment occurs. The simulation itself requires computation of the rates

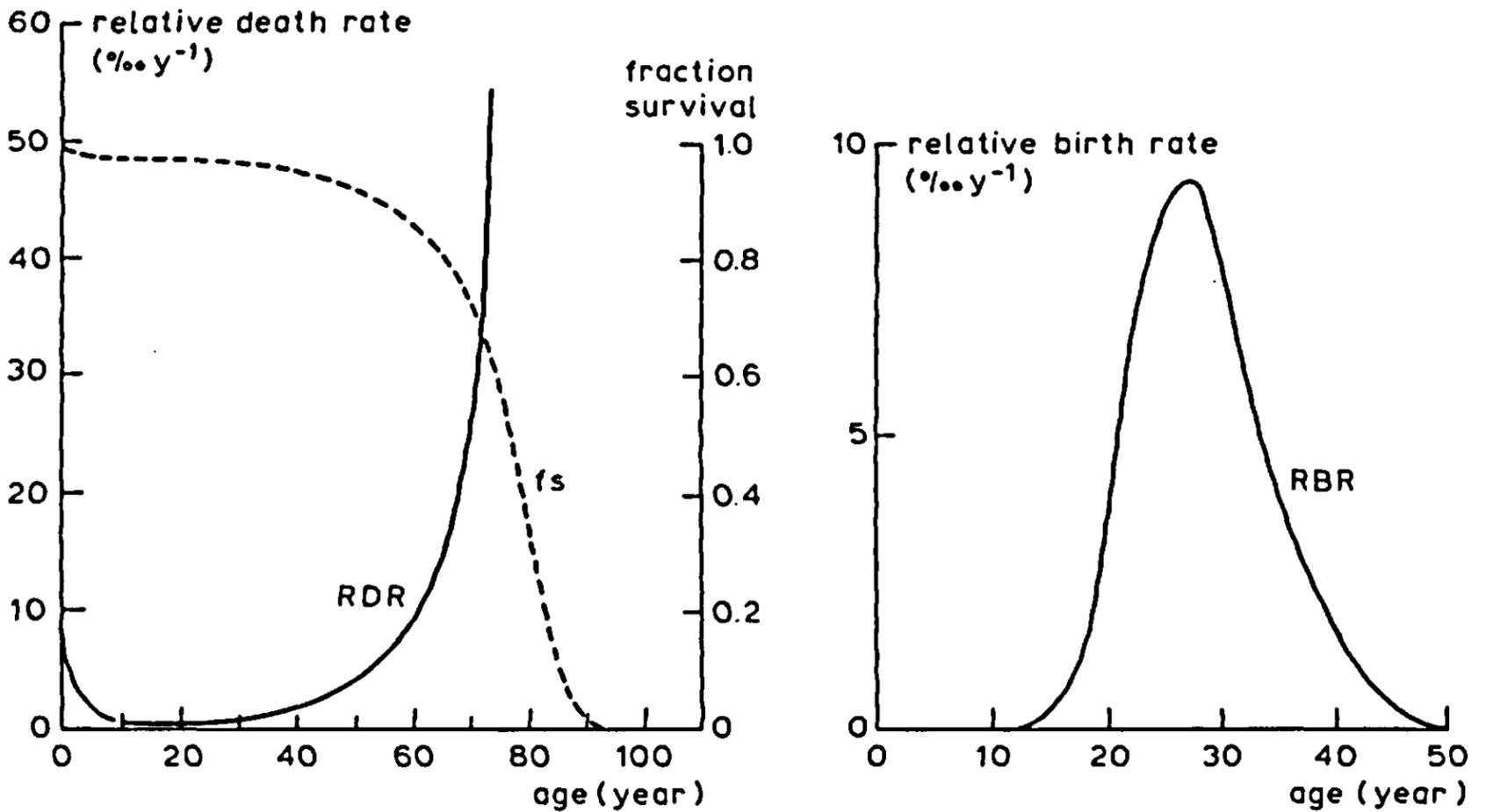


Figure 20. The age dependence of relative death rate (RDR) and of relative birth rate (RBR). The fraction survival (fs, dashed line) is a function of RDR.

Figure 21. Listing of the program of the model simulating the growth of the Netherlands population.

```

TITLE GROWTH OF THE NETHERLANDS POPULATION
STORAGE AGE(20),MORR(20),FRACA(20)
FIXED I,N

INITIAL
NOSORT
*   Relative death rate, in promille per year, as a function of age
    FUNCTION RMRTB= 0.,10., 2.5,4., 5.,1.8, 7.5,0.8, 10.,0.5, ...
                   15.,0.3, 20.,0.3, 30.,0.6, 40.,1.6, 50.,4.9, ...
                   60.,8.5, 65.,14., 70.,25., 75.,55., 82.5,180., ...
                   87.5,380., 92.5,760., 97.5,900., 105.,900.
*   Relative birth rate per year, as a function of age
    FUNCTION RBRTB= 0.,0., 12.5,0., 17.5,.02, 22.5,.137, 25.,.166,...
                   27.5,.188, 30.,.166, 32.5,.113, 37.5,.055, ...
                   42.5,.016, 47.5,.002, 50.,0., 105.,0.
*   Fraction young born boys
    PARAM FRBOY=0.512
*   100 years of age (AGETOT=ADP) is covered in 20 classes (N)
    PARAM AGETOT=100.,N=20
*   Residence time in one age class
    TC    =AGETOT/N
*   Development rate is standardized and constant
    DEVR  =1./AGETOT
*   Development width covered by one ageclass
    GAMMA =1./N
*   The fraction girl of the young borns
    FRGIRL=1.-FRBOY
* Calculation of the initial amounts

