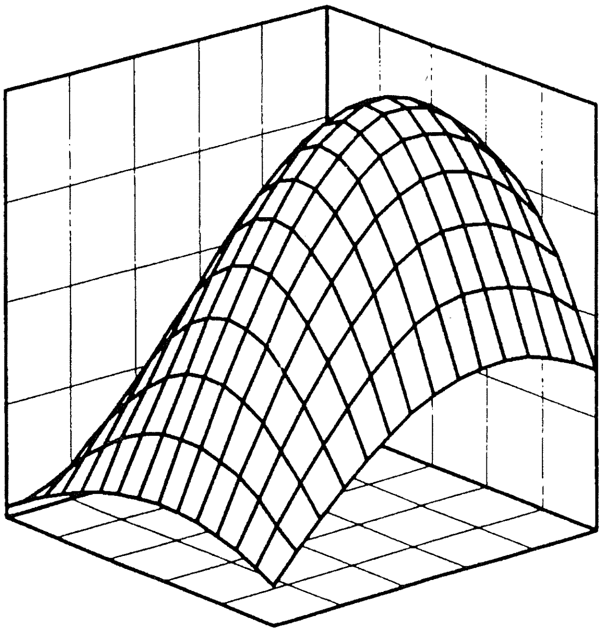
Description, parameterisation and user guide of BACSIM

An epidemiological model of the control of beet armyworm, *Spodoptera exigua*, with baculoviruses



PRODUCTION ECOLOGY

PLANT RESEARCH INTERNATIONAL

BIBLIOGRAPHIC DATA

F.J.J.A. Bianchi & W. van der Werf

Description, parameterisation and user guide of BACSIM
F.J.J.A. Bianchi and W. van der Werf. - Wageningen :
Plant Research International;
Wageningen : The C.T. de Wit Graduate School for
Production Ecology. - (Quantitative approaches in systems
analysis ; no. 22)
NUGI 835
Subject headings: baculoviruses, biological control

Keywords

baculoviruses, biological control

Guidelines 'Quantitative Approaches in Systems Analysis'

Manuscripts or suggestions should be submitted to the editorial board (H.F.M. ten Berge, Plant Research International, or M.K. van Ittersum, TPE-WAU). The final version of the manuscripts should be delivered to the editors camera-ready for reproduction. The submission letter should indicate the scope and aim of the manuscript (e.g. to support scientific publications in journals, program manual, educational purposes). The costs of printing and mailing are borne by the authors.

The English language is preferred. Authors are responsible for correct language and lay-out. Overall guidelines for the format of the texts, figures and graphs can be obtained from the publication editor at Plant Research International, or from the PE office:

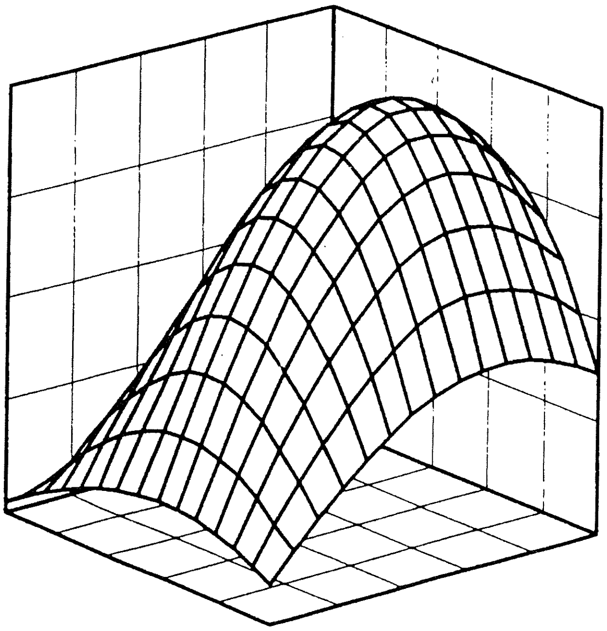
H. Terburg
Plant Research International

P.O. Box 14
NL-6700 AA Wageningen
Phone: (+) 31 317.475723
Fax: (+) 31 317.418094
E-mail: h.terburg@plant.wag-ur.nl

Th.H. Jetten
Secretariat C.T. de Wit Graduate School
for Production Ecology
Lawickse Allee 13
NL-6701 AN Wageningen
Phone: (+) 31 317.485116
Fax: (+) 31 317.484855
E-mail: theo.jetten@beleid.spp.wau.nl

Description, parameterisation and user guide of BACSIM

An epidemiological model of the control of beet armyworm, *Spodoptera exigua*, with baculoviruses



PRODUCTION ECOLOGY

PLANT RESEARCH INTERNATIONAL

Quantitative Approaches in Systems Analysis

The Quantitative Approaches in Systems Analysis series provides a platform for publication and documentation of simulation models, optimization programs, Geographic Information Systems (GIS), expert systems, data bases, and utilities for the quantitative analysis of agricultural and environmental systems. The series enables staff members, students and visitors of Plant Research International and PE to publish, beyond the constraints of refereed journal articles, updates of models, extensive data sets used for validation and background material to journal articles. The QASA series thus primarily serves to support peer reviewed articles published elsewhere. The inclusion of listings of programs in an appendix is encouraged.

All manuscript are reviewed by an editorial board comprising one Plant Research International and one PE staff member. The editorial board may consult external reviewers. The review process includes assessing the following: relevance of the topic to the series, overall scientific soundness, clear structure and presentation, and completeness of the presented material(s). The editorial board evaluates manuscripts on language and lay-out matters in a general sense. However, the sole responsibility for the contents of the reports, the use of correct language and lay-out rests with the authors. Manuscripts or suggestions should be submitted to the editorial board. Reports of the series are available on request.

Quantitative Approaches in Systems Analysis are issued by the Plant Research International and The C.T. de Wit Graduate School for Production Ecology (PE).
Plant Research International carries out research into areas such as plant physiology, soil science and agro-ecology with the aim of improving the quality of soils and agricultural produce and of furthering sustainable production systems.
The 'Production Ecology' Graduate School explores options for crop production systems associated with sustainable land use and natural resource management; its activities comprise research on crop production and protection, soil management, and cropping and farming systems.

Address for ordering copies of volumes in the series:
Secretariat
TPE-WAU
Bornsesteeg 47
NL-6708 PD Wageningen
Phone: (+) 31 317.482141
Fax: (+) 31 317.484892
E-mail: office@sec.tpe.wau.nl

Description, parameterisation and user guide of BACSIM

An epidemiological model of the control of beet armyworm, *Spodoptera exigua*, with baculoviruses

F.J.J.A. Bianchi & W. van der Werf

PE Plant Research International

Quantitative Approaches
in Systems Analysis No. 22
February 2001

BIBLIOGRAPHIC DATA

F.J.J.A. Bianchi & W. van der Werf

Description, parameterisation and user guide of BACSIM
F.J.J.A. Bianchi and W. van der Werf. - Wageningen :
Plant Research International;
Wageningen : The C.T. de Wit Graduate School for
Production Ecology. - (Quantitative approaches in systems
analysis ; no. 22)
NUGI 835
Subject headings: baculoviruses, biological control

Keywords

baculoviruses, biological control

Guidelines 'Quantitative Approaches in Systems Analysis'

Manuscripts or suggestions should be submitted to the editorial board (H.F.M. ten Berge, Plant Research International, or M.K. van Ittersum, TPE-WAU). The final version of the manuscripts should be delivered to the editors camera-ready for reproduction. The submission letter should indicate the scope and aim of the manuscript (e.g. to support scientific publications in journals, program manual, educational purposes). The costs of printing and mailing are borne by the authors.

The English language is preferred. Authors are responsible for correct language and lay-out. Overall guidelines for the format of the texts, figures and graphs can be obtained from the publication editor at Plant Research International, or from the PE office:

H. Terburg
Plant Research International

P.O. Box 14
NL-6700 AA Wageningen
Phone: (+) 31 317.475723
Fax: (+) 31 317.418094
E-mail: h.terburg@plant.wag-ur.nl

Th.H. Jetten
Secretariat C.T. de Wit Graduate School
for Production Ecology
Lawickse Allee 13
NL-6701 AN Wageningen
Phone: (+) 31 317.485116
Fax: (+) 31 317.484855
E-mail: theo.jetten@beleid.spp.wau.nl

Table of Contents

Page

Samenvatting

Baculovirussen zijn van nature voorkomende pathogenen die insectenpopulaties kunnen reguleren en worden in de landbouw als biologisch bestrijdingsmiddel gebruikt. Doordat baculovirussen erg infectieus zijn en in de meeste gevallen specifiek zijn voor slechts enkele insectensoorten vormen baculovirussen een milieuvriendelijk alternatief voor chemische bestrijdingsmiddelen. Hoewel plaaginsecten effectief met baculovirussen bestreden kunnen worden, is de relatief trage werkingstijd van deze pathogenen een nadeel ten opzichte van sneller werkende chemische bestrijdingsmiddelen. Door middel van genetische modificatie is een aantal baculovirusrecombinanten gemaakt waarvan de werkingstijd met 25 tot 50% is teruggebracht ten opzichte van niet gemodificeerde baculovirussen. Er is op dit moment nog niet veel bekend over de effectiviteit van deze recombinanten en is het nog onduidelijk of het gebruik van deze recombinanten wenselijk is.

Een ruimtelijk expliciet model (BACSIM) is ontwikkeld voor de simulatie van de biologische bestrijding van floridamotlarven met baculovirussen in kaschrysant. BACSIM is gebaseerd op een gedetailleerde beschrijving van het gedrag en populatie-dynamica van de floridamot, gewasgroeikarakteristieken van chrysant, virusbesuitingen, de baculovirusinfectie-cyclus en transmissieroutes. Het model kan gebruikt worden om de populatie-dynamica van *S. exigua* en baculovirussen te simuleren onder varierende omstandigheden. Deze informatie kan gebruikt worden voor de evaluatie van de effectiviteit van baculovirussen en bespuitings-regimes. Bovendien is kwantitatieve informatie over de populatie-dynamica van genetisch gemodificeerde baculovirussen relevant voor het inschatten van risico's bij het gebruik van deze virussen.

Dit rapport geeft een uitgebreide documentatie van het model BACSIM met een beschrijving van het model, de kwantificering van modelparameters, richtlijnen voor het gebruik van het model en een complete listing van BACSIM. Het doel van dit rapport is het gebruik van BACSIM voor andere gebruikers te vergemakkelijken.

Summary

Baculoviruses are naturally occurring insect pathogens that are used as biological control agents in forestry and agriculture. Their infectivity, specificity and safety to non-target organisms make them promising candidates to replace chemical insecticides. Although the baculoviruses are in general effective biological insecticides, their relatively slow speed of kill limit their use in practice. Recently, recombinant baculoviruses have been constructed with reductions of 25 to 50% in time to kill as compared to the wild-type viruses. However, little is known about the efficacy of these recombinant baculoviruses in the field.

A spatially explicit simulation model (BACSIM) has been developed that simulates the epidemiology and efficacy of baculoviruses in greenhouses. The model is based on a detailed quantitative description of the behaviour and population dynamics of insects, plant growth characteristics, spray deposition, the baculovirus infection cycle and transmission routes. The model can be used to simulate the population dynamics of *S. exigua* and (genetically modified) baculoviruses under various conditions. This information can be used for the evaluation of the agricultural effectiveness of baculoviruses and spraying regimes. In addition, quantitative information of the population dynamics of genetically modified baculoviruses may be relevant for the assessment of the risks associated with the release of these baculoviruses in greenhouses.

This report gives a detailed documentation of BACSIM, including a model description, the quantification of model parameters, practical guidelines for the use of the model and the complete listing of the model. Objective of the report is to facilitate the use of the model for other users.

1 Introduction

1.1 Objective and outline of the report

Baculoviruses are pathogens that can cause lethal infections in many insect species. In nature they have large impacts on the dynamics of insect populations, and are used as biological control agents of insect pests in agriculture and forestry (Moscardi, 1999; Black *et al.*, 1997). There is growing commercial interest in baculoviruses as biological control agents. They are target-specific and not harmful for non-target insects, unlike most conventional pesticides. For insect species that have developed resistance to insecticides, such as the beet armyworm *S. exigua*, the use of baculoviruses is one of the few options left for control (Brewer and Tumble, 1989).

A major drawback of the use of baculoviruses as crop protection agents is their high production costs, relatively slow speed of action, narrow host range and high inactivation rate compared to chemical insecticides (Moscardi, 1999). Recently, numerous recombinant baculoviruses have been constructed with improved insecticidal properties (Van Beek and Hughes, 1998). In addition, chemical synergists such as optical brighteners have been developed that improve baculovirus efficacy as biocontrol agents (Dougherty *et al.*, 1992). The determination of the commercial potential of new recombinant baculoviruses and formulations involve costly field-testing programs. In the case of recombinant baculoviruses, these experiments are subject to strong regulations and require costly facilities and procedures.

Computer simulation of baculovirus applications, insect and baculovirus population dynamics may be used to assess the agricultural effectiveness of viruses or formulations with altered insecticidal properties. Computer simulation may be used for the first screening of the agricultural effectiveness of baculoviruses, as a tool to set up field experiments, and may reduce the number of field experiments when model simulations are in line with field experiments. In addition, simulations can assist in the determination of effective spraying regimes of baculoviruses in varying crop development stages and at different temperatures for cultivation practice (de Moed *et al.*, 1990; van der Werf *et al.*, 1991). Finally, simulation models can be used to gain insight in the insect-baculovirus-crop system and the relative importance of viral characteristics for crop protection purposes. This knowledge may be used to direct future efforts for genetic modification to obtain baculoviruses with improved insecticidal characteristics.

This report describes a spatially explicit simulation model that simulates the epidemiology and spread of baculoviruses in greenhouses. The model is based on a detailed quantitative description of the population dynamics of insects, plant growth characteristics, spray deposition, the baculovirus infection cycle and transmission routes. The model can be used to simulate the population dynamics of *S. exigua* and (genetically modified) baculoviruses under various conditions. This information can be used for the evaluation of the agricultural effectiveness of baculoviruses and spraying regimes. In addition, quantitative information of the population dynamics of genetically modified baculoviruses may be relevant for the assessment of the risks associated with the release of these baculoviruses in greenhouses.

The model presented here builds on preliminary concepts presented by de Moed *et al.* (1990) and van der Werf *et al.* (1991). The new model was rebuilt from scratch in order to include an individual-based treatment of the encounter process between healthy and virus-killed cadavers in infestation patches, resulting in horizontal virus transmission. Parameters were re-estimated on the basis of new experimental results, e.g. with respect to the effect of host plant on virus infectivity and insect development rate.

The objective of the report is to give a detailed documentation of the model BACSIM so that it can be used properly or further developed by others. In chapter 2 the model concept and the consequences for the coding are described. Chapter 3 gives a description of the model. Chapter 4 describes the quantification of model parameters. Chapter 5 gives practical guidelines for using the model and in chapter 6 the model implementation is outlined. The listing of the input files containing the parameter values are given in Appendix I, acronyms are presented in Appendix II and the listing of the model is given in Appendix III.

1.2 The insect-pathogen system

1.2.1 Life cycle and development of *S. exigua*

S. exigua is a lepidopterous species belonging to the family Noctuidae. Originating from Southeast Asia, *S. exigua* is reported to be a pest in the USA, Mexico, Spain, Thailand and The Netherlands (Moscardi, 1999; Kolodny-Hirsch *et al.*, 1993; Alvaraso-Rodriguez, 1987; Caballero *et al.,* 1992; Kolodny-Hirsch *et al.*, 1997; Smits, 1987). In the Netherlands *S. exigua* occurs in greenhouses where climatic conditions are favourable and crops are cultivated throughout the year (Smits, 1987). Here, infestations are mainly found in chrysanthemum and gerbera but recently also in paprika. The life cycle of *S. exigua* consists of an egg stage, five larval stages, a pupal stage and an adult stage (Ali and Gaylor, 1992; Lee *et al.,* 1991a,b; Fye and McAda, 1972). Females of *S. exigua* deposit egg batches on the underside of chrysanthemum leaves near the ground, preferably in young crops (Smits *et al.*, 1986). The first instar larvae are not very mobile and stay near the plant on which the egg batch was laid. They feed predominantly on the lower leaves. During their larval development the foraging domain increases and larvae tend to feed on the upper parts of the plants (Smits *et al.*, 1987). Exclusively the larvae cause host plant injury. The amount of leaf material eaten daily increases with larval instar: the first three instars contribute only to 5% of the crop injury, while 75% of the crop injury is caused by the fifth instar larvae (Smits *et al.*, 1987). After the fifth stage the larvae pupate in the soil. From the pupae new adults emerge. The life cycle parameters, such as developmental times at different temperatures, fecundity, egg batch size, sex ratio and attrition (background mortality) rates are well documented in literature (Ali and Gaylor, 1992; Lee *et al.* 1991a,b; Fye and McAda, 1972).

1.2.2 Dynamics of infection with baculoviruses

Baculoviruses are pathogens that can kill insects. The elementary infectious units of baculoviruses are nucleocapsids, containing double-stranded DNA (Figure 1). Several nucleocapsids are enclosed within a membrane and then called virion. The virions are embedded in a protein body, referred to as a polyhedron. In MNPV's (multicapsid nucleopolyhedroviruses) multiple virions are embedded in a polyhedron or occlusion body (Vlak, 1993).

The polyhedra are taken up orally by the larvae together with leaf material. After ingestion the polyhedra dissolve under the alkaline conditions in the midgut. The virions are liberated and infect the midgut epithelial cells. Virus replication takes place in the nucleus of infected cells. During infection of *S. exigua* MNPV (SeMNPV), up to 10^9 polyhedra can be produced in a single fifth instar larvae (Smits and Vlak, 1988a). A few days after infection the

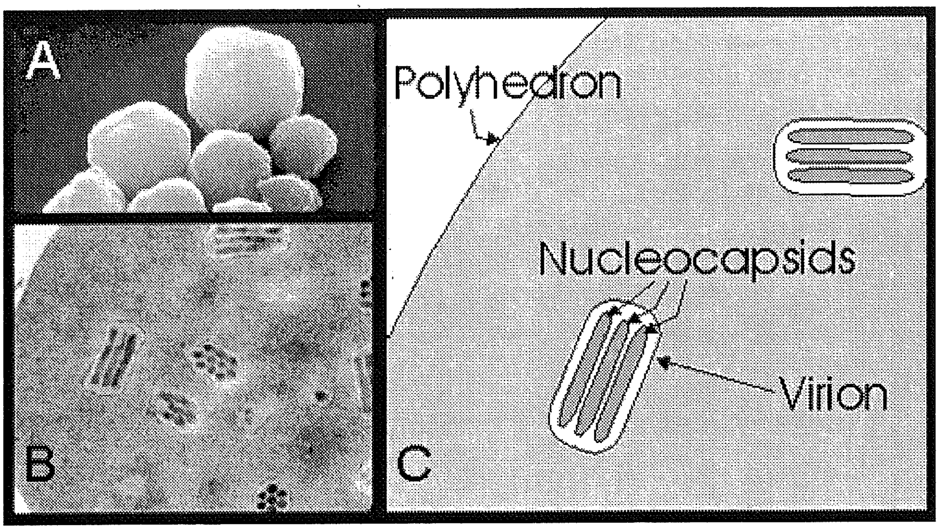


Figure 1. Scanning electron micrograph of polyhedra of LdMNPV (a), transmission micrograph of a polyhedron (b), and schematic representation of a polyhedron (c). (Graphic kindly supplied by Vince D'Amico.)

larvae die and their bodies disintegrate. The body contents of the cadavers containing the polyhedra are spilled on the plant and soil, and can cause new infections when ingested by uninfected larvae. The delay between larval infection and death is called virus incubation time and is strongly affected by temperature. Virus incubation times generally decrease with increasing temperature (Tuan *et al.*, 1989; Boucias *et al,* 1980; Stairs, 1978). It is well established that susceptibility to baculoviruses decrease sharply as larvae age. Lethal doses of first instar larvae to SeMNPV are several polyhedra, whereas fifth instar larvae are almost resistant to baculoviruses (Bianchi *et al.*, 2000a; Smits and Vlak, 1988b). Polyhedra can be inactivated by ultra-violet light (Jaques, 1985; Jones *et al.*, 1993). Polyhedra exposed to direct sunlight can be inactivated within one day (Ignoffo *et al.*, 1997), but polyhedra can persist for extended periods if they are protected against UV radiation (McLeod *et al.*, 1982; Evans, 1986). The glass of greenhouses filters UV radiation, which leads to a reduced virus inactivation in greenhouses compared to the field (Bianchi *et al.*, 1999).