```

```

* Initial contents of ageclasses of 5 years wide, expressed in thousands
TABLE AI(1-20)=582.,587.,553.,543.,554.,420.,380.,381.,378., ...
                376.,330.,323.,298.,226.,226.,150.,70.,25.,13.,0.
* Initialization of integrals
ATOTI=0.
DO I=1,N
    AI(I)=AI(I)*1000.
    A(I) =AI(I)
    ATOTI=ATOTI + AI(I)
ENDDO
AO =0.
G =0.
GCYCL=0.

DYNAMIC
NOSORT
* Calculation of the states
* Development
G =INTGRL(0.,DEVR)
GCYCL =INTGRL(0.,DEVR)
* Calculation of the number of each boxcar
AO =INTGRL(0.,RA0)
A =INTGRL(AI,RA,20)
AOUT =INTGRL(0.,OUTFL)
* When GCYCL exceeds GAMMA, the shift is applied.
* In this situation, this is after every 5 years
PUSH =IMPULS(5.,5.)
IF(PUSH.EQ.1.) CALL SHIFT(N,GAMMA,GCYCL,A,AO)
* Calculation of the rates
* The total birth rate of girls is first set to zero
TBR =0.
**The zero's boxcar
* The age of the centre
AGE0 = 0.5 * GCYCL*AGETOT
* The mortality rate
MORRO =AO * 0.001*AFGEN(RMRTB,AGE0)
TMORT =MORRO
**The 1 - (N-1)'s boxcar
DO I=1,N-1
* The age of the centre of each boxcar
AGE(I) =TC*(I-0.5)+GCYCL*AGETOT
* Mortality rate of each class
MORR(I)=A(I)*0.001*AFGEN(RMRTB,AGE(I))
* Total mortality
TMORT =TMORT + MORR(I)
* The 1 - (N-1)'s class only change by death
RA(I) =-MORR(I)
* Total birth rate of girls
TBR =TBR + A(I) * AFGEN(RBRTB,AGE(I))*FRGIRL
ENDDO
**The N's boxcar
AGE(N) =TC*(N-0.5)+0.5*GCYCL*AGETOT
RMRN =0.001*AFGEN(RMRTB,AGE(N))
MORR(N)=A(N)*RMRN
TMORT =TMORT + MORR(N)
* The zero's class increases by birth, and decreases by death
RA0 =TBR - MORRO
* The N's boxcar decreases by death and by an outflow of people

```

```

* older than 100 years old.
  CN    =A(N)/(GAMMA-GCYCL)
  OUTFL =AMIN1(DEVR*CN*(1.-RMRN),A(N)/DELT)
  RA(N) =-OUTFL - MORR(N)
* The balance should be zero:
  ATOT  =A0
  DO I=1,N
    ATOT  =ATOT+A(I)
  ENDDO
  TOTMOR=INTGRL(0.,TMORT)
  TOTBIR=INTGRL(0.,TBR)
  BALANC=ATOT +TOTMOR + AOUT - ATOTI - TOTBIR
* Calculation of the age-distribution
  FRACAO=A0/ATOT
  DO I=1,N
    FRACA(I)=A(I)/ATOT
  ENDDO
* Time is expressed in years
  TIMER FINTIM=1000.,DELT=1.,PRDEL=100.
  PRINT A0,A(1-20),AOUT,ATOT,FRACAO,FRACA(1-20),TBR, TMORT, BALANC
  METHOD RECT
END
STOP
SUBROUTINE SHIFT(N,GAMMA,GCYCL,A,A0)
DIMENSION A(N)
DO I=N,2,-1
  A(I)=A(I-1)
ENDDO
A(1) =A0
A0   =0.
GCYCL=GCYCL-GAMMA
RETURN
END
ENDJOB

```

of change and, of course, the integration of these rates, which is done by the INTGRL statement. Also, whole population totals are computed by summation over all age classes in a regular FORTRAN DO loop. The data supplied for the FUNCTIONS are read by an AFGEN statement.

---

### Exercise 29

Find FS (Figure 20) by simulation. Make the birth rate constant, for instance 1000 per year, and after 100 years the age distribution will have the same shape as the fraction FS.

---

### Exercise 30

In the normal simulation, with the birth rate coupled to the number of fertile women, the population size will not be constant. After a long enough period of

simulation, exponential growth will result, with a corresponding age distribution and relative growth rate. To find these, run the model, as given here, for 1000 years. Compare the age distribution with FS found in Exercise 29.

---

*The fractional boxcar train (see Figure 22)* This boxcar train can be used when the relative dispersion of through-flow is known to vary during simulation, so both the desired delay and dispersion must be computed in the DYNAMIC segment of the main program. The rate of development,  $v$  or DEVR, often has a simpler relation with environmental conditions than the delay (germination period, longevity, etc.) itself:

$$\text{DEVR} = \text{AFGEN}(\text{DEVT}, \text{TEMP})$$

In this example, the rate of development is a function of temperature. The shape of the function must be specified in a separate list of paired numbers with the name DEVT. In a similar way, the relative dispersion  $\text{RD}$  ( $\sigma_{\text{total}}/T_{\text{total}}$ ) is a function of temperature:

$$\text{RD} = \text{AFGEN}(\text{RDT}, \text{TEMP})$$

The procedure of the fractional boxcar train is described in a subroutine called 'BOXCAR' which can be used as a separate module. In the main program, this subroutine is called:

```
A0,A,ATOT,MORFL,OUTFL,GAMMA,GCYCL = ...  
  BOXCAR(1,AI,DEVR,RD,RMR,INFL,N,DELT,TIME)
```

Inputs for this subroutine are: a number which serves to identify the boxcar train in the event of a computing error, the initial amount in the boxcars of the train ( $\text{AI}(1 - \text{N})$ ), the development rate (DEVR), the relative dispersion (RD), the relative mortality rate (RMR), the inflow into the zero boxcar (INFL), the number of boxcars in the train (N), DELT and TIME, respectively. These inputs must be given or calculated in the main program. Time is available by default. Outputs of this subroutine are the amount in the zero boxcar (A0), the amount in the other boxcars ( $\text{A}(1 - \text{N})$ ), the total amount in the boxcar train (ATOT), the total mortality rate in the boxcar train (MORFL), the outflow of the last ( $\text{N}^{\text{th}}$ ) boxcar (OUTFL), the development width (GAMMA) and the cyclic development stage (GCYCL). In the main program, memory storage should be reserved for two array variables AI and A, e.g. when N equals 10:

```
STORAGE AI(10), A(10)
```

N, the number of boxcars, should be declared integer

```
FIXED N
```

In the initial segment of the main program, the initial amount in the boxcar train and the number of boxcars should be given:

TABLE AI(1 - 10) = ....  
PARAM N = 10

In the subroutine 'BOXCAR', the boxcar train is initialized by calling subroutine 'BOXINI'. In this subroutine, A0 and GCYCL are set to zero, GAMMA is computed as  $1/N$  and  $A(1 - N)$  is initialized.

The fraction F for the fractional boxcar train is calculated by Equation 34, written with F on the left:

$$F = 1 - N * RD * RD$$

When this computed value of F is used, the fractional boxcar train will produce the desired dispersion.

Although this method is flexible, it cannot be stretched beyond its limits. An undue imbalance between the number of boxcars N and the desired RD will be revealed in nonsense values of F. Theoretically, it is immediately clear that F must lie between zero and unity. The upper end of this range is exceeded if the number of boxcars N is too small for the low RD that we wish to simulate. To select an appropriate value for N, it should be kept in mind that a minimum amount of dispersion cannot be avoided. Even when F is set at unity, some dispersion remains within the boxcar width. Usually this amount can be neglected.

The lower end of the range of F is zero, but a closer inspection indicates that F must also be larger than the fraction that would normally be transferred in each integration interval in the fixed boxcar train:  $DEL T * DEVR / GAMMA$ . If F is smaller than this value, the chosen number of boxcars N was too high. N must be smaller than 1 divided by the maximally occurring value of RD squared. About three-quarters of this maximum number is usually sufficient. If this does not solve the problems, it could mean that DELT is too large. It is good practice to include finish conditions for these sorts of criteria:

```
IF (DEL T.GT.(F * GAMMA/DEVR)) THEN  
  CALL EXIT  
ENDIF
```

Both time-interval of integration and residence-time in the boxcars of a boxcar train characterize the degree of temporal resolution required in the model. Therefore, there is little point in choosing  $\Delta t$  much smaller than the residence time  $\gamma/v$ . In this case, the general rule that  $\Delta t$  must not exceed one-tenth of the time coefficient of the fastest model component can be relaxed a little. The requirement here is less stringent, thanks to the negative feedback within the boxcar train: a numerical error in the rate of transition only affects the distribution among adjacent boxcars and has hardly any effect on removal from the boxcar train as a whole. The conclusion is that  $\Delta t$  should not exceed  $F\gamma/v$ , but may well be greater than  $1/10$  of  $\gamma/v$ .