Infection of the larvae with SeMNPV can take place via three routes. First, larvae can ingest *sprayed polyhedra*. If the dose ingested is high enough, the larvae will become lethally infected and die in a few days, spilling newly produced polyhedra on leaves and soil. The second transmission route is *horizontal transmission*. Horizontal transmission is the infection of larvae by ingestion of polyhedra released from cadavers, and can be seen as an encounter process of larvae with contaminated leaves. Because of the large number of polyhedra released from the cadavers, infections caused by horizontal transmission will almost always be lethal. The third route of infection is *vertical transmission*, which is the direct transfer of virus from parents to their own offspring. Fifth instar larvae exposed to polyhedra may become sublethally infected. These larvae develop into sublethally infected moths, which are able to transmit the virus infection to part of their offspring (Smits and Vlak, 1988b).

2 Model Concept

2.1 Mixed simulation

Mixed simulation is the simulation of discrete (discontinuous) and continuous processes within one model (Kettenis, 1990). In BACSIM both continuous processes (e.g. inactivation of polyhedra on leaves) and discrete processes (e.g. deposition of an egg batch, harvest of chrysanthemums) are simulated. BACSIM is a spatial model in which locations of cadavers and egg batches are recorded at the individual level. The bookkeeping of locations is therefore an individual-based process.

The order of calculation in continuous models is given in Figure 2a. These models are initialised in the initial part and auxiliary variables, rate variables, output and state variables are calculated in the dynamic part (van Kraalingen, 1995). In BACSIM continuous processes are simulated according this integration order.

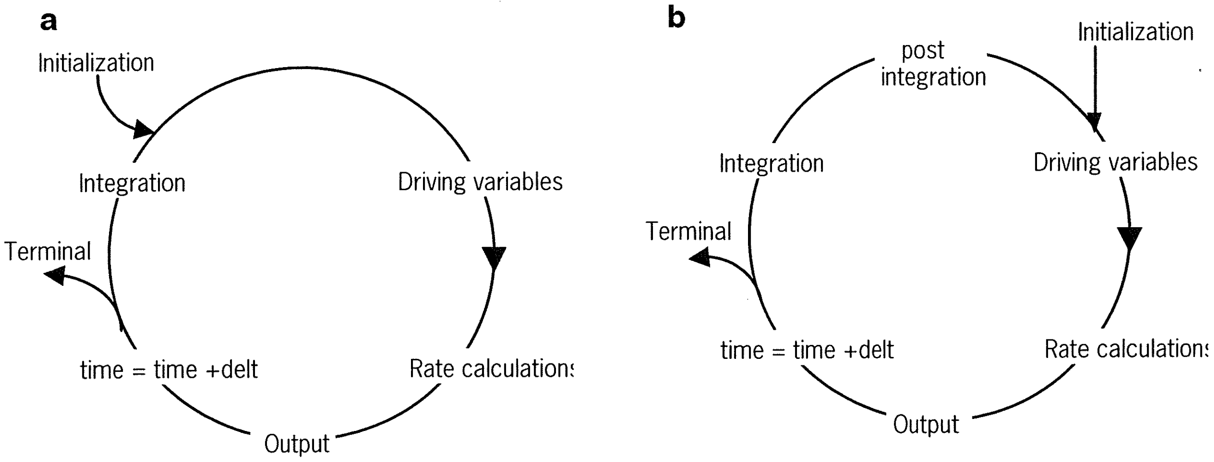


Figure 2. Integration order in continuous models (a) and mixed continuous/discrete models (b).

The regular integration order of continuous processes cannot be used for the simulation of discrete processes because for discrete processes new state variables have to be initialised in the dynamic part of the simulation model. This can be illustrated by the following three examples. When a female moth deposits an egg batch, a new patch is initiated. For this new patch the egg batch size should be initialised during the dynamic integration part. A second example is the harvest of chrysanthemums. If a bed of chrysanthemums is harvested and a new bed is planted in its place, the LAI of the new bed has to be initialised, while patches, cadavers and sprayed polyhedra on the harvested crop have to be removed. This means that numbers of eggs, larvae, etc within all patches in the harvested bed have to be reset during the dynamic integration part. A third example is a calculation of a shift in a boxcar train (Goudriaan and van Roermund, 1993). As soon as the cyclic development stage exceeds a certain value, boxcar contents are shifted to the next boxcar, resulting in adjusted distribution of boxcar contents over the boxcars. In all three examples states variables have to be updated for a discrete event during the dynamic integration part.

This updating of states by discrete events can best be done immediately after the completion of the integration step. Thus, for the simulation of discrete events and bookkeeping an additional calculation step is introduced. In this step, discrete events such as state variable shifting within boxcars, harvests of chrysanthemum beds and the caterpillars in them and the initialisation of egg batches can be simulated, and the bookkeeping can be done. The post integration step is carried out after the integration of state variables because the discontinuous events are triggered by values of state variables, and before the rate calculations because the rates are determined by the (updated) state variables. The integration order of mixed models in the dynamic part is thus: calculation of auxiliary variables, rate variables, output, integration of state variables and discrete events (Figure 2b)

2.2 Modelling different levels of spatial detail

Spatial processes play an important role in the epidemiology of *S. exigua* and baculoviruses. For example, the location of cadavers relative to patches with susceptible larvae is important for the horizontal transmission process. When a cadaver is located in the foraging domain of larvae, the larvae may become infected by feeding from a contaminated leaf. However, not all spatial processes are important. A less important spatial process is, for example, the location of pupae in the greenhouse developing in moths. The location of the pupae has no direct effect on the infection process or population dynamics of the insect.

In BACSIM spatial processes are simulated at four scales of spatial detail: the greenhouse scale, the bed scale, the patch scale and leaf scale (Table 1). The population pupae and moths is simulated on a greenhouse scale; further detail of spatial distribution is not necessary. The density of sprayed polyhedra is simulated at a bed scale. Each chrysanthemum bed (plants with the same age) has a specific polyhedra distribution over the leaf layers after a virus application. The eggs and larval populations are simulated at a patch scale. The development of the larvae within different patches is simulated separately. Per patch the location of the plant on which the egg batch is deposited and the five larval foraging domains are stored. Further spatial detail is introduced by assuming that larvae of specific larval instars spend different times in three canopy layers. The location of cadavers is simulated at a leaf scale. The dying locations (leaves) of infected larvae are drawn from a uniform distribution over their foraging domain. For further details on cadaver allocation see sections 4.3.3. and 6.3.

Table. 1 Four levels of spatial detail in BACSIM

Spatial level	Process
Greenhouse	Development of pupae and moths
Bed	Crop growth (LAI and number of leaves), Density profile of spayed polyhedra
Patch	Development of eggs and larvae, Infection processes
Leaf	Allocation of cadavers

2.3 Modelling discrete events in time

BACSIM is a spatially explicit simulation model. When larvae die as a result of baculovirus infection, these cadavers have to be assigned to specific leaves that will be contaminated. Therefore, insect development and mortality has to be simulated in a discrete fashion. If insect development was simulated in a continuous fashion, a non-discrete number of cadavers might have to be assigned to leaves. However, it is impossible to assign, for example, 0.23 cadaver to a leaf. Therefore, insect development is simulated in a discrete way. For this purpose a fractional integer boxcar train has been developed that expresses boxcar contents and rates (differences) as integers. Results of the fractional integer boxcar train converge for high numbers within the boxcar to those for fractional boxcar trains, but the simulated dynamics deviate in a random fashion from the ordinary fractional boxcar as population numbers become smaller.

2.4 Stochasticity

BACSIM contains stochastic elements. The following processes are simulated using a random generator: mating of sublethally infected and uninfected moths, the simulation of egg batch sizes and their location in the crop, the number of contaminated eggs in egg batches laid by sublethally infected moths, the location of cadavers in the crop and the rounding off of population numbers to discrete numbers during insect development in the fractional integer boxcar trains.

3 Model description

3.1 Greenhouse description and crop growth

The greenhouse is considered as a closed compartment with plants in growing beds. Each bed consists of plants of the same age and plant characteristics. The crop in each bed is characterised by leaf area index (LAI) and number of leaves per plant. The growth of the LAI and number of leaves per plant are described as logistic growth curves which start with an initial LAI and number of leaves of newly planted cuttings:

$$\mathrm{LAI} = \mathrm{LAI0} + (\mathrm{LAImax} - \mathrm{LAI0})/(1+\mathrm{e}{-}\mathrm{rgr} \times (\mathrm{t}-\mathrm{m})) \qquad (1)$$

where LAI_0 is the initial LAI, LAI_{max} is the maximum LAI, rgr is the relative growth rate of the LAI, t is the time and m is the inflection point of the curve. No effects of radiation or temperature are incorporated in the crop model. All plants are divided into a number of LAI strata. At planting, chrysanthemums consist of 3 LAI strata, whereas full-grown chrysanthemums have 60 LAI strata. The thickness of a LAI stratum is calculated as the maximum LAI (7.0 m^2 m^{-2}) divided by 60, i.e. each LAI stratum represents 7.0/60 m^2 m^{-2}. The LAI strata are clustered into an upper, middle and lower canopy layer. Each canopy layer contains (approximately) one-third of the actual LAI strata (Figure 3). The LAI strata are used to store information of the densities sprayed polyhedra, whereas the canopy layers are used to account for the foraging behaviour of the different larval instars. The beds are harvested when a certain LAI is reached. New cuttings are planted immediately after harvest.

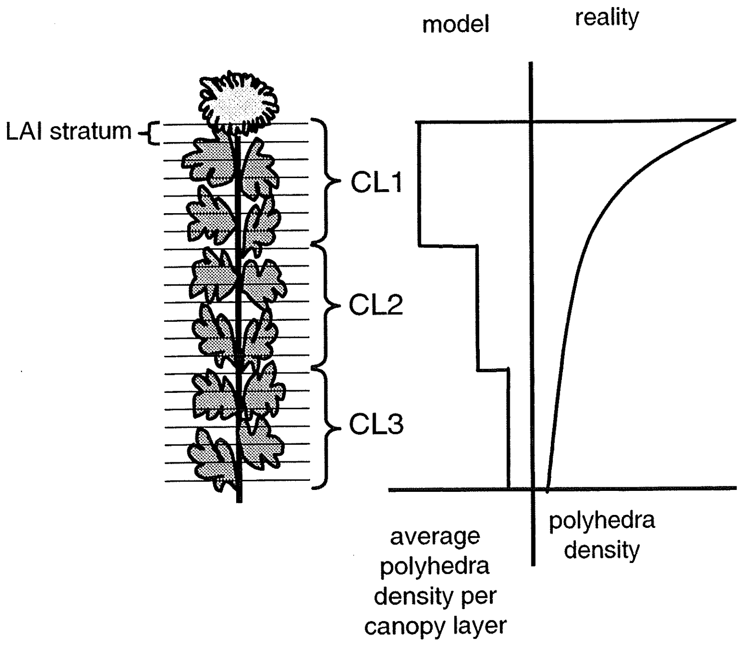


Figure 3. Schematic representation of a chrysanthemum crop divided into LAI strata and canopy layers (CL).

3.2 Development of *S. exigua*

S. exigua passes several stages during its development from egg to adult. The development, variation in development (dispersion) and attrition (mortality by any cause except baculovirus infection) is simulated using the fractional integer boxcar train. This is an adapted version of the fractional boxcar train described by Goudriaan and Van Roermund (1993). In fractional boxcar trains the total development duration of one stage is divided into a number of classes or boxcars, each representing a constant proportion of development time. Development is simulated by shifting a fraction of the population in each boxcar to the next after a certain time interval. The time between shifts and the fraction shifted is determined by the development rate, the number of boxcars and the relative dispersion in development time that is to be simulated. Attrition is simulated by removing individuals from the boxcar train. At the end of a boxcar train, individuals can be shifted to a subsequent boxcar train, representing the next development stage.

The relational diagram of the life cycle of *S. exigua* is presented in Figure 4. *S. exigua* has an egg stage, five larval stages, a pupal stage and an adult stage. All stages have specific development times, dispersion and attrition rates that are affected by temperature (Lee *et al.*, 1991a; Fye and McAda, 1972). Each stage is described by a fractional integer boxcar train. The development of third, fourth and fifth instar larvae is simulated in two parallel boxcar trains for relatively susceptible and resistant larval sub-populations (see section 3.6). Half of the second instar larvae develops in relatively susceptible third instar larvae whereas the other half develops in relatively resistant third instar larvae. Relative susceptible third instar larvae remain susceptible in the following larval instars. All stages have 4 boxcars, except for the adult stages that have 2 boxcars. The input data for the boxcar train of each stage are development rate and standard deviation as function of temperature of that stage, as well as the relative attrition rate. The fraction of larvae that develops normally are shifted to the boxcar train of the next stage, a fraction may die by attrition and a fraction may become infected.

For each patch another set of boxcar trains simulates the development and attrition of eggs and larvae. Larvae that become infected are transferred from the boxcar trains of uninfected larvae to the boxcar trains for infected larvae. A fraction of infected larvae may die by attrition and the rest develops into infectious cadavers. Bodies of infected larvae that die by attrition are not considered infectious. The delay between infection and death of virus-killed larvae is the virus' speed of action. The virus' speed of action is affected by temperature and may differ for different larval instars. The mean time to kill of a virus and its standard deviation, both as function of temperature, are input data for the boxcar trains for infected larvae.

For each patch another set of boxcar trains simulates the development and attrition of eggs and larvae. Larvae that become infected are transferred from the boxcar trains of uninfected larvae to the boxcar trains for infected larvae. A fraction of infected larvae may die by attrition and the rest develops into infectious cadavers. Bodies of infected larvae that die by attrition are not considered infectious. The delay between infection and death of virus-killed larvae is the virus' speed of action. The virus' speed of action is affected by temperature and may differ for different larval instars. The mean time to kill of a virus and its standard deviation, both as function of temperature, are input data for the boxcar trains for infected larvae.

Uninfected L4 larvae that develop normally are shifted to a boxcar train that simulates the development of L5 larvae. A fraction of L5 larvae develops into pupae, whereas other fractions may die by attrition, or may become lethally or sublethally infected. The development of insects in these various stages are simulated in sets of boxcar trains for uninfected pupae, lethally infected L5 larvae, sublethally infected L5 larvae and sublethally

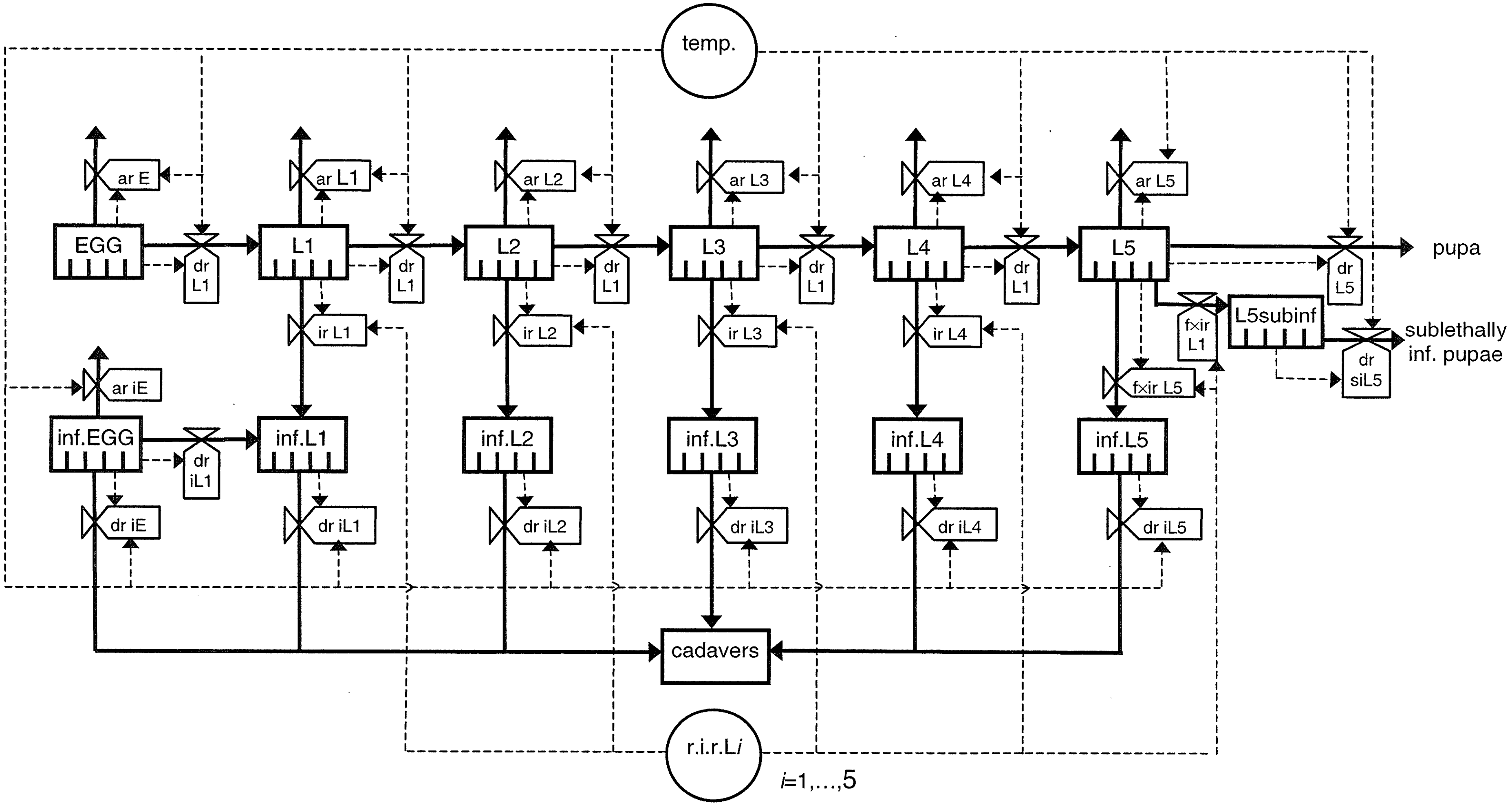


Figure 4. Relational diagram of the development of eggs and larvae of *S. exigua* at the patch scale. Third, fourth and fifth instar larvae are simulated in two parallel boxcar trains for relatively susceptible and resistant sub-populations. Legend: dr E = development rate of eggs; dr Ei = development rate of contaminated eggs; dr Ln = development rate of Ln larvae; dr Lni = development rate of infected Ln larvae; dr L5si = development rate of sublethally infected L5 larvae; ar E = attrition rate of eggs; ar Ei = attrition rate of contaminated eggs; ar Ln = attrition rate of Ln larvae; ar Lni = attrition rate of infected Ln larvae; ir Ln = infection rate of Ln larvae; rir Ln = relative infection rate of Ln larvae; f = fraction; n=1,..,5.

infected pupae. The sex ratio determines the fraction pupae that develop into male or female moths, respectively. It is assumed that there are always enough males to fertilize all females. Fertilized female moths deposit a fixed number of egg batches that vary in size. The number of eggs per egg batch is drawn from a negative binomial distribution with a mean of 34.2 eggs per egg batch and a dispersion parameter of 2.89 (derived from Smits *et al.*, 1986). From these egg batches new clusters of larvae develop that feed on the crop.

3.3 Spatial distribution of *S. exigua*

Female moths have a preference for oviposition on young plants. The egg batches are deposited randomly in chrysanthemum beds that have an LAI lower than 3.5 (derived from Smits *et al.*, 1986). When no chrysanthemum beds with an LAI below 3.5 are available, the egg batches are deposited randomly over all available beds. Egg batches are always deposited in the lowest canopy layer. The locations of egg batches in the crop are recorded during simulation using the index number of bed, row and the index number of the plant in the row. From egg batches new patches will develop, characterized by the co-ordinates of the plant on which the eggs were laid. The foraging domain of each larval instar consists of a fixed number of plants, and the borders are characterized by a number of plants north, west, south and east of the plant on which the eggs were laid. Hence, the larval foraging domains have a rectangular shape. The foraging domain of each instar includes the foraging domain of the preceding instar. The spatial distribution of patches in the crop will be used for the simulation of horizontal transmission.

Larval preference for the upper, middle or lower canopy layers is simulated by allocating fractions of residence time of specific larval instars over the three canopy layers of the plants in their foraging domains (Smits *et al.,* 1987). L1 and L2 larvae forage mainly in the lower and middle canopy layers, whereas the later instar larvae forage mainly in the middle and upper canopy layers. The individual foraging behaviour of the larvae is not simulated and the larvae are assumed to be distributed homogeneously within each canopy layer.

3.4 Spray deposition

The model simulates the deposition of baculovirus over canopy depth for greenhouse chrysanthemum crops that are sprayed from the top. The simulated polyhedra density on leaves after baculovirus applications decreases from top to bottom of the plants. Leaves are assumed to intercept the same fraction of incoming polyhedra in each LAI stratum (equivalent to 7.0/60 m^2 m^{-2}). The number of intercepted polyhedra per LAI stratum can be calculated as the number of incoming polyhedra at the top of the LAI stratum minus the number of outgoing polyhedra at the bottom of the LAI stratum. The number of outgoing polyhedra is equal to the number of incoming polyhedra in the next lower LAI stratum. This results in an exponential polyhedra interception profile, which is characterized by an extinction factor k (Figure 3). The intercepted polyhedra of a LAI stratum after a virus application are added to the polyhedra, which were still present from previous applications. The number of intercepted polyhedra per LAI stratum is converted to a mean density of sprayed polyhedra per canopy layer to calculate horizontal transmission.

3.5 Inactivation of polyhedra

Polyhedra that are sprayed on the crop are inactivated by UV radiation. Polyhedra have exponential inactivation profiles over time, hence inactivation of sprayed polyhedra can be simulated by assuming a constant relative inactivation rate and a residual number of inactivated polyhedra, which are assumed to be resistant to inactivation (Bianchi *et al.*, 1999):

$$PIB = PIB_{res} + (PIB_0 - PIB_{res}) \times e^{(-rir \times t)} \tag{2}$$

where PIB is the density of infectious polyhedra, PIB_0 is the initial density infectious of polyhedra, PIB_{res} is the density of residual infectious polyhedra, rir is the relative polyhedra inactivation rate and t is time. Infected larvae that turn into cadavers spill polyhedra on the leaves. These polyhedra are assumed to be inactivated in the same fashion as sprayed polyhedra.

3.6 Infection by ingestion of sprayed polyhedra

Larvae may become infected after ingestion of polyhedra. The relationship between the dose of ingested polyhedra dose and larval mortality is described with the exponential model (Peto, 1953; Hughes *et al.*, 1984). This model assumes that each ingested polyhedron has the same infection chance of causing a lethal infection (p) and that all larvae are equally susceptible to the virus. The exponential model can be noted as:

$$M = 1 - (1-p)^{n} \tag{3}$$

where M is the fraction mortality, p is the infection chance per ingested polyhedron and n is the polyhedra dose. Since larvae become more resistant to baculovirus infections in their later instars, each instar has a specific infection chance per ingested polyhedron. Populations of third, fourth and fifth instar *S. exigua* larvae generally show variation in susceptibility against baculoviruses, which is reflected in reduced slopes of dose-mortality relationships (Bianchi *et al.*, 2000a). Therefore, the development of third, fourth and fifth instar larvae is simulated in two parallel boxcar trains for relatively susceptible and resistant sub-populations of equal size. Each sub-population has a specific infection chance per ingested polyhedron. The infection chance per ingested polyhedron for susceptible and resistant sub-populations is determined by fitting the double exponential model (Equation 4) to results of dose-mortality bioassays.

$$M = 1 - 0.5 \times (1-p_s)^{n} - 0.5 \times (1-p_r)^{n} \tag{4}$$

where M is the fraction mortality, p_s is the infection chance per ingested polyhedron for the susceptible sub-population, p_r is the infection chance per ingested polyhedron for the resistant sub-population and n is the polyhedron dose.

The number of larvae that will be infected by ingestion of sprayed polyhedra in a patch can be calculated by the multiplication of the number of larvae that are in a particular instar and the relative infection rate by ingestion of sprayed polyhedra for that instar. The relative infection rate by ingestion of sprayed polyhedra (r_s, d^{-1}) depends on the leaf consumption rate of the larvae (*C*, cm^2 d^{-1}), the polyhedra density (*PD*) in the canopy layer (cm^{-2}) and the infection chance per ingested polyhedron (p) for each instar. The relative infection rate by sprayed polyhedra for a specific larval instar can then be calculated from:

$$r_s = C \times PD \times p \qquad (5)$$

where r_s is the relative infection rate by ingestion of sprayed polyhedra (d^{-1}), C the leaf consumption rate (cm^2 d^{-1}), *PD* the polyhedra density (cm^2) and p the infection chance per ingested polyhedron for that larval instar (-).

3.7 Infection by horizontal transmission

Horizontal transmission is the infection of uninfected larvae by ingestion of polyhedra spread by cadavers, which is basically an encounter process between foraging larvae with contaminated leaves. The location of cadavers spreading polyhedra are simulated at the leaf scale, and this is used to calculate the relative rate of infection by horizontal transmission of larvae foraging in patches, r_{ht}. The relative infection rate by horizontal transmission is the same as the probability per unit time that a larva consumes part of a contaminated leaf. Each infected cadaver is assumed to contaminate a whole leaf with a lethal dose of polyhedra.

For each patch, the fraction contaminated leaves in the foraging domains of each larval instar are calculated on the basis of the co-ordinates of cadavers, the number of plants in the foraging domain and the number of leaves per plant. The relative infection rate by horizontal transmission per larval instar is determined by the fraction of contaminated leaves in the larval foraging domain and the number of leaf visits per day of the larvae. The relative infection rate by horizontal transmission can thus be calculated as:

$$r_{ht} = \Delta t^{-1} \times (1 - (1 - f)^{LV \times \Delta t}) \qquad (6)$$

where r_{ht} is the relative infection rate by horizontal transmission (d^{-1}), Δt the time step of integration (d), *f* the fraction of contaminated leaves in the larval foraging domain (-) and LV the number of leaf visits per day (d^{-1}).

3.8 Infection by vertical transmission

Vertical transmission is the direct transfer of virus from sublethally infected moths to their own offspring. Fifth instar larvae exposed to polyhedra may become sublethally infected. These larvae develop into sublethally infected moths, which are able to transmit the virus infection to part of their offspring (Bianchi *et al.*, 2001a; Smits and Vlak, 1988b), most likely by contamination of egg batches. Uninfected and sublethally infected female moths are assumed to have no mating preference for uninfected or sublethally infected male moths. Assuming that female moths mate only once and males mate as often as is needed to fertilize all females, the chance of an uninfected female mating with a sublethally infected male is equal to the proportion of sublethally infected males. Both sublethally infected female moths and uninfected female moths that mated with a sublethally infected male are assumed to be able to deposit contaminated egg batches. Not all egg batches from these female moths are contaminated. In a study of Bianchi *et al.* (2001a) using sublethally SeMNPV infected *S. exigua* moths, 70% of the egg batches were virus free, whereas the remaining 30% included at least one contaminated egg. Egg batches containing forty eggs or more generally gave rise to high fractions of contaminated eggs. In contrast, egg batches containing less than forty eggs seldom gave rise to contaminated eggs, and if these egg batches contained contaminated eggs, the contaminated eggs only constituted a small fraction of the total number of eggs. Vertical transmission is simulated using a critical egg batch size that determines whether an egg batch is large or small, and large and small egg batches are assumed to have a different chance to contain at least one contaminated egg. The number of contaminated eggs in small and large contaminated egg batches is drawn from a uniform distribution, and both type of egg batches have specific maximal fractions of contaminated eggs per egg batch. Contaminated eggs develop into infected L1 larvae, which can transmit the virus horizontally when they turn into cadavers.

3.9 Crop injury

Two indicator variables are calculated to express crop injury by *S. exigua* larvae. One is the area foliage consumed and the second is the number of damaged plants. The area foliage consumption in the greenhouse is calculated as the sum of the daily foliage consumption (C, m^2 d^{-1}) in all patches. The foliage consumption of infected larvae is determined by the instar in which the larvae become infected and the incubation time of the virus. As a consequence, larvae infected with a fast killing virus will feed less than larvae infected with a slower killing virus.

The rate at which *S. exigua* larvae infest chrysanthemum plants is derived from the larval leaf visit rate (LV, d^{-1}) and the larval foraging domains. The plant visit rate for each larval instar is calculated as a fraction of the larval leaf visit rate. Larvae are assumed to injure plants within the foraging domain of their instar and have no preference for feeding on intact or damaged plants. The plant damage caused by infected larvae is determined by instar in which the larvae become infected and the incubation time of the virus. Larvae infected with a fast killing virus will infest a smaller number of plants than larvae infected with a slower killing virus.

4 Model parameterisation

4.1 Crop development

4.1.1 LAI increment of chrysanthemum

LAI increment of greenhouse chrysanthemum (cultivar Cassa) was studied by Heuvelink and Lee (1998). The LAI increment of chrysanthemum can well be described by a logistic curve. The measured LAI increment and the fitted logistic curve is shown in Figure 5.

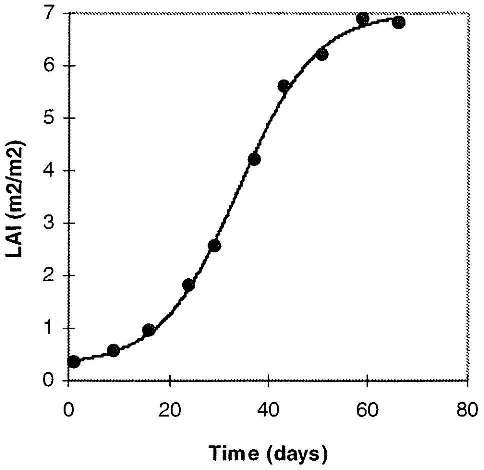


Figure 5. Measured and fitted LAI increment of greenhouse chrysanthemum.

In Equation 7 the logistic function that is fitted to the chrysanthemum LAI increment data is given. The fitted parameter values are presented in Table 2.

$$LAI = LAI_0 + (LAI_{max} - LAI_0)/(1+e^{-rgr \times (t-m)}) \qquad (7)$$

Table 2. Parameter values of fitted logistic LAI increment curve

Symbol	Value	Unit	Acronym	Description
LAI_0	0.29	m^2m^{-2}	LAI0	Initial LAI
LAI_{max}	7.01	m^2m^{-2}	LAIMAX	Maximal LAI
rgr	0.1287	d^{-1}	RGRLAI	Relative growth rate of LAI
m	33.86	d	DAPLAI50	Days after planting when inflection point of logistic curve is reached

4.1.2 Leaves increment of chrysanthemum

The increment of the number of leaves in greenhouse chrysanthemum is also based on the study of Heuvelink and Lee (1998). The increase of the number of leaves over time during the growth of the crop can be described by a logistic curve as well. The measured number of leaves per plant and the fitted curve is shown in Figure 6.

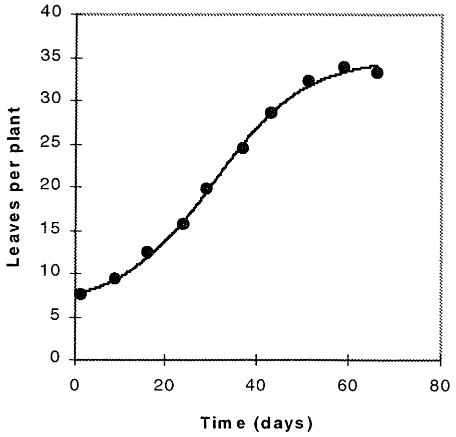


Figure 6. Measured and fitted leaves per plant increment of greenhouse chrysanthemum.

In Equation 8 the logistic function that is fitted to the chrysanthemum leaves per plant increment data is given. The fitted parameter values are presented in Table 3.

$$LPP = LPP_0 + (LPP_{max} - LPP_0)/(1+e^{-rgr \times (t-m)}) \tag{8}$$

Table 3. Parameter values of fitted logistic leaves per plant increment curve

Symbol	Value	Unit	Acronym	Description
LPP_0	6.35	-	LPP0	Initial number of leaves per plant
LPP_{max}	34.91	-	LPPMAX	Maximal number of leaves per plant
rgr	0.0998	d^{-1}	RGRLPP	Relative growth rate of leaves per plant
m	30.58	d	DAPLPP50	Days after planting when inflection point of logistic curve is reached

4.1.3 Chrysanthemum LAI at harvest

In Dutch cultivation practice chrysanthemums are harvested eight to ten weeks after planting, depending on the season. In spring and summer, when *S. exigua* is a pest, chrysanthemums are generally harvested around eight weeks after planting. The LAI at harvest according to Equation 1 will then be approximately 6.8. Therefore, the parameter that defines chrysanthemum LAI at harvest, LAIHARV, is set to 6.8.

4.1.4 Preference of *S. exigua* for egg batch deposition

Smits *et al.* (1986) reported that *S. exigua* females have a preference for young chrysanthemum crops for egg batch deposition. The maximal LAI that is still preferred for egg deposition (LAIPREF) is estimated from these data and is set to 3.5. This LAI corresponds to a chrysanthemum height of approximately 40 cm.

4.1.5 Extinction coefficient of spray in chrysanthemum

The extinction coefficient that characterizes the polyhedra interception profile within the chrysanthemum canopy depends on crop characteristics and the application technique. The extinction factor of an AZO pressure air hand sprayer (4 atmosphere) with four Birchmeier helicon safrei nozzles no. 120 in a 4 and 8 weeks old greenhouse chrysanthemum crop have been determined (Bianchi, unpublished data). The extinction factors in a 4 weeks old crop (LAI = 3.7) and 8 weeks old crop (LAI = 7.3) were 0.87 and 0.80 $(m^2 m^{-2})^{-1}$, respectively. The value for the extinction factor (K) is set to 0.80.

4.2 Development of *S. exigua*

4.2.1 Developmental rates of *S. exigua*

S. exigua bionomics have been studied by Fye and McAda (1972) and Lee *et al.* (1991a). In both studies insects were reared on artificial diet at four temperatures. For each development stage the developmental times and their standard deviation were recorded. In Table 4 development times and their standard deviation, as reported by Fye and McAda (1972), are presented. The dataset of Fye and McAda (1972) is selected because the temperature range of this study has the best correspondence with the temperature range in greenhouses for the Dutch situation. The threshold temperature for development of *S. exigua* is approximately 14°C (Ali and Gaylor, 1992; Lee *et al.*, 1991a). The standard parameter set for *S. exigua* developmental rates (DVR) and standard deviation of *S. exigua* developmental rates (SD) can be found in Appendix I.

Table 4. *S. exigua* development times and standard deviation at four temperatures (Fye and McAda, 1972).

Stage	Temperature 20°C		25°C		30°C		33°C	
Egg	5.6	± 0.7	2.9	± 0.5	2.0	± 0.5	1.8	± 0.1
L1	3.6	± 0.5	3.2	± 0.5	2.6	± 0.5	2.0	± 0.0
L2	2.9	± 0.5	1.9	± 0.5	1.5	± 0.5	1.2	± 0.5
L3	2.8	± 0.7	1.7	± 0.5	1.2	± 0.4	1.2	± 0.3
L4	3.3	± 0.6	2.1	± 0.6	1.5	± 0.6	1.3	± 0.5
L5	6.1	± 0.9	4.1	± 0.9	3.1	± 0.7	2.5	± 0.6
pupae	10.4	± 1.2	7.7	± 1.2	5.1	± 0.6	5.1	± 0.6
adult	11.0	± 5.0	11.0	± 3.0	11.0	± 3.0	9.0	± 5.0
egg batch	3.5	± 0.6	3.3	± 1.6	3.8	± 0.8	3.6	± 1.0

4.2.2 Conversion factor for *S. exigua* developmental rates from artificial diet to chrysanthemum

The larval development of *S. exigua* is affected by the food source. Bianchi *et al.* (2001b) studied the developmental rates of *S. exigua* populations in chrysanthemum in a greenhouse. Developmental rates in chrysanthemum were 36% lower than what could be expected from the developmental rates reported by Fye and McAda (1972) of *S. exigua* larvae that were reared on artificial diet. Therefore, developmental rates of larvae reared on artificial diet are converted to developmental rates of larvae reared on chrysanthemum by the incorporation of a conversion factor CFDVR of 0.64 ((100-36)/100).

4.2.3 Attrition rates of *S. exigua*

Smits *et al.* (1987) estimated that 50% of first instar larvae died by attrition in the first instar and that attrition during the following larval stages was low in greenhouse chrysanthemum. Bianchi *et al.* (2001b) studied attrition of populations of *S. exigua* larvae in chrysanthemum in a greenhouse (Table 5). Overall survival from the first to the fifth instar was 40%. This finding corresponds with another study of Bianchi *et al.* (2001a) in greenhouse chrysanthemum in which the overall survival from the first to the fifth instar was 43%. The data in Table 5 are used in the model, unless experiment -specific information is available, e.g. in the case of validation trials (Bianchi *et al.*, 2000b; 2001c).

Table 5. Fraction survival of S. exigua per larval instar in greenhouse chrysanthemum

Stage	Fraction survival	Acronym
L1	0.60	FSURVL1
L2	0.80	FSURVL2
L3	0.88	FSURVL3
L4	0.95	FSURVL4
L5	1.0	FSURVL5

4.2.4 Consumption rates of *S. exigua*

Larval consumption values of *S. exigua* on greenhouse chrysanthemum were determined by Bianchi (unpublished data). In Table 6 foliage consumption during the five larval stages are presented.

Table 6. Total foliage consumption (cm^2) per instar of *S. exigua* in chrysanthemum.

Stage	Uninfected larvae
L1	0.12
L2	0.32
L3	1.2
L4	8.2
L5	20[†]

[†] *Parameters are not based on direct measurements.*

4.2.5 Sex ratio of *S. exigua*

The sex ratios of *S. exigua* were studied by Lee *et al.* (1991a), Griswold and Tumble (1985) and Steiner (1936). There is some variation in sex ratios among these studies, but an overall sex ratio of 0.5 appears to be representative (Table 7). Therefore, the parameter that defines *S. exigua* sex ratio, SEXRAT, is set to 0.5.