If the program is aborted on account of one of these finish conditions, the programmer should first search for bugs in the data set such as AFGEN functions, or in other parts of the program before trying to repair the failure by adapting of N or DELT. Now that F has been adequately defined, it is used in the trigger for the shift:

```
IF (GCYCL.GE.F*GAMMA) CALL SHIFT...  
  (N, F, A0, A, GAMMA, GCYCL)
```

In the subroutine SHIFT, the list of arguments is repeated, and the array declaration follows:

```
DIMENSION A(N)  
A(N) = A(N) + A(N-1) * F  
DO I = N - 1, 2, -1  
  A(I) = A(I) * (1 - F) + A(I-1) * F  
ENDDO  
A(1) = A(1) * (1 - F) + A0  
A0 = 0.  
GCYCL = GCYCL - F * GAMMA
```

These lines suffice to implement the theory of Subsection 2.2.6. There are no further changes in comparison to the escalator boxcar train.

### 2.2.9 *A practical application using the fractional boxcar train*

The fractional boxcar train can be used to simulate the population development of insects. This will be illustrated with an example of an orchard tortricid moth *Pandemis heparana* (DENN. et SCHIFF.), which is an important pest in European apple orchards. Control of these insects mainly relies on the use of broad-spectrum insecticides, but since these also kill beneficial insects, research has been directed towards more specific insect growth-regulators. These prevent metamorphosis but are only effective if applied at the right time: at emergence of the last-instar larvae. The best application time can be well predicted using a simulation model of population development.

*P. heparana* usually has one generation per year. Adult female moths deposit their eggs in August. Second or third larval instars hibernate until the end of March when they become active; the sixth larval instar pupates in June. Adult moths are, therefore, found from June to August.

The model presented in Figure 22 is a simplified version of that described by de Reede & de Wilde (1986), which simulates the post-hibernation phenology of *P. heparana* (in 1982 in Wageningen, the Netherlands). Each development stage (diapause-stage, 3-5<sup>th</sup> stage larvae, 6<sup>th</sup> stage male and female larvae, and male and female pupae) is described by a fractional boxcar train, which mimics the mean delay and temporal dispersion. Experiments showed temperature to be the only important determinant of the development rate. Temperature is computed as

Figure 22. Listing of a program, including the fractional boxcar train.

```

TITLE DEVELOPMENT PANDEMIS HEPARANA 1982
STORAGE DIAINI(4) , DIA(4) , ...
      L3INI(10), L3(10), ...
      L4INI(8) , L4(8) , ...
      L5INI(3) , L5(3) , ...
      ML6INI(10), ML6(10), ...
      FL6INI(4) , FL6(4) , ...
      MPINI(10), MP(10), ...
      FPINI(10), FP(10)

INITIAL

*Initial numbers in boxcartrains
TABLE DIAINI(1-4) = 4*25.
TABLE L3INI(1-10)=10* 0.
TABLE L4INI(1-8) = 8* 0.
TABLE L5INI(1-3) = 3* 0.
TABLE ML6INI(1-10)=10* 0.
TABLE FL6INI(1-4) = 4* 0.
TABLE MPINI(1-10)=10* 0.
TABLE FPINI(1-10)=10* 0.

*Total initial amount of diapause larvae
TINDIA=100.

*Fraction male
PARAM SEXR=0.5

*Fraction DIA3/DIA2
PARAM G=0.7

PI=ATAN(1.)*4.

DYNAMIC

*Temperature
TEMP =AVTEMP + AMPTMP*(-COS(2.*PI*TIME))
AVTEMP=0.5 * (MAXT+MINT)
AMPTMP=0.5 * (MAXT-MINT)
MAXT  =AFGEN(MXTT,DAY)
MINT  =AFGEN(MNTT,DAY)

DAY   =STDAY + TIME
PARAM STDAY= 32.

*Relative Mortality Rate
PARAMETER RMRL3 =0., RMRL4=0., RMRL5=0., RMRML6=0., ...
      RMRFL6=0., RMRDIA=0., RMRMP=0., RMRFP =0.

*Development Rate
DRDIA =AFGEN(DRDIAT,TEMP)
DRL3  =AFGEN(DRL3T ,TEMP)
DRL4  =AFGEN(DRL4T ,TEMP)
DRL5  =AFGEN(DRL5T ,TEMP)
DRML6 =AFGEN(DRML6T,TEMP)
DRFL6 =AFGEN(DRFL6T,TEMP)
DRMP  =AFGEN(DRMPT ,TEMP)
DRFP  =AFGEN(DRFPT ,TEMP)

```

\*Relative Dispersion

RDDIA =AFGEN(RDDIAT,TEMP)  
RDL3 =AFGEN(RDL3T ,TEMP)  
RDL4 =AFGEN(RDL4T ,TEMP)  
RDL5 =AFGEN(RDL5T ,TEMP)  
RDML6 =AFGEN(RDML6T,TEMP)

RDFL6 =AFGEN(RDFL6T,TEMP)  
RDMP =AFGEN(RDMPT ,TEMP)  
RDFP =AFGEN(RDFPT ,TEMP)

\*Subsequent boxcar calls must be done in a nosort section

NOSORT

PIDIA=0.