Table 7. *S. exigua* sex ratio.

Medium	Temperature	Sex ratio	Source
Artificial diet	20	0.5	Lee *et al.*, 1991a
Artificial diet	25	0.44	Lee *et al.*, 1991a
Artificial diet	30	0.44	Lee *et al.*, 1991a
Celery	17	0.66	Griswold and Tumble, 1985
Celery	17	0.60	Griswold and Tumble, 1985
Artificial diet	25	0.38	Steiner, 1936
Artificial diet	25	0.45	Steiner, 1936

4.2.6 Egg batch size of *S. exigua*

The number of eggs per egg batch was studied by Smits *et al.* (1986). *S. exigua* moths were allowed to mate and deposit egg batches in a greenhouse chrysanthemum crop. The number, location and size of egg batches were recorded. The cumulative frequency distribution of the egg batch size can be characterized with a negative binomial distribution with $\mu = 34.2$ and $k = 2.89$. In Figure 7 the observed and fitted frequency distribution of *S. exigua* egg batch size is given. In BACSIM this function is characterized by the arrays X and FX and is used to simulate egg batch size.

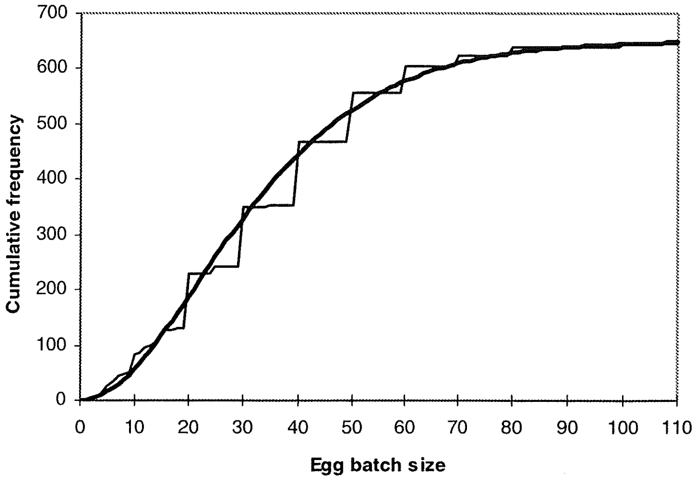


Figure 7. Observed (thin line) and fitted frequency distribution of *S. exigua* egg batch size (thick line).

4.2.7 Number of egg batches deposited by *S. exigua*

The average number of egg batches deposited per female *S. exigua* is derived from studies on *S. exigua* fecundity and the mean size of egg batches deposited by *S. exigua* moths in greenhouse chrysanthemum (Smits *et al.,* 1986). Typical values for fecundity are 400-500 eggs per female, but fecundity can vary considerable, ranging from 200 to 1500 eggs per female (Lee *et al.,* 1991a; Hanna *et al.*, 1977; Ismail *et al.*, 1976; Fye and McAda, 1972; Steiner 1936). The mean egg batch size found by Smits *et al.* (1986) was 35 eggs per egg batch and assuming a mean fecundity of approximately 450 eggs per female moth, the mean number of egg batches deposited per female will be 13. Therefore, the parameter that defines the number of egg batches deposited per female *S. exigua*, IBATNR, is set to 13.

4.3 Spatial distribution of *S. exigua*

4.3.1 Foraging domains of *S. exigua* larvae

The foraging behaviour of *S. exigua* larvae in greenhouse chrysanthemum was investigated by Smits *et al.* (1987). In 30 cm and 60 cm high greenhouse chrysanthemum crops, the number of injured plants were recorded during the development of *S. exigua* larvae from the first until the fifth instar. Since not all plants in the foraging domains were injured, it is assumed that the larval foraging domains cover an area of 1.5 times the number of injured plants (Smits, unpublished data). In Table 8 the mean number of injured plants in larval foraging domains and the estimated number of plants in the larval foraging domains are presented.

Table 8. Mean number of injured plants in the foraging domains of *S. exigua* larvae and the estimated number of plants in the foraging domains of *S. exigua* larvae in greenhouse chrysanthemum.

Stage	Injured plants	Foraging domain (plants)
L1	6	9
L2	20	30
L3	30	45
L4	50	75
L5	70	105

It is further assumed that the larval foraging domains have a rectangular shape. The shape of the larval foraging domain is defined by the IHORDIST array. The first line contains the numbers of plants that L1 larvae colonize to the west, east, north and south of the plant on which the egg batch was deposited. The second line gives the same information for L2 larvae, and so on. Foraging domains of later larval stages overlap the foraging domains of earlier stages. The IHORDIST array is given:

```
IHORDIST =      1,1,1,1,
                2,3,2,2,
                3,3,2,3,
                4,4,3,4,
                4,4,5,5
```

This parameterisation of IHORDIST approximates the number of plants per larval foraging domain given in Table 9. The number of plants per larval foraging domain is 9, 30, 42, 72 and 99 plants for first to fifth instar *S. exigua* larvae. The shape of the foraging domain is not influential in the simulations because no clustering within the domains is accounted for.

4.3.2 Distribution of *S. exigua* larvae over canopy layers

The distribution of *S. exigua* larvae over the three canopy layers was studied by Smits *et al.* (1987). First instar larvae were mainly distributed in the lowest and middle canopy layers whereas fifth instar larvae were mainly distributed over the middle and top canopy layers. The distribution of residence time by *S. exigua* larvae over the canopy layers is characterized by the VERTDIST array. The first line contains the fraction of time spend by L1 larvae in the lowest, middle and upper canopy layer, the second line the fraction of time spend by L2 larvae in the lowest, middle and upper canopy layer, and so on.

```
VERTDIST =      0.5 , 0.25, 0.25,
                0.5 , 0.25, 0.25,
                0.17, 0.33, 0.5 ,
                0.17, 0.33, 0.5 ,
                0.17, 0.33, 0.5
```

4.3.3 Action radii of infected *S. exigua* larvae

S. exigua larvae that become infected by horizontal transmission will die in the neighbourhood of the contaminated leaf that was the source of infection. The action radius per larval instar, expressed in number of plants, is defined by the parameters IACTRADL1 to IACTRADL5. The parameter values for L1 to L5 are rough estimates derived from the larval leaf visit rates.

IACTRADL1 = 1
IACTRADL2 = 2
IACTRADL3 = 2
IACTRADL4 = 3
IACTRADL5 = 2

The dying locations of infected larvae that are infected by horizontal transmission are drawn randomly from the plants that are located within the action radius of the larvae. However, larvae will always die within the foraging domain of their instar.

4.4 Virus dynamics

4.4.1 Polyhedral inactivation of AcMNPV and SeMNPV

Polyhedral inactivation of AcMNPV on greenhouse chrysanthemum was studied by Bianchi *et al.* (1999). Polyhedron inactivation curves could best be described by a model that distinguishes two fractions of polyhedra: a residual fraction of infectious polyhedra that is not inactivated and a fraction of infectious polyhedra that is inactivated in an exponential fashion. In Equation 9 the polyhedron inactivation function is given:

$$PIB = PIB_{res} + (PIB_0\text{-}PIB_{res}) \times e^{(-rir \times t)} \qquad (9)$$

The inactivation curve of SeMNPV is derived according to the same procedure as that of AcMNPV (Bianchi *et al.*, 1999). In Figure 8 the measured and simulated inactivation are presented. The parameters of AcMNPV and SeMNPV are given in Table 9.

Table 9. Residual density infectious polyhedra (m^{-2} leaf) and relative polyhedra inactivation rate (d^{-1}) of AcMNPV and SeMNPV on greenhouse chrysanthemum.

Symbol	AcMNPV	SeMNPV	Acronym	Description
PIBres	1.3 x 10^7	8 x 10^5	RESPHD	Residual density infectious polyhedra
rir		0.16	0.10	RINACTR Relative polyhedra inactivation rate

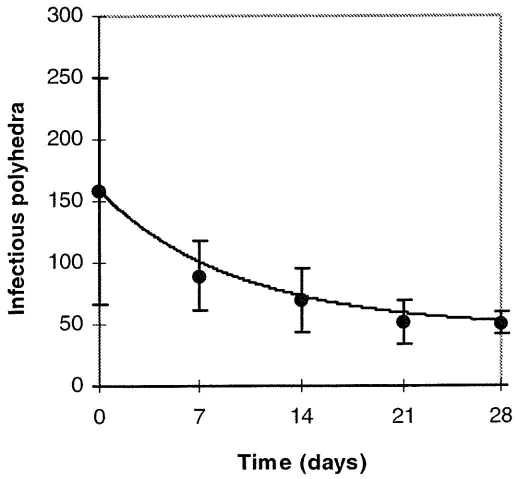


Figure 8 Measured and simulated SeMNPV inactivation on greenhouse chrysanthemum.

4.4.2 AcMNPV and SeMNPV infection chances per ingested polyhedron

Infection chances per ingested polyhedron of AcMNPV and SeMNPV are based on dose-mortality experiments using the leaf disc bioassay (Bianchi *et al.*, 1999) and the droplet-feeding bioassay (Bianchi *et al.*, 2000a). The leaf disc bioassay generates more realistic values than the droplet feeding bioassay because the larvae ingest the polyhedra together with leaf material. The droplet-feeding bioassay methods gives higher estimates of the infection chance per ingested polyhedron because larvae are starved before the bioassay and there is no food in the larval gut that can interfere with the infection process. The leaf disc assay was used to determine the infection chance per ingested polyhedron of AcMNPV a SeMNPV for second, third and fourth instar larvae and for fifth instar larvae for SeMNPV (Table 10). The infection chance per ingested polyhedron is determined by fitting Equation 3 for second instar *S. exigua* larvae and Equation 4 for third, fourth and fifth instar larvae to dose-mortality data. An example of the fit of the double exponential model to dose-mortality data is given in Figure 9. The leaf disc bioassay cannot be used for first instar larvae because the larvae are too small to ingest an entire leaf disc. Infection chances per ingested polyhedron of AcMNPV and SeMNPV for first instar larvae and AcMNPV for fifth instar larvae are based on droplet-feeding bioassay (Bianchi *et al.*, 2000a). LD_{50} values of the droplet-feeding bioassay were converted to more realistic LD_{50} values by a conversion factor that is based on a comparison of LD_{50} values for leaf disc and droplet feeding bioassays with second, third and fourth instar larvae. In Table 10 AcMNPV a SeMNPV infection chances per ingested polyhedron for L1 to L5 larvae are presented.

4.4.3 Conversion factor for infection chance per ingested polyhedron from artificial diet to chrysanthemum

The susceptibility of *S. exigua* to baculoviruses is affected by the food source on which the larvae are reared. *S. exigua* populations that were reared on chrysanthemum had 1.5-3 times higher infection chances per ingested polyhedron than larvae that were reared on artificial diet (Bianchi *et al.*, 2001b). Thus, larvae in greenhouse chrysanthemum are more susceptible than larvae reared on artificial diet in the laboratory. In addition, first to third instar *S. exigua* larvae feed preferably on the underside of chrysanthemum leaves without perforating the leaf (Smits *et al.*, 1987). Since at this location virtually no polyhedra are deposited, the larval polyhedra intake will be reduced (Bianchi, unpublished data).

Table 10. Estimated AcMNPV an SeMNPV infection chances per ingested polyhedron for relatively susceptible (p_s) and relatively resistant (p_r) sub-populations of L1 to L5 *S. exigua* larvae.

Stage	AcMNPV		SeMNPV		Acronym
	p_s	p_r	p_s	p_r	
L1	4.1×10^{-2}†	-	2.9×10^{-2}†	-	PL1
L2	$7.5\ 10^{-3}$	-	$2.6\ 10^{-2}$	-	PL2
L3	$4.7\ 10^{-3}$	$3.4\ 10^{-4}$	$2.5\ 10^{-2}$	$4.3\ 10^{-3}$	PL3S, PL3R
L4	$6.6\ 10^{-4}$	$7.9\ 10^{-5}$	$1.7\ 10^{-3}$	$2.6\ 10^{-3}$	PL4S, PL4R
L5	$1.2\ 10^{-5}$	$4.8\ 10^{-6}$	$8.5\ 10^{-5}$	$1.9\ 10^{-5}$	PL5S, PL5R

† *Parameters derived from droplet-feeding bioassays*

Thus, larvae feeding on chrysanthemum are more susceptible to polyhedra than those feeding on artificial diet, but because these larvae are mainly feeding on the underside of the leaves, their polyhedra uptake is limited. Neither aspect is accounted for in the model other than by an inclusion of a correction factor for the infection chance per ingested polyhedron. In the presented model implementation it is assumed that both processes neutralise each other. Therefore, the conversion factor for the infection chance per ingested polyhedron from artificial diet to chrysanthemum, CFP, is set to 1.

4.4.4 AcMNPV and SeMNPV incubation times in *S. exigua* larvae

The virus incubation time is the time between the uptake of a polyhedron dose by larva and larval death. Virus incubation times of AcMNPV and SeMNPV in third and fourth instar *S. exigua* larvae have been determined by Bianchi *et al.* (2001c) at 23, 28 and 33°C using the droplet-feeding bioassay (Table 11 and 12). The virus incubation times of AcMNPV and SeMNPV in first, second and fifth instar *S. exigua* larvae are extrapolated from the data of third and fourth *S. exigua* larvae. The threshold temperature of *S. exigua* development is also used as threshold temperature for AcMNPV and SeMNPV incubation times. Virus incubation times of AcMNPV and SeMNPV and their standard deviations are defined by the parameters DVRL1I-DVRL5ID and SDL1I-SDL5ID. The complete parameter set can be found in Appendix I.

Table 11. Mean AcMNPV incubation times of infected *S. exigua* larvae at 23, 28 and 33°C (Bianchi *et al.*, 2001c).

		Temperature					
Stage	% Mortality	23°C		28°C		33°C	
L3	44	6.8	± 0.5	4.5	± 0.7	3.8	± 0.3
L4	80	7.2	± 0.8	4.5	± 1.0	4.4	± 1.2

Table 12. Mean SeMNPV incubation times of infected *S. exigua* larvae at 23, 28 and 33°C (Bianchi *et al.*, 2001c).

		Temperature					
Stage	% mortality	23°C		28°C		33°C	
L3	72	5.4	± 0.6	3.7	± 0.5	2.7	± 0.4
L4	94	6.3	± 1.6	3.4	± 0.6	2.7	± 0.3

4.4.5 AcMNPV and SeMNPV polyhedron production in *S. exigua* larvae

The production of polyhedra of AcMNPV and SeMNPV by infected *S. exigua* larvae was determined by Gutiérrez (1998; Table 13). *S. exigua* larvae that were infected always died in the following instar. The polyhedra production per larval instar is defined by PHDPROD and can be found in Appendix I.

Table 13. Polyhedra production ± SD (*10^6) by AcMNPV and SeMNPV infected *S. exigua* larvae.

Stage	AcMNPV		SeMNPV	
L1	4	± 3	3	± 4
L2	120	± 42	80	± 47
L3	400	± 230	355	± 321
L4	693	± 220	767	± 653
L5	nd		nd	

4.5 Infection by horizontal transmission

4.5.1 Leaf visit rate of *S. exigua* larvae

Leaf visit rates of *S. exigua* larvae, measured by Bianchi *et al.* (2001b), are presented in Table 14.

Table 14. Leaf visit rate (d^{-1}) for five larval instars of *S. exigua.*

Stage	Leaf visit rate	Acronym
L1	1[†]	LPDL1
L2	2.0	LPDL2
L3	2.0	LPDL3
L4	2.9	LPDL4
L5	1.5	LPDL5

[†] *Parameter is based on observations in greenhouse chrysanthemum.*

4.6 Infection by vertical transmission

4.6.1 Fraction infected L5 larvae developing into sublethally infected *S. exigua* moths

Smits and Vlak (1988b) and Bianchi *et al.* (2001a) showed that *S. exigua* L5 larvae exposed to high doses of SeMNPV polyhedra can become sublethally infected and transmit SeMNPV to their progeny. However, the fraction larvae developing into sublethally infected adults is difficult to assess. Since L5 larvae are almost resistant to virus it is assumed that as many individuals become sublethally infected as become lethally infected in dose-response trials (expert judgement Smits). Therefore, the fraction fifth instar larvae developing into sublethally infected adults, FINFL5S, is set to 0.5.

4.6.2 Fraction contaminated egg batches deposited by sublethally infected *S. exigua* moths

The distribution of infected first instar *S. exigua* larvae developing from egg batches deposited by sublethally infected moths is studied by Bianchi *et al.* (2001a). Large egg batches generally gave raise to a high number of infected larvae, whereas the larvae developing from small egg batches were mostly uninfected. Therefore, large and small egg batches are distinguished for the simulation of vertical transmission by sublethally infected *S. exigua* moths. The size of a large egg batch is defined by the critical egg batch size, CBATSIZE, and is set to 40. Large egg batches contain 40 or more eggs, whereas small egg batches contain fewer than 40 eggs. The fraction small egg batches that gives raise to one or more infected larvae, FCBATS, is set to 0.10, whereas this fraction for large egg batches, FCBATL, is set to 0.80.

4.6.3 Maximal fraction contaminated eggs in contaminated egg batches

The fraction contaminated eggs in contaminated egg batches deposited by sublethally infected *S. exigua* moths are based on a study of Bianchi *et al.* (2001a). For small egg batches the maximal fraction infected progeny, FCEGGS, is set to 0.20 whereas this fraction for large egg batches, FCEGGL, is set to 0.5. During simulation, the number of infected progeny in contaminated egg batches is drawn from a uniform distribution over the range from 0 to these maximum values.

5 Use of the model

BACSIM is written in FORTRAN-90. It is recommended to use Digital Visual FORTRAN 6.0 (or a later version) for the use of the model. In the next section guidelines for the installation and use of BACSIM are given.

5.1 Installation of the model

1. Install Digital Visual FORTRAN 6.0 on your computer.
2. Create a C:\sys\fst directory and copy the TTSELDEC.EXE file to this directory.
3. Start Digital FORTRAN 6.0 or 6.1 and define a BACSIM project by clicking the File menu and selecting New. Choose the Project tab from the dialog box and select the Fortran Console Application.
4. Specify the name of the project (BACSIM) and the location (C:\PROGRAM FILES\MICROSOFT VISUAL STUDIO\MyProjects\).
5. Select an empty project without links to Fortran DLL Import Libraries. A new workspace for the BACSIM project will be created.
6. Copy the files of the BACSIM installation diskette from A: to C:\PROGRAM FILES\ MICROSOFT VISUAL STUDIO\MyProjects\BACSIM (BACSIM.FOR, INCON.DAT, CROP.DAT, SEDEV.DAT, SPADIS.DAT, VIRUS.DAT, TTUTIL.LIB, DRIVERS.LIB, HELVB.FON).
7. Add files to the BACSIM Project by clicking the FORTRAN Project menu, Add to Project, Files. Select all files that have been copied from the BACSIM installation diskette and have been copied to C:\PROGRAM FILES\MICROSOFT VISUAL STUDIO\MyProjects\BACSIM.
8. Build and execute BACSIM by clicking Build and selecting execute BACSIM.exe.
9. Install TTSELECT by clicking Tools, Customize and selecting the Tools tab from the dialog box. Specify the menu content (TTSELECT), the command (c:\sys\fst\ttseldec.exe) and select the file directory as initial directory. Select the Command tab from the dialog box and select the Tools category. Drag the “Hammer” button that corresponds with the TTSELECT tool to the Toolbar.

Simulations of BACSIM can now be graphically represented by clicking the “Hammer” button form the Toolbar, which will start TTSELECT.

5.2 Initial conditions, parameter settings and output

BACSIM does not require interactive input during execution. The runs are completely specified in the input files. There are 5 input files: one input file with initial conditions and four input files with parameter values (Table 15). The user has to supply initial conditions by editing the input file (INCON.DAT) for each simulation run. Parameter settings can be modified by editing the parameter files. For detailed information on input and parameter file contents is referred to Appendix I.

Table 15. Input and parameter files of BACSIM

Name	File type	Contents
INCON.DAT	Input file	Initial conditions for simulation run (e.g. initial population numbers, number of beds in greenhouse)
CROP.DAT	Parameter file	Crop growth parameters
SEDEV.DAT	Parameter file	*S. exigua* development parameters
SPADIS.DAT	Parameter file	*S. exigua* spatial distribution parameters
VIRUS.DAT	Parameter file	Baculovirus parameters

Special care is needed for the following situations:

1 The initial conditions INSPRAY and INCLVS specify the number of sprays during the simulation and the initial number contaminated leaves in the greenhouse. When INSPRAY and INCLVS are set to 0, the timing and dose of virus applications, as well as the location of contaminated leaves are ignored.

2 The location of contaminated leaves in the greenhouse is initialized in the array INFOCLVS by the co-ordinates: bed, row, plant per row and leaf number. The co-ordinates should always range between 1 and IBEDMAX, IROWMAX, IPPRMAX and the maximum number of leaves of the plants in the bed. When an initial co-ordinate is invalid, an error message is send to the screen.

3 It is not possible to initialize infected larvae (ILI, IL2I, IL3I, IL4I and L5ID) because there are no dying locations available for these larvae. The variables IEGGI, IPUPI, IMALI, IFEMI and BATCHI can be initialized because no dying locations are needed for these insect stages.

4 The number of initial patches (INPAT) should correspond with the initialization of population numbers (See appendix I for example).

BACSIM output can be graphically represented using the TTSELECT program (Rappoldt and Van Kraalingen, 1990). In TTSELECT variables can be plotted against each other. In Table 16 the list of default variables that can be plotted is given.

Table 16. Default variable list that can be plotted using TTSELECT

Name	Unit	Description
Time	d	Time
Itotegg	#/patch	Total number of eggs in the greenhouse
Itotl1-Itotl5	#/patch	Total number of L1 to L5 larvae in the greenhouse
Itotpup	#	Total number of pupae in the greenhouse
Itotmal	#	Total number of male moths in the greenhouse
Itotfem	#	Total number of female moths in the greenhouse
Itoteggi	#/patch	Total number of contaminated eggs in the greenhouse
Itotl1i-Itotl4	#/patch	Total number of infected L1 to L4 larvae patches in the greenhouse
Itotl5is	#/patch	Total number of sublethally infected L5 larvae in all patches in the greenhouse
Itotl5id	#/patch	Total number of lethally infected L5 larvae in all patches in the greenhouse
Itotpupi	#	Total number of sublethally infected pupae in the greenhouse
Itotmali	#	Total number of sublethally infected male moths in the greenhouse
Itotfemi	#	Total number of sublethally infected female moths in the greenhouse
Itotip	#	Total number of insects living in patches (eggs and larvae)
Itotop	#	Number of insects living outside patches (pupae and moths)
Itot	#	Total number of insects in greenhouse
Ipatch	#	Number of patches in greenhouse
Contleaf	#	Number of contaminated leaves in greenhouse
Icumcadaver	#	Cumulative number of cadavers in greenhouse
Phdlartot	#	Total number of polyhedra released from cadavers present in the greenhouse
Phdbed1	polyhedra/m^2	Number of sprayed polyhedra on the crop in bed 1
Lai1	m^2/m^2	LAI of chrysanthemums in patch 1
Laibed1	m^2/m^2	LAI of chrysanthemums of bed1
Ilpp1	#	Number of leaves of chrysanthemums in patch 1
Ilppbed1	#	Number of leaves of chrysanthemums in bed 1
Fccum	m^2	Cumulative area of foliage consumption in greenhouse
Dpcumo	#	Cumulative number of damaged plants, adjusted for overlapping patches
Rirl1ht-	d^{-1}	Relative infection rate by horizontal transmission of L1 to L5 Rirl5ht larvae in patch 1
Rirl1s-Rirl5s	d^{-1}	Relative infection rate by ingestion of sprayed polyhedra of L1 to L5 larvae in patch 1

6 Implementation of the model

In the next section the model structure and the coherence of the subroutines will be outlined. Each subroutine will be discussed separately and the simulation of the specific processes will be explained. In Table 17 the subroutine structure of BACSIM is given. In the FILES subroutine output files can be declared. The INITIM subroutine is only called in the initial part of the program to initialise time variables. There are six subroutines that are called in the initial and dynamic part: CROP, RIRSPR, RIRHT, SEDEV, SPADIS and VIRUS. The main program calls these subroutines for each integration step (section 2.1). There are six integration steps: initialisation, calculation of auxiliary variables, rate variables, generation of output, integration of state variables and a post integration step. The type of calculation that a subroutine has to perform is indicated by the variable ITASK. If a subroutine has been called and ITASK is equal to one, the subroutine performs the initialisation. ITASK two performs the calculation of auxiliary variables, ITASK three the calculation of rate variables, ITASK four generates the output, ITASK five integrates the state variables and ITASK six the post integration step.

Table 17. BACSIM subroutine structure.

BACSIM	INITIM		
	SPADIS	NEWINF	
		NEWDIL	
		CONTROL	
		HARV	
	SEDEV	IBOXCAR	
		IBOXPUP	
		IBOXPATL5	DRWMORT
		IBOXPAT	DRWMORT
		IBOXPATL2	DRWMORT
		IBOXPTIS	
		IBOXPTI	
		MATING	
		ACCOUNT	
		DEPOSIT	
	VIRUS		
	CROP		
	RIRHT		
	RIRSPR		

6.1 The SEDEV subroutine

In the SEDEV subroutine the *S. exigua* development and population dynamics are simulated. In ITASK one parameters and initial conditions are read and initial conditions are checked for invalid input values. In ITASK two developmental rates, coefficients of variation of developmental rates and attrition rates for infected and uninfected insects are calculated as function of temperature. In ITASK three the transfer of individuals from one stage to the other are calculated (differences).

These transfer differences include infection differences by horizontal transmission, infection differences by ingestion of sprayed polyhedra, attrition differences, death differences of infected larvae, death differences of healthy or sublethally infected moths and development differences to the next stage. The transfer differences for each development stage are calculated in a set of boxcar trains (section 3.2). The ITASK three section continuous with the calculation of the number of fifth instar larvae that enter into the pupal stage. Since the development of egg batches in female oviduct, pupae, male and female moths are simulated on a greenhouse scale, the boxcar trains of these insect stages contain the whole greenhouse population. In contrast, the egg and larval stages are simulated at the patch scale, thus the boxcar trains of these stages contain the population totals per patch. To simulate the transfer of fifth instar larvae from a patch scale into pupae at the greenhouse scale, accumulated fifth instar larvae transfer differences are used as input for the pupal boxcar trains. The ITASK three section ends with call to the subroutine MATING. In this subroutine the mating of newly emerged female moths with male moths is simulated. All newly produced egg batches in the females oviduct per time step are accumulated and flow into the boxcar train for the egg batches in the female oviduct (IBATIN and IBATINI). In ITASK four the output is generated. In the end of this section the subroutine ACCOUNT is called that calculates population totals in and outside patches. In ITASK five the populations of all stages are integrated in the boxcar trains. In the ITASK six larvae located in harvested beds are removed and the bookkeeping of the number of patches takes place. The actual patch number is raised with the number of new patches initiated by egg batch deposition. When the new number of patches exceeds the maximum number of patches (IPMAX) the program is terminated. When new patches are initiated the subroutine DEPOSIT is called. This subroutine simulates the size of the newly deposited egg batches, the number of contaminated and uncontaminated egg batches and the distribution of contaminated and uncontaminated eggs in contaminated egg batches. The egg batch size is drawn from a cumulative frequency distribution FX. The presence of contaminated eggs in an egg batch deposited by sublethally infected moths is drawn. If a random value between 0 and 1 is lower than FCBAT (fraction contaminated egg batches of sublethally infected moths), the egg batch will be contaminated. The distribution of contaminated and uncontaminated eggs in contaminated egg batches are drawn from a uniform distribution. Contaminated egg batches contain at least one and at maximum a fraction of FCEGG (fraction contaminated eggs in contaminated egg batches) contaminated eggs. The ITASK six section is finished with the calculation of the shift in the boxcar trains for the different insect stages.

6.2 The boxcar subroutines

In BACSIM seven different boxcar trains are used: IBOXCAR, IBOXPUP, IBOXPATL2, IBOXPATL5, IBOXPAT, IBOXPTIS and IBOXPTI. These boxcars are based on the same principle but have small differences. Boxcar contents, rates (differences) and shifted boxcar contents in all seven boxcars types are expressed as integers. In ITASK three developmental rates are initially calculated as real values and then converted to differences by stochastic rounding with the function IROUND. This function rounds real values to the nearest integers. The probability that a real value is rounded to the higher integer value equals the value of the decimals of that real value. For example, 2.9 will be rounded to 3 in 90 % of the cases and rounded to 2 in 10 % of the cases. The integration of the boxcar contents is conducted in the ITASK five section. The integration is simply the addition or subtraction of the boxcar content of the last time step with the differences. During the shift (ITASK six), boxcar contents that are shifted to the next boxcar are first calculated as real values and then rounded with the IROUND function.

The IBOXPAT boxcar train subroutine simulates the development and population dynamics of uninfected eggs, L1, L2, L3 and L4 larvae. In the boxcar trains of larvae, individuals can be removed from boxcars by attrition, infection by horizontal transmission and infection by ingestion of sprayed polyhedra. It is possible that the number of larvae claimed for transfer by these three processes is higher than the number of individuals present in the boxcar. In this case the DRWMORT subroutine is called. This subroutine allocates the outflow from larvae in a boxcar to the three categories by a weighed drawing over the relative attrition and infection rates. Infected larvae are transferred to the IBOXPTI subroutine. The IBOXPTI subroutine simulates the development and population dynamics of contaminated egg batches and infected larvae. Newly infected individuals are placed in the zeroth boxcar of this boxcar train. The developmental rate used in this boxcar train corresponds with the virus incubation times of the different larval instars. Individuals that flow out of the IBOXPTI boxcar turn into cadavers. The development and population dynamics of uninfected L2 larvae is simulated in the IBOXPATL2 boxcar train. Half of the population uninfected second instar larvae will flow in the boxcar train of relatively susceptible third instar larvae whereas the other half flows into the boxcar train of relatively resistant third instar larvae. The development and population dynamics of uninfected L5 larvae is simulated in the IBOXPATL5 boxcar train. Fifth instar larvae that become infected can either develop in lethally or sublethally infected fifth instar larvae. Lethally infected larvae are transferred to a boxcar train for infected larvae (IBOXPTI) and sublethally infected are transferred to a boxcar train for sublethally infected L5 larvae (IBOXPTIS). The proportion infected fifth instar larvae developing in lethally and sublethally infected larvae depends on the parameter FINFL5S (Fraction infected L5 larvae developing in sublethally infected moths). Fifth instar larvae that become sublethally infected and are transferred to the IBOXPTIS boxcar train are placed in the same boxcar as they were removed from in the IBOXPATL5 boxcar train. Hence, the sublethal infection process does not lead to a prolonged larval development period. Sublethally infected fifth instar larvae develop into sublethally infected pupae, whereas uninfected L5 larvae develop in uninfected pupae. The IBOXPUP boxcar train simulates the development and population dynamics of uninfected and sublethally infected pupae. Pupae are not susceptible for virus, thus these boxcar trains do not incorporate infection processes. Individuals that flow out of this boxcar train are divided in males and female moths. The IBOXCAR subroutine is used to simulate the development and population dynamics of uninfected and sublethally infected female and male moths, and egg batches in female oviduct. IBOXCAR does not incorporate infection processes.

6.3 The SPADIS subroutine

In the SPADIS subroutine the spatial distribution of *S. exigua* is simulated. The location of egg batches and foraging domains of larvae as well as the location of contaminated leaves (leaves with one or more cadavers) in the crop are stored. The locations of contaminated leaves are stored in the array INFOCLVS. This array contains the co-ordinates of the contaminated leaves: bed, row, plant per row and leaf number, as well as the larval instar of the cadaver. In ITASK one parameters and initial conditions are read and initial conditions are checked for invalid values. Next, the number of plants per larval foraging domain is calculated and the locations of contaminated leaves are stored in the INFOCLVS array. In ITASK two the number of contaminated leaves per larval foraging domain (ICLVSL1 to ICLVSL5) and per canopy layer within a larval foraging domain (ICLVSL1CL to ICLVSL5CL) is determined for each patch. When a contaminated leaf is located in a patch, the rank number of that contaminated leaf in the INFOCLVS array is set by the variables ICLVSNRL1 to ICLVSNRL5. The fraction of contaminated leaves per canopy layer is calculated using the number of contaminated leaves and the total number of leaves in the canopy layer. Next, the determination and bookkeeping of the dying locations of newly infected larvae takes place. This is done in the NEWINF subroutine. The dying location of a newly infected larva in a patch depends on what infection process caused the infection. The NEWINF subroutine is called for L1 larvae that become infected by vertical transmission, for L1 to L5 larvae that become infected by horizontal transmission and for L1 to L5 larvae that become infected by ingestion of sprayed polyhedra. Hatched larvae infected by vertical transmission are assigned randomly to a plant within the foraging domain of first instar larvae using the function IDRWLOT. This function draws a random value between a minimum and maximum integer value. Larvae that become infected by horizontal transmission will die on a plant in the neighbourhood of the contaminated leaf that was the source of infection. The leaf that was the source of infection is drawn from the contaminated leaves within the larval foraging domain. The location of this leaf is read from the INFOCLVS array using the variable ICLVSNR. Each larval instar has a specific action radius, expressed in number of plants, where the plant on which the infected larvae will die is drawn. However, infected larvae are never located outside the foraging domain of that particular instar. Larvae infected by ingestion of sprayed polyhedra are assigned randomly to plants within their foraging domains. The leaf number of the plant on which the infected larvae died is determined by the IDRWLEAF function. This function draws a leaf number weighed by larval residence times spent in the three canopy layers. Thus, if a L1 larva becomes infected, it has the highest probability to die on a leaf in the lowest canopy layer since L1 larvae spend most time in this stratum. Larvae cannot be placed outside the beds. The dying locations of newly infected larvae are stored in the INFOLIL array (Living Infected Larvae). This array contains information about the infected larvae: bed, row, plant per row, leaf number, patch number and larval instar. When an infected larva dies in a patch, the co-ordinates of the oldest infected larvae of the specific instar and patch are transferred from the INFOLIL array to the INFOCLVS array. This is done in the NEWDIL subroutine. The data describing the newly died larvae are removed from the INFOLIL array. In the ITASK four section output is generated and the CONTROL subroutine is called. This subroutine checks whether the number of infected larvae in the boxcar trains correspond with the number of larvae in the INFOLIL array per larval instar. If these numbers are not equal, it is possible to send an error message to the screen or to terminate the program. However, in the default setting attrition of infected larvae is allowed, which will result in differences in the number of larvae in the boxcar trains and in the INFOLIL array (larvae that die by attrition are removed from the boxcar trains but not in the INFOLIL array). Therefore, for the default setting differences in number of larvae in boxcar trains and the INFOLIL array are ignored. The ITASK six starts with a call to the HARV subroutine. This subroutine removes locations of infected larvae and cadavers in the INFOLIL and INFOCLVS arrays of larvae located in harvested patches. The determination of the location of plants where egg batches are deposited takes place next. Female moths will deposit eggs in beds that have a LAI lower than LAIPREF. When no beds with such young plants are available, female moths deposit eggs randomly over the beds.

6.4 The CROP subroutine

In the CROP subroutine the chrysanthemum development and crop injury is simulated. In ITASK one parameters and initial conditions are read, initial conditions are checked for invalid values and the initial LAI and leaves per plant are calculated for each bed. In ITASK two a harvest is triggered when the LAI value of a bed exceeds the value of LAIHARV (LAI when the crop is harvested). Further, the larval leaf visit rate corrected for temperature is calculated. In ITASK three the foliage consumption and the plant damage event rates are calculated. In ITASK four output is generated. In ITASK five the age of each bed (DAP: Days After Planting), foliage consumption and number of damaged plants are integrated. With the updated DAP value the new LAI and leaves per plant are calculated for each bed. The LAI, leaves per plant and foliage consumption are calculated next for each patch. Patches located in the same bed will have the same LAI and the same number of leaves per plant. In the ITASK six section the DAP, LAI and leaves per plant will be initialised when a bed is harvested. In the same section LAI and leaves per plant values are assigned to new patches.

6.5 The VIRUS subroutine

In the VIRUS subroutine the quantity and distribution of sprayed polyhedra over the LAI and canopy layers of the crop is simulated. To simulate the distribution of sprayed polyhedra within the crop, the crop is divided in 60 LAI layers. The layer thickness (DELTLAI) is calculated as the maximum LAI (LAIHARV) divided by 60. Hence, young crops consist only of a few LAI layers whereas old crops can have up to 60 LAI layers. In ITASK one parameters and initial conditions are read, DELTLAI is calculated and the initial polyhedron density for each bed and patches is set to zero. In ITASK two the new quantity and distribution of polyhedra for each LAI layer in each bed after a spraying event is calculated. The distribution of sprayed polyhedra over the LAI layers of each bed is simulated by calculating the fraction incoming and outgoing polyhedra for each LAI layer, assuming a negative exponential extinction profile. Next, the new quantity and distribution of polyhedra for each LAI layer of all patches is calculated. Although the simulation of a spraying event does strictly not belong in the auxiliary variables section (ITASK two), this has the advantage that a spray can be conducted in the first time step. In ITASK three the inactivation rate of polyhedra is calculated for each LAI layer using a relative polyhedron inactivation rate and a residual density of infectious polyhedra on the leaf surface. In ITASK four output is generated. In ITASK five the quantity of polyhedra in each LAI layer of the beds are integrated. Next, the new quantity and distribution of polyheda per patch is calculated. In the ITASK six section the quantity of polyhedra per leaf layer is set to zero if a chrysanthemum bed is harvested. In the same section the polyhedra quantities per LAI layer are assigned to new patches.

6.6 The RIRHT subroutine

In the RIRHT subroutine the relative infection rate by horizontal transmission is calculated. In ITASK one parameters are read. In ITASK three the relative infection rates by horizontal transmission per larval instar and canopy layer are calculated using the fraction contaminated leaves (calculated in SPADIS) and the larval leaf visit rate. The relative infection rates by horizontal transmission per larval instar and canopy layer are averaged proportionally over the larval residence time per canopy layer to obtain relative infection rates by horizontal transmission. The relative infection rates by horizontal transmission per larval instar are used as input variables for the SEDEV subroutine. In ITASK four output is generated.

6.7 The RIRSPR subroutine

In the RIRSPR subroutine the relative infection rate by ingestion of sprayed polyhedra is calculated. In ITASK one parameters are read. In ITASK three the density of sprayed polyhedra are calculated per canopy layer. Relative infection rates by ingestion of sprayed polyhedra for each larval instar are calculated with polyhedra densities, infection chance per ingested polyhedron and larval consumption rates. The relative infection rates by horizontal transmission for each larval instar are used as input variables for the SEDEV subroutine. In ITASK is four output is generated.