DIA0,DIA,DIATOT,MRDIA,PDDIA,GAMMA1,GCYCL1=...  
BOXCAR(1,DIAINI,DRDIA,RDDIA,RMRDIA,PIDIA , 4,DELT,TIME)

PIL3=(1.-G)\*PDDIA  
L30, L3, L3TOT, MRL3, PDL3, GAMMA2,GCYCL2=...  
BOXCAR(2, L3INI, DRL3, RDL3, RMRL3, PIL3 ,10,DELT,TIME)

PIL4=G \*PDDIA + PDL3  
L40, L4, L4TOT, MRL4, PDL4, GAMMA3,GCYCL3=...  
BOXCAR(3, L4INI, DRL4, RDL4, RMRL4, PIL4 , 8,DELT,TIME)

PIL5=PDL4  
L50, L5, L5TOT, MRL5, PDL5, GAMMA4,GCYCL4=...  
BOXCAR(4, L5INI, DRL5, RDL5, RMRL5, PIL5 , 3,DELT,TIME)

PIML6=SEXR \* PDL5  
ML60,ML6,ML6TOT,MRML6,PDML6,GAMMA5,GCYCL5=...  
BOXCAR(5,ML6INI,DRML6,RDML6,RMRML6,PIML6,10,DELT,TIME)

PIFL6=(1.-SEXR) \* PDL5  
FL60,FL6,FL6TOT,MRFL6,PDFL6,GAMMA6,GCYCL6=...  
BOXCAR(6,FL6INI,DRFL6,RDFL6,RMRFL6,PIFL6 ,4,DELT,TIME)

PIMP=PDML6  
MPO, MP, MPTOT, MRMP, PDMP, GAMMA7,GCYCL7=...  
BOXCAR(7, MPINI, DRMP, RDMP, RMRMP, PIMP ,10,DELT,TIME)

PIFP=PDFL6  
FPO, FP, FPTOT, MRFP, PDFP, GAMMA8,GCYCL8=...  
BOXCAR(8, FPINI, DRFP, RDFP, RMRFP, PIFP ,10,DELT,TIME)

SORT

CUML3 =INTGRL(0.,PIL3)  
CUML4 =INTGRL(0.,PIL4)  
CUML5 =INTGRL(0.,PIL5)  
CUMML6=INTGRL(0.,PIML6)  
CUMFL6=INTGRL(0.,PIFL6)  
CUMMP =INTGRL(0.,PIMP)  
CUMFP =INTGRL(0.,PIFP)  
CUMMM =INTGRL(0.,PDMP)  
CUMFM =INTGRL(0.,PDFP)

CUPL3 =100.\*CUML3/(TINDIA\*(1.-G))  
CUPL4 =100.\*CUML4/TINDIA  
CUPL5 =100.\*CUML5/TINDIA  
CUPML6=100.\*CUMML6/(TINDIA\*SEXR)  
CUPFL6=100.\*CUMFL6/(TINDIA\*(1.-SEXR))  
CUPMP =100.\*CUMMP/(TINDIA\*SEXR)  
CUPFP =100.\*CUMFP/(TINDIA\*(1.-SEXR))  
CUPMM =100.\*CUMMM/(TINDIA\*SEXR)  
CUPFM =100.\*CUMFM/(TINDIA\*(1.-SEXR))

\* Balance should be zero:

TMORR=MRDIA + MRL3 + MRL4 + MRL5 + MRML6 + MRFL6 + MRMP + MRFP  
ATOT =DIATOT+ L3TOT+L4TOT +L5TOT + ML6TOT+ FL6TOT+ MPTOT+FPTOT ...  
+ CUMMM + CUMFM  
BALANC=ATOT-TINDIA+INTGRL(0.,TMORR-PIDIA)

PRINT DAY,DIATOT,CUPL3,CUPL4,CUPL5,CUPML6,CUPFL6,...  
CUPMP,CUPFP,CUPMM,CUPFM,BALANC  
TIMER FINTIM=220.,DELT=0.0417,PRDEL=2.  
METHOD RECT

\*Development Rate

FUNCTION DRDIAT = -10.,0.0000, 6.0,0.0000, 11.,0.0181, 13.,0.0251,...  
16.,0.0412, 19.,0.0621, 22.,0.0831, 25.,0.1015,...  
35.,0.1015  
FUNCTION DRL3T = -10.,0.0000, 9.4,0.0000, 16.,0.1093, 25.,0.2585,...  
35.,0.2585  
FUNCTION DRL4T = -10.,0.0000, 8.0,0.0000, 16.,0.1155, 25.,0.2460,...  
35.,0.2460  
FUNCTION DRL5T = -10.,0.0000, 7.2,0.0000, 16.,0.1088, 25.,0.2204,...  
35.,0.2204  
FUNCTION DRML6T = -10.,0.0000, 6.3,0.0000, 16.,0.0704, 25.,0.1355,...  
35.,0.1355  
FUNCTION DRFL6T = -10.,0.0000, 7.8,0.0000, 16.,0.0594, 25.,0.1246,...  
35.,0.1246  
FUNCTION DRMPT = -10.,0.0000, 8.2,0.0000, 16.,0.0551, 25.,0.1190,...  
35.,0.1190  
FUNCTION DRFPT = -10.,0.0000, 8.2,0.0000, 16.,0.0551, 25.,0.1190,...  
35.,0.1190