References

Ali, A. and Gaylor, M.J. 1992. Effects of temperature and larval diet on development of the beet armyworm (Lepidoptera: Noctuidae). Environ. Entomol. 21: 780-786.

Alvaraso-Rodriguez, B. 1987. Parasites and diseases associated with larvae of beet armyworm, *Spodoptera exigua* (Lepidoptera: Noctuidae) infesting processing tomatoes in Sinaloa, Mexico. Florida entomologist 70: 444-449.

Bianchi, F.J.J.A., Joosten, N.N., Gutiérrez, S., Reijnen, T.M., van der Werf, W., Smits, P.H., and Vlak J.M. 1999. polyhedral membrane does not protect polyhedra of AcMNPV against inactivation on greenhouse chrysanthemum. Biocontrol Sci. Technol. 9: 523-527.

Bianchi, F.J.J.A., Snoeijing, I., Van der Werf, W, Mans, R.M.W., Smits, P.H. and Vlak, J.M. 2000a. Biological activity of SeMNPV, AcMNPV and three AcMNPV deletion mutants against *Spodoptera exigua* larvae (Lepidoptera: Noctuidae). J. Invertebrebr. Pathol. 75: 28-35.

Bianchi, F.J.J.A., Joosten, N.N., Vlak, J.M., and van der Werf, W. 2000b. Greenhouse evaluation of dose- and time mortality relationships of two nucleopolyhedroviruses for the control of beet armyworm, *Spodoptera exigua*, on chrysanthemum. Biological Control 19: 252-258.

Bianchi, F.J.J.A., van Essen, P.H.A., Smits, P.H., and van der Werf, W. 2001a. Transmission dynamics of the multicapsid nucleopolyhedrovirus SeMNPV in *Spodoptera exigua* populations in greenhouse chrysanthemum. Proceedings of the Section Experimental and Applied Entomology of the Netherlands Entomological Society. *In press.*

Bianchi, F.J.J.A., Joosten, N.N., Vlak, J.M., and van der Werf, W. 2001b. The influence of greenhouse chrysanthemum on the interaction between the beet armyworm, *Spodoptera exigua*, and the baculovirus SeMNPV: parameter quantification for a process-based simulation model. *Submitted.*

Bianchi, F.J.J.A., Vlak, J.M., and van der Werf, W. 2001c. Simulation of biological control of beet armyworm, *Spodoptera exigua,* with baculoviruses in greenhouses II: Validation of BACSIM. *Submitted.*

Black, B.C., Brennan, L.A., Dierks, P.M., and Gard, I.E. 1997. Commercialization of baculoviral insecticides. *In* "The Baculoviruses" (L.K. Miller, ed.), pp. 341-387. Plenum Press, New York.

Boucias D.G, Johnson, D.W., and Allan, G.E. 1980. Effects of host age, virus dosage, and temperature on the infectivity of a nucleopolyhedrosis virus against the velvetbean caterpillar, *Anticarsia gemmetalis*, larvae. Environ. Entomol. 9: 59-61.

Brewer, M.J. and Tumble, J.T. 1989. Field monitoring for insecticide resistance in beet armyworm (Lepidoptera: Noctuidae). J. Ec. Entomol. 82: 1520-1526.

Caballero, P., Aldebis, H.K., Vargas-Osuna, E., and Santiago-Alvarez, C. 1992. Epizootics caused by a nuclear polyhedrosis virus in populations of *Spodoptera exigua* in southern Spain. Biocontrol Sci. Technol. 3: 35-38.

De Moed, G.H., van der Werf, W., and Smits, P.H. 1990. Modelling the epizootiology of *Spodoptera exigua* nuclear polyhedrosis virus in a spatially distributed population of *Spodoptera exigua* in greenhouse chrysanthemums. SROP/WPRS Bull. XIII/5: 135-141.

Dougherty, E.M., Guthrie, K.P., and Shapiro, M. 1996. Optical brighteners provide baculovirus activity enhancement and UV radiation protection. Biological control 7: 71-74.

Elnagar S., and Abul Nasr, S. 1980. Effect of direct sunlight on virulence of NPV (nuclear polyhedrosis virus) of the cotton leafworm, *Spodoptera littoralis* (Boisd.). Z. Ang. Ent., 90: 75-80.

Evans, H.F. 1986. Ecology and epizootiology of baculoviruses. In: Granados, R.R. and B.A. Federici, eds. The biology of baculoviruses, vol 2: 89-132. CRC Press, Boca Raton, FL.

Fye, R.E., and McAda, W.C. 1972. Laboratory studies on the development, longevity, and fecundity of six lepidopterous pests of cotton in Arizona. Tech. Bull. 1454 USDA, 73 pp.

Goudriaan, J., and van Roermund, H. J. W. 1993. Modelling of ageing, development, delays and dispersion. In: Leffelaar P.A. (ed). On systems analysis and simulation of ecological processes: 89-126. Kluwer academic, Dordrecht, the Netherlands.

Griswold, M.J. and Tumble, J.T. 1985. Consumption and utilization of celery, *Apium graveolens*, by the beet armyworm *Spodoptera exigua*. Entomol. Exp. Appl. 38: 73-79.

Gutiérrez, S. 1998. Quantification of biological parameters of wild-type and recombinant baculoviruses. MSc-Thesis Agricultural University Wageningen, The Netherlands, 41 pp.

Hanna, H.M., Hamad, N.E.F., and Azad, S.G. 1977. Effect of larval food on the biology of the lesser cotton leafworm. Bull. Soc. Ent. Egypte 61: 225-234.

Heuvelink, E., and Lee, J.H. 1998. Effects of plant density and length of the long-day period on dry matter production in cut chrysanthemum: measurement and simulation. XXV International Horticultural Congress, Brussel, Belgium. Book of abstracts, pp. 386.

Hughes, P.R., Wood, H.A., Burand, J.P., and Granados, R.R. 1984. Quantification of the dose-mortality respons of *Trichoplusia ni*, *Heliothis zea*, and *Spodoptera frugiperda* to nuclear polyhedrosis viruses: applicability of an exponential model. J. Invertebrebr. Pathol. 43: 343-350.

Ignoffo, C.M., Garcia, C. and Saathoff, S.G. 1997. Sunlight stability and rain-fastness of formulations of *Baculovirus heliothis*. Environ. Entomol. 26: 1470-1474.

Ismail, I.I., Megahed, M.M., and Abd-El-Maksoud, Z.M. 1976. A comparative study on the effect of diet on *Spodopters exigua* Hb. Acta Phytopathologica Academiae Scientiarium Hungaricae 11: 111-117.

Jaques, R.P. 1972. The inactivation of foliar deposits of viruses of *Trichoplusia ni* (Lepidoptera: Noctuidae) and *Pieris rapae* (Lepidoptera: Pieridae) and tests on protectant additives. Can. Ent. 104: 1985-1994.

Jaques, R.P. 1985. Stability of insect viruses in the environment. In: Maramorosch, K. and K.E. Sherman (eds.). Viral insecticides for biological control: 285-360. Acad. Press, London.

Jones, K.A., Moawad, G., McKinley, D.J., and Grzywacz, D. 1993. The effect of natural sunlight on *Spodoptera littoralis* nuclear polyhedrosis virus. Biocontrol Sci. Technol. 3: 189-197.

Kettenis, D.L. 1990. Simulatie: met continue en discrete modellen. Stenfert Kroese, Leiden/Antwerpen, 188 pp.

Kolodny-Hirsch, D.M., Warkentin, D.L., Alvarado-Rodrigues, B., and Kirkland, R. 1993. *Spodoptera exigua* nuclear polyhedrosis virus as a candidate viral insecticide for the beet armyworm (Lepidoptera: Noctuidae). J. Econ. Entomol. 86: 314-321.

Kolodny-Hirsch, D.M., Sitchawat, T., Jansiri, T., Chenrchaivachirakul, A. and Ketunuti, U. 1997. Field evaluation of a commercial formulation of the *Spodoptera exigua* (Lepidoptera: Noctuidae) nuclear polyhedrosis virus for control of beet armyworm on vegetable crops in Thailand. Biocontrol Sci. Technol. 7: 475-488.

Lee, S.D., Ahn, S.B., Cho, W.S., and Choi, K.M.1991a. Effects of temperature on the development of beet armyworm, *Spodoptera exigua* Hübner (Lepidoptera: Noctuidae). Research Reports of the Rural Development Administration (Suwan) 33: 58-62.

Lee, S.D., Ahn, S.B., Hong, K.J., and Cho, W.S. 1991b. Effects of three host plants on the development of beet armyworm, *Spodoptera exigua* Hübner (Lepidoptera: Noctuidae). Research Reports of the Rural Development Administration (Suwan) 33: 53-57.

McLeod, P.J., Young, S.Y., and Yearian, W.C. 1982. Application of a baculovirus of *Pseudoplusia includens* to soybean: efficacy and seasonal persistence. Environ. Entomol. 11: 412-416.

Moscardi, F. 1999. Assessment of the application of baculoviruses for control of lepidoptera. Annu. Rev. Entomol. 44: 257-289.

Peto, S. 1953. A dose-response equation for the invasion of microorganisms. Biometrics 9: 320-335.

Rappoldt, C. and van Kraalingen, D.W.G. 1990. FORTRAN utility library TTUTIL. Simulation report CABO-TT no. 20. Centre for Agribiological Research and Dept. of Theoretical Production Ecology, Wageningen, The Netherlands, 54 pp.

Smits, P.H., van de Vrie, M., and Vlak, J.M. 1986. Oviposition of beet armyworm (Lepidoptera: Noctuidea) on greenhouse crops. Environ. Entomol. 15: 680-682.

Smits, P.H., 1987. Nuclear polyhedrosis virus as a biological control agent of *Spodoptera exigua*. PhD-Thesis Agricultural University Wageningen, The Netherlands, 127 pp.

Smits, P.H., van Velden, M.C., van de Vrie, M., and Vlak, J.M. 1987. Feeding and dispersion of *Spodoptera exigua* larvae and its relevance for control with a nuclear polyhedrosis virus. Entomol. Exp. Appl. 43: 67-72.

Smits, P.H., and Vlak, J.M. 1988a. Quantitative and qualitative aspects in the production of a nuclear polyhedrosis virus in *Spodoptera exigua* larvae. Ann. Appl. Biol. 122: 249-257.

Smits, P.H., and Vlak, J.M. 1988b. Biological activity of *Spodoptera exigua* nuclear polyhedrosis virus against *S. exigua* larvae. J. Invertebr. Pathol 51: 107-114.

Stairs, G.R. 1978. Effects of a wide range of temperature on the development of *Galleria mellonella* and its specific baculovirus. Environ. Entomol. 7: 297-299.

Steiner, P. 1936. Beitrage zur Kenntnis der Schadlingsfauna Klein-Asiens III. *Laphygma exigua*, ein grossschadling der Zuckerrube in Anatolien. Z. Ang. Ent. 23: 178-222.

Tuan, S.J., Tang, L.C., and Hou, R.F. 1989. Factors affecting pathogenicity of NPV preparations to the corn earworm, *Heliothis armigera*. Entomophaga 34: 541-549.

Van Beek, N.A.M., and Hughes, P.R. 1998. The response time of insect larvae infected with recombinant baculoviruses. J. Invertebrebr. Pathol. 72: 338-347.

Van der Werf, W., de Moed, G.H. and Smits, P.H. 1991. A comprehensive simulation model of the epidemiology of *Spodoptera exigua* nuclear polyhedrosis virus on beet armyworm in glasshouse chrysanthemums. IOBC/WPRS Bulletin, XIV, 108-117.

Van Kraalingen, D.W.G. 1995. The FSE system for crop simulation, version 2.1. DLO Reseach Institute for Agrobiology and Soil Fertility, C.T. de Wit Graduate school for Production Ecology, 58 pp.

Vlak, J.M. 1993. Genetic engineering of baculoviruses for insect control. *In*: Molecular approaches to pure and applied entomology (M.J. Whitten and J.G. Oakeshot, eds). Springer Series in Experimental Entomology: 90-127.