\*Relative Dispersion

FUNCTION RDDIAT = -10.,0.00, 6.0,0.00, 11.,0.23, 13.,0.32, 16.,0.18,...  
19.,0.45, 22.,0.20, 25.,0.24, 35.,0.24  
FUNCTION RDL3T = -10.,0.00, 9.4,0.00, 13.,0.04, 16.,0.11, 19.,0.11,...  
22.,0.10, 25.,0.13, 35.,0.13  
FUNCTION RDL4T = -10.,0.00, 8.0,0.00, 13.,0.30, 16.,0.19, 19.,0.18,...  
22.,0.24, 25.,0.30, 35.,0.30  
FUNCTION RDL5T = -10.,0.00, 7.2,0.00, 11.,0.23, 13.,0.57, 16.,0.15,...  
19.,0.23, 22.,0.11, 25.,0.13, 35.,0.13  
FUNCTION RDML6T = -10.,0.00, 6.3,0.00, 11.,0.25, 13.,0.16, 16.,0.14,...  
19.,0.14, 22.,0.07, 25.,0.08, 35.,0.08  
FUNCTION RDFL6T = -10.,0.00, 7.8,0.00, 11.,0.25, 13.,0.06, 16.,0.49,...  
19.,0.10, 22.,0.13, 25.,0.16, 35.,0.16  
FUNCTION RDMPT = -10.,0.00, 8.2,0.00, 11.,0.03, 13.,0.06, 16.,0.08,...  
19.,0.06, 22.,0.22, 25.,0.26, 35.,0.26  
FUNCTION RDFPT = -10.,0.00, 8.2,0.00, 11.,0.03, 13.,0.06, 16.,0.08,...  
19.,0.06, 22.,0.22, 25.,0.26, 35.,0.26

\*Maximum daily temperature

FUNCTION MXTT= 1., 7., 2., 11., 3., 11.6, 4., 11.4, 5., 9.8 , ...WAG1982  
etc. weatherdata Wageningen, 1982

\*Minimum daily temperature

FUNCTION MNTT= 1., 2.1, 2., 4.5, 3., 6.4, 4., 7.4, 5., 0.2, ...WAG1982  
etc. weatherdata Wageningen, 1982

END  
STOP

```

C*****C
  SUBROUTINE BOXCAR(COUNT, AI, DEVR, RD, RMR, INFL, N, DELT, TIME,
    $              AO, A, ATOT, MORFL, OUTFL, GAMMA, GCYCL)
C*****C
C   To use this subroutine, memory storage has to be reserved for
C   the initial- and actual values of each boxcar of a particular boxcar
C   train by typing in the INITIAL-part: STORAGE AI(N), A(N)
C   in which AI and A are the names of the arrays of that boxcar train.
C   For N, the total number of boxcars in that boxcar train has to be
C   substituted. N is an integer, which has to be declared by: FIXED N
C   The initial conditions of AI can be given by means of a
C   TABLE statement: TABLE AI(1-N)=..., ..., or can be calculated in
C   a DO loop.
C   N has to be calculated as a function of the residence time and
C   its standard deviation and has to be given as a parameter in the
C   main program. For a fractional boxcar train:
C       N < minimum of 1/RD**2
C   Usually 3/4 * minimum of 1/RD**2 is taken. (In practice a value of
C   N=4 usually seems to mimick delay and dispersion very well).

  IMPLICIT REAL(A-Z)
  INTEGER      I, N, COUNT
  DIMENSION    AI(N), A(N), MORR(50)

C-----C
C   Initiation of the boxcar train                               C
C-----C
  IF (TIME.EQ.0.) CALL BOXINI(AI, N, AO, A, GAMMA, GCYCL)

C-----C
C   Calculation of fraction F                                     C
C-----C
  CALL FRACT(COUNT, DEVR, RD, N, DELT, GAMMA, F)

C-----C
C   Calculation of the rates                                     C
C-----C
  IF (TIME.EQ.0.) GO TO 10
C-----the rate of inflow (INFL) is given or calculated in the main program
C-----mortality rate (MORR) and total mortality flow (MORFL)
  MORRO = RMR * AO
  MORFL = MORRO
  DO I=1, N
    MORR(I) = RMR * A(I)
    MORFL = MORFL + MORR(I)
  ENDDO

C-----the rate of outflow (OUTFL) is calculated
C-----note the outflow is also subject to mortality
  CN = A(N) / (GAMMA - GCYCL)
  OUTFL = DEVR * CN * (1. - RMR * DELT)

C-----C
C   Calculation of the states (integrals)                         C
C-----C
C-----development
  GCYCL = GCYCL + DEVR * DELT

```

```

C-----amount in each boxcar (A), after mortality flow and
C-----inflow and outflow in respectively AO and A(N)
      AO =AO - MORRO*DELT + INFL*DELT
      DO I=1,N-1
          A(I) =A(I) - MORR(I)*DELT
      ENDDO
      A(N) =A(N) -MORR(N)*DELT - OUTFL*DELT

C-----amount in each boxcar (A), after shift (discontinuous process)
      IF(GCYCL.GE.F*GAMMA) CALL SHIFT(N,F,AO,A,GAMMA,GCYCL)

10  CONTINUE

C-----total amount in boxcar train (ATOT)
      ATOT= AO
      DO I=1,N
          ATOT =ATOT + A(I)
      ENDDO

      RETURN
      END

C*****C
      SUBROUTINE BOXINI(AI,N,AO,A,GAMMA,GCYCL)
C*****C
      IMPLICIT REAL(A-Z)
      INTEGER      I,N
      DIMENSION    AI(N),A(N)

      GCYCL =0.
      GAMMA =1./FLOAT(N)

      AO =0.
      DO I=1,N
          A(I) = AI(I)
      ENDDO

      RETURN
      END

C*****C
      SUBROUTINE FRACT(COUNT,DEVR,RD,N,DELT,GAMMA,F)
C*****C
      IMPLICIT REAL(A-Z)
      INTEGER      N,COUNT

      F = 1. - N * RD * RD

C-----DELT has to be smaller than a fraction F of the smallest time coefficient
C of one boxcar
      IF (DELT.GT.(F*GAMMA/(DEVR+1.E-10))) THEN
          WRITE (6,'(A,I2,A)') ' Delt too large for boxcar no: ',COUNT
          $           , ' or too many boxes N: F too small '
          CALL EXIT
      ENDIF

      RETURN
      END

```

```

C*****C
  SUBROUTINE SHIFT(N,F,AO,A,GAMMA,GCYCL)
C*****C
  IMPLICIT REAL(A-Z)
  INTEGER      I,N
  DIMENSION   A(N)

  A(N) = A(N) + A(N-1)*F
  DO I=N-1,2,-1
    A(I)=A(I)*(1.-F) + A(I-1)*F
  ENDDO
  A(1)=A(1)*(1.-F) + AO
  AO = 0.

  GCYCL =GCYCL - F*GAMMA

  RETURN
  END
C-----C
ENDJOB

```

a simple sinusoidal curve through the daily maximum and minimum temperatures, which were measured in a Stevenson screen, 1.5 m high. The time step for numerical integration is one hour ( $1/24 = 0.0417$  day), to allow for diurnal fluctuations in temperature. The model is started by initializing the total number of diapause larvae entering the first boxcar trains; the numbers of all other stages are set to zero. Input data are the development rates and relative dispersions of the various stages at different temperatures, the relative mortality rate, the sex ratio, the initial ratio of the numbers of  $L_3$  and  $L_2$  larvae in diapause, the startday and the minimum and maximum daily temperatures. For each development stage (except the adult stage), the subroutine BOXCAR is called at each time step. The inputs for this subroutine are: the initial content of each boxcar, which is an array variable (...INI); the development rate (DR...); the relative dispersion (RD...); the relative mortality rate (RMR...); the inflow into the boxcar train (PI...); the number of boxcars (N); DELT and TIME.

The outputs of the BOXCAR subroutine are: the content of the zero boxcar (...0); the content of the other boxcars, which is an array variable (e.g. L3(I) in which I is the number of the boxcar from 1 – N); the total content of the boxcar train (...TOT), which equals the sum of the contents of each boxcar (from 0 – N); the total mortality rate (MR...); the outflow of the boxcar train (PD...); the cyclic development stage (GCYCL.) and the width of the boxcar in development units (GAMMA.).

Figure 23 compares the simulation results with field sample data on  $L_5$ ,  $L_6$ , pupae and adult moths (in which the sexes are combined). From this figure, it is clear that phenology can be simulated well, using the fractional boxcar train when the temperature fluctuates.