Appendix I: Listing of input files

```
*-----------------------------------------------------------------*
*     INCON.DAT                                                   *
*     Input file for SPODXNPV.FOR                                 *
*-----------------------------------------------------------------*


*     Time when simulation is terminated (d)
      FINTIM = 3.
*     Time step (d)
      DELT = 0.05
*     Temperature switch: 0 = measured temperatures, 1 = constant
*     temperature
      ICTEMP = 1
*     Constant temperature (degree C)
      TEMP = 25.0
*     Initial number of patches
      INPAT = 4
*     Array containing coordinates of patches in greenhouse
*     (bed, row, plant number in row)
      IPATDAT = 2,25,5,
                15,53,3,
                5,27,7,
                9,88,2
*     Number of uninfected insects per boxcar per patch
      IEGG   = 0,999*0
      IL1    = 100,100,998*0
      IL2    = 0,999*0
      IL3    = 100,999*0
      IL4    = 100,999*0
      IL5    = 0,999*0
      IPUP   = 0,4*0
      IMAL   = 0,2*0
      IFEM   = 0,2*0
      IBATCH = 0,4*0
*     Number of infected insects per boxcar per patch
      IEGGI = 1000*0
      IL1I  = 1000*0
      IL2I  = 1000*0
      IL3I  = 1000*0
      IL4I  = 1000*0
      IL5ID = 1000*0
      IL5IS = 1000*0
      IPUPI = 5*0
      IMALI = 3*0
      IFEMI = 3*0
      IBATCHI = 5*0
```

```
*     Maximal number of beds
      IBEDMAX = 18
*     Maximal number plant rows in bed
      IROWMAX = 100
*     Maximal number of plants per row in bed
      IPPRMAX = 8
*     Initial age of plants in beds (d)
      DAP = 0.,0.,7.,7.,14.,14.,21.,21.,28.,28.,35.,35.,42.,42.,
            49.,49.,56.,56.,2*0.
*     Number of sprays during simulation
      INSPRAY = 3
*     Time of spray (d)
      TSPRAY = 0.,7.,14.
*     Sprayed polyhedra density of application (#.m-2 ground)
      PHDSPR = 3.E7,3.E7,3.E7
*     Initial number of contaminated leaves
      INCLVS = 5
*     Array containing contaminated leaves co-ordinates
*     (bed, row, plant number in row, leaf, larval stadium)
      INFOCLVS= 1,73,8,5,1,
                5,32,5,12,4,
                8,14,2,13,2,
                11,76,3,8,3,
                14,83,7,15,2
*     Measured hourly temperatures (degree C)
      TEMPT =
      0.0000,   25.00,
      0.0417,   26.29,
      0.0833,   27.50,
      0.1250,   28.54,
      0.1667,   29.33,
      0.2083,   29.83,
      0.2500,   30.00,
      0.2917,   29.83,
      0.3333,   29.33,
      0.3750,   28.54,
      0.4167,   27.50,
      0.4583,   26.29,
      0.5000,   25.00,
      0.5417,   23.71,
      0.5833,   22.50,
      0.6250,   21.46,
      0.6667,   20.67,
      0.7083,   20.17,
      0.7500,   20.00,
      0.7917,   20.17,
      0.8333,   20.67,
          .,        .,
     39.9583,   24.35
```

```
*--------------------------------------------------------------------*
*     CROP.DAT                                                        *
*     Parameter file for SPODXNPV.FOR                                 *
*--------------------------------------------------------------------*

*     Relative growth rate of LAI (d-1)
      RGRLAI = 0.1287
*     Inflection point of logistic LAI growth curve (d)
      DAPLAI50 = 33.86
*     Initial leaf area index of crop (m2.m-2)
      LAI0 = 0.29
*     Maximal leaf area index of crop (m2.m-2)
      LAIMAX = 7.01
*     Relative growth rate of leaves per plant (d-1)
      RGRLPP = 0.0998
*     Inflection point of logistic leaves per plant growth curve (d)
      DAPLPP50 = 30.58
*     Initial number of leaves per plant in patch (#)
      LPP0 = 6.35
*     Maximal number of leaves per plant (#)
      LPPMAX = 34.91
*     Leaf area index of crop at crop harvest (m2.m-2)
      LAIHARV = 6.8
*     Maximal leaf area index of crop to be prefered for egg
*     deposition (m2.m-2)
      LAIPREF = 3.5
*     Extinction coefficient of crop for spray (m2.m-2)
      K = 0.8
```

```
*------------------------------------------------------------------*
*     SEDEV.DAT                                                    *
*     Parameter file for SPODXNPV.FOR                              *
*------------------------------------------------------------------*

*     Development rates uninfected insects as function of temperature
*     (d-1)(Fye and McADA, 1972)
      DVREGGT = 14.,0.,20.,.179,25.,.345,30.,.500,33.,.556,37.,.556
      DVRL1T  = 14.,0.,20.,.278,25.,.313,30.,.385,33.,.500,37.,.500
      DVRL2T  = 14.,0.,20.,.345,25.,.526,30.,.667,33.,.833,37.,.833
      DVRL3T  = 14.,0.,20.,.357,25.,.588,30.,.833,33.,.833,37.,.833
      DVRL4T  = 14.,0.,20.,.303,25.,.476,30.,.667,33.,.769,37.,.769
      DVRL5T  = 14.,0.,20.,.164,25.,.244,30.,.323,33.,.400,37.,.400
      DVRPUPT = 14.,0.,20.,.096,25.,.130,30.,.196,33.,.196,37.,.196
      DVRMALT = 14.,0.,20.,.091,25.,.091,30.,.091,33.,.111,37.,.111
      DVRFEMT = 14.,0.,20.,.091,25.,.091,30.,.091,33.,.111,37.,.111
      DVRBATT = 14.,0.,20.,.282,25.,.301,30.,.265,33.,.279,37.,.279
*     Conversion factor of larval development rate from artificial
*     diet to chrysanthemum
      CFDVR = 0.7
*     Standard deviation of development times of uninfected insects
*     as function of temperature (d) (Fye and McAda, 1972)
      SDEGG   = 14.,0.7,20.,0.7,25.,0.5,30.,0.5,33.,0.1,37.,0.1
      SDL1    = 14.,0.5,20.,0.5,25.,0.5,30.,0.5,33.,0.0,37.,0.0
      SDL2    = 14.,0.5,20.,0.5,25.,0.5,30.,0.5,33.,0.5,37.,0.5
      SDL3    = 14.,0.7,20.,0.7,25.,0.5,30.,0.4,33.,0.3,37.,0.3
      SDL4    = 14.,0.6,20.,0.6,25.,0.6,30.,0.6,33.,0.5,37.,0.5
      SDL5    = 14.,0.9,20.,0.9,25.,0.9,30.,0.7,33.,0.6,37.,0.6
      SDPUP   = 14.,1.2,20.,1.2,25.,1.2,30.,0.6,33.,0.6,37.,0.6
      SDMAL   = 14.,5.0,20.,5.0,25.,3.0,30.,3.0,33.,5.0,37.,5.0
      SDFEM   = 14.,5.0,20.,5.0,25.,3.0,30.,3.0,33.,5.0,37.,5.0
      SDBAT   = 14.,0.6,20.,0.6,25.,1.6,30.,0.8,33.,1.0,37.,1.0
*     Fraction survival per larval instar
      FSURVL1  = 0.6
      FSURVL2  = 0.8
      FSURVL3  = 0.88
      FSURVL4  = 0.95
      FSURVL5  = 1.0
      FSURVPUP = 1.0
      FSURVMAL = 1.0
      FSURVFEM = 1.0
*     Consumption per larval instar (m2)
      CONSL1 = 0.000012
      CONSL2 = 0.000032
      CONSL3 = 0.00012
      CONSL4 = 0.00082
      CONSL5 = 0.002
*     Sexratio
      SEXRAT = 0.5
*     Number of eggbatches per female moth
```

```
      IBATNR = 13
*     Fraction of infected L5 larvae that are sublethally infected
      FINFL5S = 0.5
*     Critical eggbatch size to be a large egg batch
      ICBATSIZE = 40
*     Fraction of small eggbatches that are infected by sublethally
*     infected moths
      FCBATS = 0.1
*     Fraction of large eggbatches which are infected by sublethally
*     infected moths
      FCBATL = 0.8
*     Maximal fraction infected eggs in small eggbatches
      FCEGGS = 0.2
*     Maximal fraction infected eggs in large eggbatches
      FCEGGL = 0.5
*     Cumulative frequency distribution of eggnumber per eggbatch
      FX = 0.0023,0.0053,0.0098,0.0159,0.0237,0.0331,0.0441,
           0.0567,0.0707,0.0861,0.1028,0.1205,0.1393,0.1589,
           0.1793,0.2003,0.2218,0.2438,0.2660,0.2885,0.3111,
           0.3337,0.3563,0.3787,0.4010,0.4230,0.4448,0.4662,
           0.4872,0.5078,0.5280,0.5477,0.5669,0.5856,0.6037,
           0.6214,0.6385,0.6550,0.6710,0.6865,0.7014,0.7158,
           0.7296,0.7429,0.7557,0.7680,0.7798,0.7911,0.8019,
           0.8122,0.8221,0.8316,0.8406,0.8493,0.8575,0.8653,
           0.8728,0.8799,0.8867,0.8931,0.8993,0.9051,0.9106,
           0.9159,0.9209,0.9256,0.9301,0.9343,0.9383,0.9422,
           0.9458,0.9492,0.9524,0.9555,0.9584,0.9611,0.9637,
           0.9661,0.9684,0.9706,0.9726,0.9746,0.9764,0.9781,
           0.9797,0.9813,0.9827,0.9841,0.9854,0.9866,0.9877,
           0.9888,0.9898,0.9907,0.9916,0.9924,0.9932,0.9940,
           0.9947,0.9953,0.9959,0.9965,0.9970,0.9975,0.9980,
           0.9985,0.9989,0.9993,0.9996,1.0000
*     X-axis of cumulative eggnumber per eggbatch frequency distribution
      X =  1.,2.,3.,4.,5.,6.,7.,8.,
           9.,10.,11.,12.,13.,14.,15.,16.,
          17.,18.,19.,20.,21.,22.,23.,24.,
          25.,26.,27.,28.,29.,30.,31.,32.,
          33.,34.,35.,36.,37.,38.,39.,40.,
          41.,42.,43.,44.,45.,46.,47.,48.,
          49.,50.,51.,52.,53.,54.,55.,56.,
          57.,58.,59.,60.,61.,62.,63.,64.,
          65.,66.,67.,68.,69.,70.,71.,72.,
          73.,74.,75.,76.,77.,78.,79.,80.,
          81.,82.,83.,84.,85.,86.,87.,88.,
          89.,90.,91.,92.,93.,94.,95.,96.,
          97.,98.,99.,100.,101.,102.,103.,
          104.,105.,106.,107.,108.,109.,110.
```

```
*-----------------------------------------------------------------*
*     SPADIS.DAT                                                  *
*     Parameter file for SPODXNPV.FOR                             *
*-----------------------------------------------------------------*

*     Horizontal distribution of larval foraging areas in patch.
*     The numbers correspond with the number of plants west, east,
*     north and south of the central plant containing the eggbatch.
*     The foraging areas are rectangular and the borders can be
*     determined with these co-ordinates. Larvae in later stadia are
*     also foraging in foraging areas of younger larvae.
      IHORDIST = 1,1,1,1,
                 2,3,2,2,
                 3,3,2,3,
                 4,4,3,4,
                 4,4,5,5
*     Fraction larvae per vertical larval distribution stratum.
*     The numbers correspond with the lowest, middle and upper
*     stratum.
      VERTDIST = 0.5 , 0.25, 0.25,
                 0.5 , 0.25, 0.25
                 0.17, 0.33, 0.5 ,
                 0.17, 0.33, 0.5 ,
                 0.17, 0.33, 0.5
*     Action radia of infected larvae (# plants)
      IACTRADL1 = 2
      IACTRADL2 = 2
      IACTRADL3 = 3
      IACTRADL4 = 3
      IACTRADL5 = 2
*     Mean number of leaves visits per day per larval instar (d-1)
      LPDL1   = 1.
      LPDL2   = 2.
      LPDL3   = 2.
      LPDL4   = 2.9
      LPDL5   = 1.5
```

```
*-------------------------------------------------------------------*
*     VIRUS.DAT                                                     *
*     Parameter file for SPODXNPV.FOR                               *
*-------------------------------------------------------------------*

*     SeMNPV relative polyhedra inactivation rate (d-1)
      RINACTR    = 0.10
*     Residual density of infectious polyhedra on leaf surface (m-2)
      RESPHD     = 8.E5
*     Array containing SeMNPV polyhedra production per larval instar
*     (#.larvae-1)
      PHDPROD    = 3.E6,
                   80.E6,
                   355.E6,
                   767.E6,
                   767.E6
*     SeMNPV infection chance per ingested polyhedron per larval
*     instar (-)
      PL1     = 0.029
      PL2     = 0.026
      PL3S    = 0.025
      PL3R    = 0.0043
      PL4S    = 0.0026
      PL4R    = 0.0017
      PL5S    = 0.000085
      PL5R    = 0.0000193
*     Conversion factor of infection chance per ingested polyhedron
*     from artificial diet to chrysanthemum (-)
      CFP = 1.
*     SeMNPV development rates infected insects as function of
*     temperature (d-1) partly based on Fye and McAda, 1972
      DVREGGIT = 14.,0.,20.,.179,25.,.345,30.,.500,33.,.556,37.,.556
      DVRL1IT  = 14.,0.,23.,.186,28.,.270,33.,.374,36.,.374,37.,.374
      DVRL2IT  = 14.,0.,23.,.186,28.,.270,33.,.374,36.,.374,37.,.374
      DVRL3IT  = 14.,0.,23.,.186,28.,.270,33.,.374,36.,.374,37.,.374
      DVRL4IT  = 14.,0.,23.,.159,28.,.296,33.,.374,36.,.374,37.,.374
      DVRL5IDT = 14.,0.,23.,.159,28.,.296,33.,.374,36.,.374,37.,.374
      DVRL5IST = 14.,0.,20.,.164,25.,.244,30.,.323,33.,.400,37.,.400
      DVRPUPIT = 14.,0.,20.,.096,25.,.130,30.,.196,33.,.196,37.,.196
      DVRMALIT = 14.,0.,20.,.091,25.,.091,30.,.091,33.,.111,37.,.111
      DVRFEMIT = 14.,0.,20.,.091,25.,.091,30.,.091,33.,.111,37.,.111
      DVRBATIT = 14.,0.,20.,.282,25.,.301,30.,.265,33.,.279,37.,.279
*     SeMNPV standard deviation of development times of infected
*     insects as function of temperature (d) partly based on Fye and
*     McAda, 1972
      SDEGGI   = 14.,0.7,20.,0.7,25.,0.5,30.,0.5,33.,0.1,37.,0.1
      SDL1I    = 14.,0.6,23.,0.6,28.,0.5,33.,0.4,36.,0.4,37.,0.4
      SDL2I    = 14.,0.6,23.,0.6,28.,0.5,33.,0.4,36.,0.4,37.,0.4
      SDL3I    = 14.,0.6,23.,0.6,28.,0.5,33.,0.4,36.,0.4,37.,0.4
      SDL4I    = 14.,1.6,23.,1.6,28.,0.6,33.,0.3,36.,0.3,37.,0.3
```

```
        SDL5ID   = 14.,1.6,23.,1.6,28.,0.6,33.,0.3,36.,0.3,37.,0.3
        SDL5IS   = 14.,0.9,20.,0.9,25.,0.9,30.,0.7,33.,0.6,37.,0.6
        SDPUPI   = 14.,1.2,20.,1.2,25.,1.2,30.,0.6,33.,0.6,37.,0.6
        SDMALI   = 14.,5.0,20.,5.0,25.,3.0,30.,3.0,33.,5.0,37.,5.0
        SDFEMI   = 14.,5.0,20.,5.0,25.,3.0,30.,3.0,33.,5.0,37.,5.0
        SDBATI   = 14.,0.6,20.,0.6,25.,1.6,30.,0.8,33.,1.0,37.,1.0
*       AcMNPV relative polyhedra inactivation rate (d-1)
        RINACTR     = 0.16
*       Residual density of infectious polyhedra on leaf surface (m-2)
        RESPHD      = 1.7E6
*       Array containing AcMNPV polyhedra production per larval instar
*       (#.larvae-1)
        PHDPROD     = 4.E6
                      120.E6
                      400.E6
                      693.E6
                      693.E6
*       AcMNPV infection chance per ingested polyhedron per larval instar
        PL1      = 0.041
        PL2      = 0.0075
        PL3S     = 0.0047
        PL3R     = 0.00034
        PL4S     = 0.00066
        PL4R     = 0.000079
        PL5S     = 0.000012
        PL5R     = 0.0000048
*       Conversion factor of infection chance per ingested polyhedron
*       from artificial diet to chrysanthemum (-)
        CFP = 1.
*       AcMNPV development rates infected insects as function of
*       temperature (d-1) partly based on Fye and McAda, 1972
        DVREGGIT = 14.,0.,20.,.179,25.,.345,30.,.500,33.,.556,37.,.556
        DVRL1IT  = 14.,0.,23.,.147,28.,.222,33.,.263,36.,.263,37.,.263
        DVRL2IT  = 14.,0.,23.,.147,28.,.222,33.,.263,36.,.263,37.,.263
        DVRL3IT  = 14.,0.,23.,.147,28.,.222,33.,.263,36.,.263,37.,.263
        DVRL4IT  = 14.,0.,23.,.139,28.,.227,33.,.227,36.,.227,37.,.227
        DVRL5IDT = 14.,0.,23.,.139,28.,.227,33.,.227,36.,.227,37.,.227
        DVRL5IST = 14.,0.,20.,.164,25.,.244,30.,.323,33.,.400,37.,.400
        DVRPUPIT = 14.,0.,20.,.096,25.,.130,30.,.196,33.,.196,37.,.196
        DVRMALIT = 14.,0.,20.,.091,25.,.091,30.,.091,33.,.111,37.,.111
        DVRFEMIT = 14.,0.,20.,.091,25.,.091,30.,.091,33.,.111,37.,.111
        DVRBATIT = 14.,0.,20.,.282,25.,.301,30.,.265,33.,.279,37.,.279
*       AcMNPV standard deviation of development rates of infected
*       insects as function of temperature (d-1) partly based on Fye
*       and McAda, 1972
        SDEGGI   = 14.,0.7,20.,0.7,25.,0.5,30.,0.5,33.,0.1,37.,0.1
        SDL1I    = 14.,0.5,23.,0.5,28.,0.7,33.,0.3,36.,0.3,37.,0.3
        SDL2I    = 14.,0.5,23.,0.5,28.,0.7,33.,0.3,36.,0.3,37.,0.3
        SDL3I    = 14.,0.5,23.,0.5,28.,0.7,33.,0.3,36.,0.3,37.,0.3
        SDL4I    = 14.,0.8,23.,0.8,28.,1.0,33.,1.2,36.,1.2,37.,1.2
```

```
SDL5ID   = 14.,0.8,23.,0.8,28.,1.0,33.,1.2,36.,1.2,37.,1.2
SDL5IS   = 14.,0.9,20.,0.9,25.,0.9,30.,0.7,33.,0.6,37.,0.6
SDPUPI   = 14.,1.2,20.,1.2,25.,1.2,30.,0.6,33.,0.6,37.,0.6
SDMALI   = 14.,5.0,20.,5.0,25.,3.0,30.,3.0,33.,5.0,37.,5.0
SDFEMI   = 14.,5.0,20.,5.0,25.,3.0,30.,3.0,33.,5.0,37.,5.0
SDBATI   = 14.,0.6,20.,0.6,25.,1.6,30.,0.8,33.,1.0,37.,1.0
```

Appendix II: List of acronyms

Principles for name giving

Integer variables always start with an I, whereas real variables never start with an I. The following abbreviations are consistently used for the different insect stages:
EGG = eggs
L1 = first instar larvae
L2 = second instar larvae
L3 = third instar larvae
L4 = fourth instar larvae
L5 = fifth instar larvae
PUP = pupae
MAL = male moths
FEM = female moths
BAT or BATCH = egg batches in oviduct
Uninfected third to fifth instar larvae are divided in two sub-populations of relatively susceptible and resistant larvae. The susceptible populations are indicated with and S, e.g. IL3S, whereas the relatively resistant subpopulations are indicated with an R, e.g. IL4R. Infected insect stages are indicated by the abbreviation of each stage followed by an I (infected), e.g. IEGGI, IL3I, IPUPI, IFEMI. Fifth instar larvae may become lethally or sublethally infected. Lethally infected fifth instar larvae are indicated with an additional D (dying), e.g. IL5ID, whereas sublethally infected fifth instar larvae are indicated with an additional S (sublethally infected), e.g. IL5IS. Population totals in the greenhouse are indicated with TOT, e.g. ITOTL1I representing the total number of infected L1 larvae in the greenhouse. Variables representing cumulative values are indicated with the abbreviation CUM, e.g. FCCUM representing the cumulative area of foliage consumption since the start of simulation.

Acronym	Variable type	Dimension	Description
CDS	state variable	-	Cyclic development stage
CDSBATCH	state variable	-	Cyclic development stage of egg batches in oviduct
CDSBATCHI	state variable	-	Cyclic development stage of infected egg batches in oviduct
CDSEGG	state variable	-	Cyclic development stage of egg batches
CDSEGG	state variable	-	Cyclic development stage of infected egg batches
CDSFEM	state variable	-	Cyclic development stage of female moths
CDSFEMI	state variable	-	Cyclic development stage of sublethally infected female moths
CDSL1	state variable	-	Cyclic development stage of L1 larvae
CDSL1I	state variable	-	Cyclic development stage infected of L1 larvae
CDSL2	state variable	-	Cyclic development stage of L2 larvae
CDSL2I	state variable	-	Cyclic development stage infected of L2 larvae
CDSL3I	state variable	-	Cyclic development stage infected of L3 larvae
CDSL3R	state variable	-	Cyclic development stage of resistant L3 larvae
CDSL3S	state variable	-	Cyclic development stage of susceptible L3 larvae
CDSL4I	state variable	-	Cyclic development stage infected of L4 larvae
CDSL4R	state variable	-	Cyclic development stage of resistant L4 larvae
CDSL4S	state variable	-	Cyclic development stage of susceptible L4 larvae
CDSL5ID	state variable	-	Cyclic development stage of lethally infected L5 larvae
CDSL5IS	state variable	-	Cyclic development stage of sublethally infected L5 larvae
CDSL5R	state variable	-	Cyclic development stage of resistant L5 larvae
CDSL5S	state variable	-	Cyclic development stage of susceptible L5 larvae
CDSMAL	state variable	-	Cyclic development stage of male moths
CDSMALI	state variable	-	Cyclic development stage of sublethally infected male moths
CDSPUP	state variable	-	Cyclic development stage of pupae
CDSPUPI	state variable	-	Cyclic development stage of sublethally infected pupae
CFDVR	parameter	-	Conversion factor of developmental rate from artificial diet to chrysanthemum
CFP	parameter	-	Conversion factor of infection chance per ingested polyhedron from artificial diet to chrysanthemum
CONSL1	parameter	m2.d-1	Consumption rate of L1 larvae
CONSL2	parameter	m2.d-1	Consumption rate L2 larvae

CONSL3	parameter	m2.d-1	Consumption rate of L3 larvae
CONSL4	parameter	m2.d-1	Consumption rate of L4 larvae
CONSL5	parameter	m2.d-1	Consumption rate of L5 larvae
CUM	auxiliary	-	Help variable for drawing procedure
CV	auxiliary	-	Coefficient of variation
CVBATCH	auxiliary	-	Coefficient of variation of egg batches in oviduct
CVBATCHI	auxiliary	-	Coefficient of variation of contaminated egg batches in oviduct
CVEGG	auxiliary	-	Coefficient of variation of egg batches
CVEGGI	auxiliary	-	Coefficient of variation of contaminated egg batches
CVFEM	auxiliary	-	Coefficient of variation of female moths
CVFEMI	auxiliary	-	Coefficient of variation of sublethally infected female moths
CVL1	auxiliary	-	Coefficient of variation of L1 larvae
CVL1I	auxiliary	-	Coefficient of variation of infected L1 larvae
CVL2	auxiliary	-	Coefficient of variation of L2 larvae
CVL2I	auxiliary	-	Coefficient of variation of infected L2 larvae
CVL3	auxiliary	-	Coefficient of variation of L3 larvae
CVL3I	auxiliary	-	Coefficient of variation of infected L3 larvae
CVL4	auxiliary	-	Coefficient of variation of L4 larvae
CVL4I	auxiliary	-	Coefficient of variation of infected L4 larvae
CVL5	auxiliary	-	Coefficient of variation of L5 larvae
CVL5ID	auxiliary	-	Coefficient of variation of lethally infected L5 larvae
CVL5IS	auxiliary	-	Coefficient of variation of sublethally infected L5 larvae
CVMAL	auxiliary	-	Coefficient of variation of male moths
CVMALI	auxiliary	-	Coefficient of variation of sublethally infected male moths
CVPUP	auxiliary	-	Coefficient of variation of pupae
CVPUPI	auxiliary	-	Coefficient of variation of sublethally infected pupae
DAP	state variable	d	Age of plants in a bed
DAPLAI50	parameter	d	Inflection point of logistic LAI growth curve
DAPLPP50	parameter	d	Inflection point of logistic leaves per plant growth curve
DELT	rate variable	d	Time step
DELTLAI	initial condition	m2.m-2	LAI stratum
DPBED	state variable	#	Number of damaged plants per bed, ignoring overlapping patches
DPBEDO	state variable	#	Number of damaged plants per bed, adjusted for overlapping patches
DPCUM	state variable	#	Cumulative number of damaged plants in the greenhouse, ignoring overlapping patches
DPCUMO	state variable	#	Cumulative number of damaged plants in the greenhouse, ignoring overlapping patches
DPFORL1	state variable	#	Number of damaged plants in foraging domain of L1 larvae