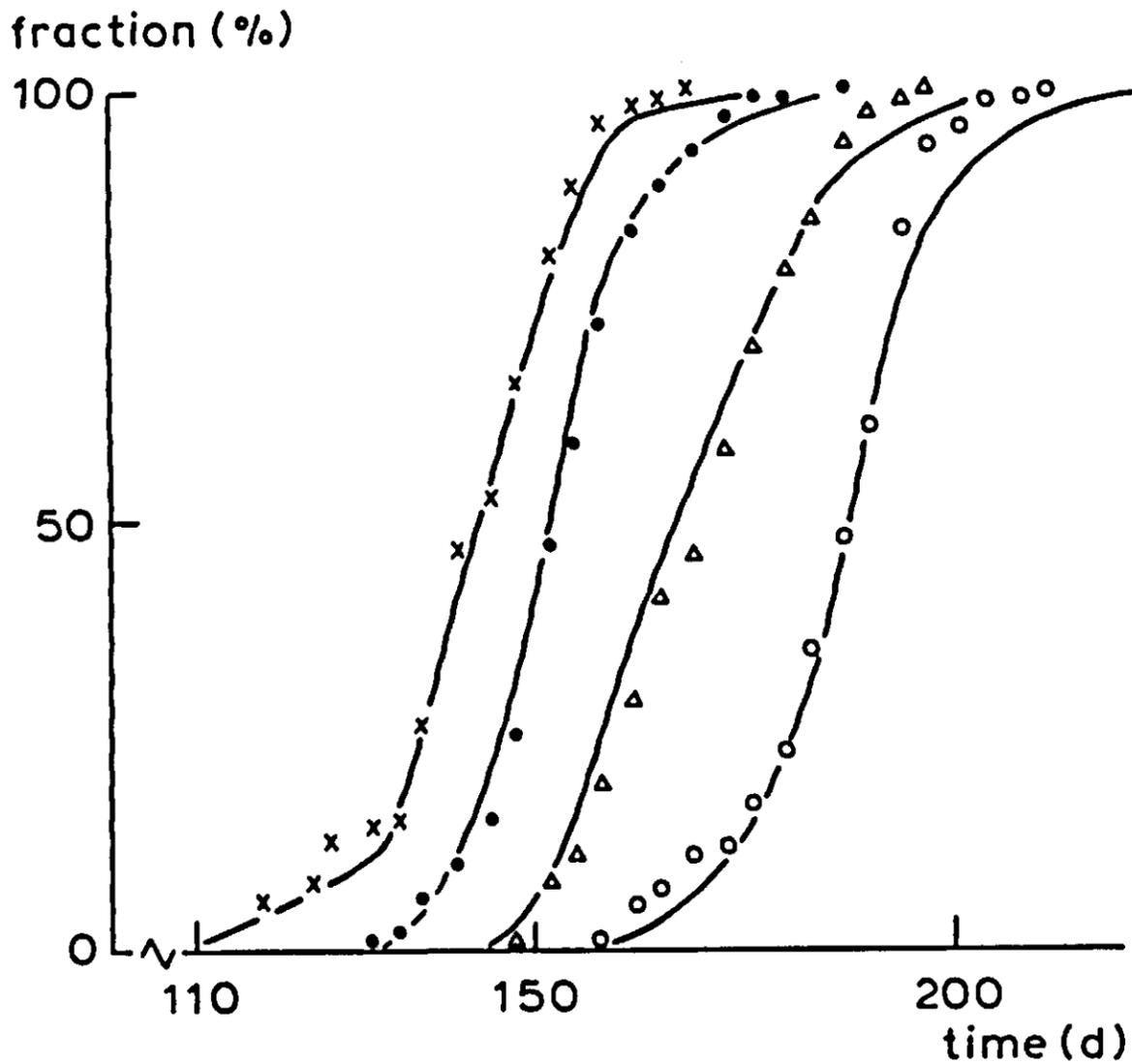


Figure 23. Simulation (—) and measured phenology of  $L_5$  (x),  $L_6$  (●), pupae ( $\Delta$ ) and adult moths ( $\circ$ ) of *Pandemis heparana*.

#### 2.2.10 Effect of the type of boxcar train on population growth

With the fractional boxcar train being intermediate, the greatest difference can be expected between the escalator boxcar train and the fixed boxcar train types. To evaluate the effect of the chosen method, the demographic example (Figures 20 and 21) was used. To establish a control, the fixed boxcar train was also run with 1-year classes, to find an accurate estimate of the equilibrium age distribution and of the relative growth rate corresponding to the used mortality and fertility distributions. This equilibrium age distribution was then used as input for both the fixed and the escalator boxcar trains with 5-year classes. The 1-year control yielded an equilibrium relative growth rate of  $8.29 \cdot 10^{-3} \text{ yr}^{-1}$ , which was closely approximated by the 5-year escalator boxcar train ( $8.27 \cdot 10^{-3} \text{ yr}^{-1}$ ). The 5-year fixed boxcar train resulted in a lower simulated relative growth rate:  $7.24 \cdot 10^{-3} \text{ yr}^{-1}$ . The explanation for this underestimation is the numerical dispersion in the fixed boxcar train method. In a growing population, as simulated here, the younger age groups will contain many more individuals, so that the numerical dispersion will cause an apparent artificial 'ageing', slowing down the simulated population growth.

The price for the high accuracy of the escalator boxcar train is a slight irregularity in the simulated relative growth rate (RGR) following a 5-year cycle around the mean value of  $8.27 \cdot 10^{-3} \text{ yr}^{-1}$ , and ranging between 7.97 and 8.59

$10^{-3} \text{ yr}^{-1}$ . This irregularity is caused by the sawtooth ageing tendency of the 5-year classes.

### 2.2.11 Discussion

Clearly, the fixed boxcar train method is the simplest and, if possible, should be preferred for that very reason. As shown in the demographic example, it is much easier to use 1-year classes in combination with a time interval of integration of one year than to use the escalator boxcar train with 5-year classes. Simplicity of program formulation is then bought for computer time, which is often a profitable deal. In both methods, numerical dispersion is avoided.

Compelling reasons for using the more complicated boxcar train versions may be found in the desire to simulate fluctuating rates of development and dispersion. However, one must still remain aware of the unpleasant fact that dispersion is simply mimicked by a numerical tool, and that it is not really simulated from the underlying processes. This should stimulate further research so that the true reasons for the dispersion can be explained.

The merit of the methods presented here is that they enable effects of the observed characteristics of population dynamics in other situations to be evaluated, and can also serve as a tool for prognostic and management purposes.