DPFORL2	state variable	#	Number of damaged plants in foraging domain of L2 larvae
DPFORL3	state variable	#	Number of damaged plants in foraging domain of L3 larvae
DPFORL4	state variable	#	Number of damaged plants in foraging domain of L4 larvae
DPFORL5	state variable	#	Number of damaged plants in foraging domain of L5 larvae
DPPATCH	state variable	#	Number of damaged plants in patch
DPTOT	state variable	-	Total number of damaged plants in greenhouse, ignoring overlapping patches
DPTOTO	state variable	-	Total number of damaged plants in greenhouse, adjusted for overlapping patches
DVR	rate variable	d-1	Development rate
DVRBATCH	rate variable	d-1	Development rate of egg batches in oviduct
DVRBATCHI	rate variable	d-1	Development rate of contaminated egg batches in oviduct
DVRBATIT	rate variable	d-1	Development rate of contaminated egg batches in oviduct as function of temperature
DVRBATT	rate variable	d-1	Development rate of egg batches in oviduct as function of temperature
DVREGG	rate variable	d-1	Development rate of egg batches
DVREGGI	rate variable	d-1	Development rate of contaminated egg batches
DVREGGIT	rate variable	d-1	Development rate of contaminated egg batches as function of temperature
DVREGGT	rate variable	d-1	Development rate of egg batches as function of temperature
DVRFEM	rate variable	d-1	Development rate of female moths
DVRFEMI	rate variable	d-1	Development rate of sublethally infected female moths
DVRFEMIT	rate variable	d-1	Development rate of sublethally infected female moths as function of temperature
DVRFEMT	rate variable	d-1	Development rate of female moths as function of temperature
DVRL1	rate variable	d-1	Development rate of L1 larvae
DVRL1I	rate variable	d-1	Development rate of infected L1 larvae
DVRL1IT	rate variable	d-1	Development rate of infected L1 larvae as function of temperature
DVRL1T	rate variable	d-1	Development rate of L1 larvae as function of temperature
DVRL2	rate variable	d-1	Development rate of L2 larvae
DVRL2I	rate variable	d-1	Development rate of infected L2 larvae
DVRL2IT	rate variable	d-1	Development rate of infected L2 larvae as function of temperature
DVRL2T	rate variable	d-1	Development rate of L2 larvae as function of temperature
DVRL2T27	rate variable	d-1	Development rate of L2 larvae at 27°C
DVRL3	rate variable	d-1	Development rate of L3 larvae
DVRL3I	rate variable	d-1	Development rate of infected L3 larvae

DVRL3IT	rate variable	d-1	Development rate of infected L3 larvae as function of temperature
DVRL3T	rate variable	d-1	Development rate of L3 larvae as function of temperature
DVRL3T27	rate variable	d-1	Development rate of L3 larvae at 27°C
DVRL4	rate variable	d-1	Development rate of L4 larvae
DVRL4I	rate variable	d-1	Development rate of infected L4 larvae
DVRL4IT	rate variable	d-1	Development rate of infected L4 larvae as function of temperature
DVRL4T	rate variable	d-1	Development rate of L4 larvae as function of temperature
DVRL4T27	rate variable	d-1	Development rate of L4 larvae at 27°C
DVRL5	rate variable	d-1	Development rate of L5 larvae
DVRL5ID	rate variable	d-1	Development rate of lethally infected L5 larvae
DVRL5IDT	rate variable	d-1	Development rate of lethally infected L5 larvae as function of temperature
DVRL5IS	rate variable	d-1	Development rate of sublethally infected L5 larvae
DVRL5IST	rate variable	d-1	Development rate of sublethally infected L5 larvae as function of temperature
DVRL5T	rate variable	d-1	Development rate of L5 larvae as function of temperature
DVRL5T27	rate variable	d-1	Development rate of L5 larvae at 27°C
DVRMAL	rate variable	d-1	Development rate of male moths
DVRMALI	rate variable	d-1	Development rate of sublethally infected male moths
DVRMALIT	rate variable	d-1	Development rate of sublethally infected male moths as function of temperature
DVRMALT	rate variable	d-1	Development rate of male moths as function of temperature
DVRPUP	rate variable	d-1	Development rate of pupae
DVRPUPI	rate variable	d-1	Development rate of sublethally infected pupae
DVRPUPIT	rate variable	d-1	Development rate of sublethally infected pupae as function of temperature
DVRPUPT	rate variable	d-1	Development rate of pupae as function of temperature
F	state variable	-	Fraction of boxcar contents that is shifted
FCBATL	parameter	-	Fraction contaminated large egg batches deposited by sublethally infected moths
FCBATS	parameter	-	Fraction contaminated small egg batches deposited by sublethally infected moths
FCCUM greenhouse	state variable	m2	Cumulative foliage consumption in
FCEGGL	parameter	-	Maximum fraction of contaminated eggs in a large egg batch
FCEGGS	parameter	-	Maximum fraction of contaminated eggs in a small egg batch
FCLVSL1	state variable	-	Fraction contaminated leaves in L1 foraging domain

FCLVSL2	state variable	-	Fraction contaminated leaves in L2 foraging domain
FCLVSL3	state variable	-	Fraction contaminated leaves in L3 foraging domain
FCLVSL4	state variable	-	Fraction contaminated leaves in L4 foraging domain
FCLVSL5	state variable	-	Fraction contaminated leaves in L5 foraging domain
FCPATCH	state variable	m2	Area foliage consumption per patch
FCR	rate variable	m2 d-1	Larval foliage consumption rate
FCRL1	rate variable	m2 d-1	Foliage consumption rate of L1 larvae
FCRL2	rate variable	m2 d-1	Foliage consumption rate of L2 larvae
FCRL3	rate variable	m2 d-1	Foliage consumption rate of L3 larvae
FCRL4	rate variable	m2 d-1	Foliage consumption rate of L4 larvae
FCRL5	rate variable	m2 d-1	Foliage consumption rate of L5 larvae
FCTOT	state variable	m2	Total area of foliage consumption in greenhouse
FDOWN	auxiliary	-	Fraction of sprayed polyhedra at the bottom of LAI stratum
FINFL5S	parameter	-	Fraction of infected L5 larvae that will become sublethally infected
FINFMAL	state variable	-	Fraction of sublethally infected male moths
FINTIM	initial condition	d	Time when simulation is terminated
FORAR	initial condition	-	Larval foraging domain
FSURVFEM	parameter	-	Fraction surviving female moths
FSURVL1	parameter	-	Fraction surviving L1 larvae in absence of virus
FSURVL2	parameter	-	Fraction surviving L2 larvae in absence of virus
FSURVL3	parameter	-	Fraction surviving L3 larvae in absence of virus
FSURVL4	parameter	-	Fraction surviving L4 larvae in absence of virus
FSURVL5	parameter	-	Fraction surviving L5 larvae in absence of virus
FSURVMAL	parameter	-	Fraction surviving male moths
FSURVPUP	parameter	-	Fraction surviving pupae
FUP	auxiliary	-	Fraction of sprayed polyhedra at the top of LAI statum
FX	parameter	-	Cumulative frequency distribution of egg batch size
IA	state variable	#	Number of individuals in boxcar
IACTRAD	initial condition	# plants	Action radius of infected larvae
IACTRADL1	initial condition	# plants	Action radius of infected L1 larvae
IACTRADL2	initial condition	# plants	Action radius of infected L2 larvae
IACTRADL3	initial condition	# plants	Action radius of infected L3 larvae
IACTRADL4	initial condition	# plants	Action radius of infected L4 larvae
IACTRADL5	initial condition	# plants	Action radius of infected L5 larvae
IAIN	state variable	#	Number of individuals flowing in boxcar train
IATOT	state variable	#	Total number of individuals in boxcar train

IATR	difference	#	Attrition difference of individuals in boxcar train per time step
IATRBAT	difference	#	Attrition difference of egg batches in oviduct per time step
IATRBATI	difference	#	Attrition difference of contaminated egg batches in oviduct per time step
IATREGG	difference	#	Attrition difference of egg batches per time step
IATRFEM	difference	#	Attrition difference of female moths per time step
IATRFEMI	difference	#	Attrition difference of sublethally infected female moths per time step
IATRL1	difference	#	Attrition difference of L1 larvae per time step
IATRL2	difference	#	Attrition difference of L2 larvae per time step
IATRL3R	difference	#	Attrition difference of resistant L3 larvae per time step
IATRL3S	difference	#	Attrition difference of susceptible L3 larvae per time step
IATRL4R	difference	#	Attrition difference of resistant L4 larvae per time step
IATRL4S	difference	#	Attrition difference of susceptible L4 larvae per time step
IATRL5IS	difference	#	Attrition difference of sublethally infected L5 larvae per time step
IATRL5R	difference	#	Attrition difference of resistant L5 larvae per time step
IATRL5S	difference	#	Attrition difference of susceptible L5 larvae per time step
IATRMAL	difference	#	Attrition difference of male moths per time step
IATRMALI	difference	#	Attrition difference of sublethally infected male moths per time step
IATRPUP	difference	#	Attrition difference of pupae per time step
IATRPUPI	difference	#	Attrition difference of sublethally infected pupae per time step
IATRRED	difference	#	Attrition difference per boxcar per time step
IBATCH	state variable	#	Number of egg batches in oviduct per boxcar
IBATCHI	state variable	#	Number of contaminated egg batches in oviduct per boxcar
IBATIN	state variable	#	Number of batches in oviduct flowing in the IBATCH boxcar train
IBATINI	state variable	#	Number of infected batches in oviduct flowing in the IBATCHI boxcar train
IBATNR	parameter	#	Number of deposited egg batches per female moth
IBATOUT	difference	#	Number of batches in oviduct flowing out IBATCH boxcar train
IBATOUTI	difference	#	Number of batches in infected oviduct flowing out IBATCHI boxcar train
IBEDDEP	auxiliary	-	Bed number where egg batch is deposited
IBEDMAX	initial condition	#	Maximum number of beds

IBEDPREF	auxiliary	-	Bed numbers that are prefered to deposit egg batches
ICAD	output	#	Number of cadavers in greenhouse
ICBATSIZE	parameter	-	Critical size of egg batch to be a large egg batch
ICLCLVS	auxiliary	-	Canopy layer where contaminated leaf is located
ICLMAX	parameter	-	Maximum number of canopy layers
ICLVS	auxiliary	-	Help variable for determination of number of contaminated leaves per patch
ICLVSL1	auxiliary	#	Number of contaminated leaves in foraging domain of L1 larvae
ICLVSL1CL	auxiliary	#	Number of contaminated leaves in foraging domain per canopy layer of L1 larvae
ICLVSL2	auxiliary	#	Number of contaminated leaves in foraging domain of L2 larvae
ICLVSL2CL	auxiliary	#	Number of contaminated leaves in foraging domain per canopy layer of L2 larvae
ICLVSL3	auxiliary	#	Number of contaminated leaves in foraging domain of L3 larvae
ICLVSL3CL	auxiliary	#	Number of contaminated leaves in foraging domain per canopy layer of L3 larvae
ICLVSL4	auxiliary	#	Number of contaminated leaves in foraging domain of L4 larvae
ICLVSL4CL	auxiliary	#	Number of contaminated leaves in foraging domain per canopy layer of L4 larvae
ICLVSL5	auxiliary	#	Number of contaminated leaves in foraging domain of L5 larvae
ICLVSL5CL	auxiliary	#	Number of contaminated leaves in foraging domain per canopy layer of L5 larvae
ICLVSMAX	parameter	-	Maximum number of contaminated leaves
ICLVSNR	auxiliary	-	Array containing rank numbers of contaminated leaves per larval foraging domain
ICLVSNRL1	auxiliary	-	Array containing rank numbers of contaminated leaves in an L1 larvae foraging domain
ICLVSNRL2	auxiliary	-	Array containing rank numbers of contaminated leaves in an L2 larvae foraging domain
ICLVSNRL3	auxiliary	-	Array containing rank numbers of contaminated leaves in an L3 larvae foraging domain
ICLVSNRL4	auxiliary	-	Array containing rank numbers of contaminated leaves in an L4 larvae foraging domain
ICLVSNRL5	auxiliary	-	Array containing rank numbers of contaminated leaves in an L5 larvae foraging domain
ICOUNT	auxiliary	-	Counter for boxcar train number

ICTEMP	initial condition	-	Switch for constant or measured greenhouse temperature
IDIFF	auxiliary	-	Help variable for drawing procedure
IDOUBLE	auxiliary	-	Help variable for check for more than one cadavers on a single leaf
IDRFEM step	difference	#	Death difference of female moths per time
IDRFEMI	difference	#	Death difference of sublethally infected female moths per time step
IDRL1I	difference	#	Death difference of infected L1 larvae per time step
IDRL2I	difference	#	Death difference of infected L2 larvae per time step
IDRL3I	difference	#	Death difference of infected L3 larvae per time step
IDRL4I	difference	#	Death difference of infected L4 larvae per time step
IDRL5ID	difference	#	Death difference of lethally infected L5 larvae per time step
IDRMAL	difference	#	Death difference of male moths per time step
IDRMALI	difference	#	Death difference of sublethally infected male moths per time step
IDRWLEAF	function	-	Stochastically chosen leaf number
IDRWLOT	function	-	Stochastically chosen number between minimum and maximum value
IEGG	state variable	#	Number of eggs per boxcar per patch
IEGGI	state variable	#	Number of contaminated eggs per boxcar per patch
IEMP	auxiliary	-	Help variable for resetting of living infected larvae array
IFEM	state variable	#	Number of female moths per boxcar
IFEMI	state variable	#	Number of sublethally infected female moths per boxcar
IHARV	auxiliary	-	Flag for harvest
IHATCH	difference	#	Hatch difference of L1 larvae per patch per time step
IHATCHI	difference	#	Hatch difference of infected L1 larvae per patch per time step
IHIT	auxiliary	-	Help variable for drawing procedure
IHORDIST	initial condition	# plants	Array containing the shape and location of larval foraging domains
IL1	state variable	#	Number of L1 larvae per boxcar per patch
IL1I	state variable	#	Number of infected L1 larvae per boxcar per patch
IL2	state variable	#	Number of L2 larvae per boxcar per patch
IL2I	state variable	#	Number of infected L2 larvae per boxcar per patch
IL3	state variable	#	Number of L3 larvae per boxcar per patch
IL3I	state variable	#	Number of infected L3 larvae per boxcar per patch

IL3R	state variable	#	Number of resistant L3 larvae per boxcar per patch
IL3S	state variable	#	Number of susceptible L3 larvae per boxcar per patch
IL4	state variable	#	Number of L4 larvae per boxcar per patch
IL4I	state variable	#	Number of infected L4 larvae per boxcar per patch
IL4R	state variable	#	Number of resistant L4 larvae per boxcar per patch
IL4S	state variable	#	Number of susceptible L4 larvae per boxcar per patch
IL5	state variable	#	Number of L5 larvae per boxcar per patch
IL5ID	state variable	#	Number of lethally infected L5 larvae per boxcar per patch
IL5IS	state variable	#	Number of sublethally infected L5 larvae per boxcar per patch
IL5R	state variable	#	Number of resistant L5 larvae per boxcar per patch
IL5S	state variable	#	Number of susceptible L5 larvae per boxcar per patch
ILAYER1	state variable	#	Number of LAI strata in lower canopy layer
ILAYER2	state variable	#	Number of LAI strata in middle canopy layer
ILAYER3	state variable	#	Number of LAI strata in upper canopy layer
ILIL	auxiliary	-	Help variable for determination of dying location of newly infected larvae
ILIP	auxiliary	#	Help variable for determination population totals
ILPP	state variable	#	Number of leaves per plant in patch
ILPPBED	state variable	#	Number of leaves per plant in bed
ILVSCL	state variable	#	Number of leaves per canopy layer
IMAL	state variable	#	Number of male moths per boxcar
IMALI	state variable	#	Number of sublethally infected male moths per boxcar
IMAX	auxiliary	-	Help variable for drawing procedure
IMIN	auxiliary	-	Help variable for drawing procedure
IMLTFEM	difference	#	Emergence difference of female moths per time step
IMLTFEMI	difference	#	Emergence difference of sublethally infected female moths per time step
IMLTL2	difference	#	Molt difference of L2 larvae per patch per time step
IMLTL3R	difference	#	Molt difference of resistant L3 larvae per patch per time step
IMLTL3S	difference	#	Molt difference of susceptible L3 larvae per patch per time step
IMLTL4R	difference	#	Molt difference of resistant L4 larvae per patch per time step
IMLTL4S	difference	#	Molt difference of susceptible L4 larvae per patch per time step

IMLTL5R	difference	#	Molt difference of resistant L5 larvae per patch per time step
IMLTL5S	difference	#	Molt difference of susceptible L5 larvae per patch per time step
IMLTMAL	difference	#	Emergence difference of male moths per time step
IMLTMALI	difference	#	Emergence difference of sublethally infected male moths per time step
IMLTPUPI	difference	#	Pupation difference of sublethally infected pupae per patch per time step
IMLTPUPR	difference	#	Pupation difference of resistant pupae per patch per time step
IMLTPUPS	difference	#	Pupation difference of susceptible pupae per patch per time step
IMOULTP	difference	#	Pupation difference of pupae per time step
IMOULTPI	difference	#	Pupation difference of sublethally infected pupae per time step
IN	initial condition	#	Number of boxcars per boxcar train
INBATINFL	initial condition	#	Number of fertilized female moths entering the greenhouse and depositing egg batches per inflight event
INCLVS	initial condition	#	Initial number of contaminated leaves
INEWNR	auxiliary	-	Help variable for harvest
INFEGG	state variable	#	Number of contaminated eggs per egg batch
INFOCLVS	auxiliary	-	Array containing co-ordinates of contaminated leaves
INFOLIL	auxiliary	-	Array containing co-ordinates of infected larvae
INFRHT	difference	#	Infection difference by horizontal transmission of larvae per boxcar train per time step
INFRHTD	difference	#	Infection difference by horizontal transmission of L5 larvae into lethally infected L5 larvae per boxcar train per time step
INFRHTS	difference	#	Infection difference by horizontal transmission of L5 larvae into sublethally infected L5 larvae in boxcar train per time step
INFRREDHT	difference	#	Infection difference by horizontal transmission of uninfected into infected larvae per boxcar per time step
INFRREDHTD	difference	#	Infection difference by horizontal transmission of L5 larvae into lethally infected L5 larvae per boxcar per time step
INFRREDHTS	difference	#	Infection difference by horizontal transmission of L5 larvae into sublethally infected L5 larvae per boxcar per time step
INFRREDS	difference	#	Infection difference by ingestion of sprayed polyhedra of larvae per boxcar per time step
INFRREDSD	difference	#	Infection difference by ingestion of sprayed polyhedra of lethally infected L5 larvae per boxcar per time step

INFRREDSS	difference	#	Infection difference by ingestion of sprayed polyhedra of sublethally infected L5 larvae per boxcar per time step
INFRS	difference	#	Infection difference by ingestion of sprayed polyhedra of individuals per boxcar train per time step
INFRSD	difference	#	Infection difference by ingestion of sprayed polyhedra to lethally infected L5 larvae in boxcar train per time step
INFRSS	difference	#	Infection difference by ingestion of sprayed polyhedra to sublethally infected L5 larvae in boxcar train per time step
INPAT	initial condition	#	Initial number of patches
INPLANT	auxiliary	-	Number of plants per bed
INSETS	auxiliary	-	TTUTIL variable
INSPRAY	initial condition	#	Number of virus applications during simulation
INSTAR	auxiliary	-	Instar of larvae
INTATR	auxiliary	#	Help variable for determination of attrition difference
INTINFHT	auxiliary	#	Help variable for determination of infection difference by horizontal transmission
INTINFS	auxiliary	#	Help variable for determination of infection difference by ingestion of sprayed polyhedra
IOLDLAR	auxiliary	-	Help variable for co-ordinates of living infected larvae
IOUTFL	difference	#	Outflow difference of individuals from final boxcar per time step
IOUTFLFEM	difference	#	Outflow difference of pupae developing into female moths in final boxcar per time step
IOUTFLMAL	difference	#	Outflow difference of pupae developing into male moths in final boxcar per time step
IOUTFLR	difference	#	Outflow difference of resistant individuals from final boxcar per time step
IOUTFLS	difference	#	Outflow difference of susceptible individuals from final boxcar per time step
IP	auxiliary	-	Pointer for patch number
ITOT1I	state variable	#	Total number of infected L1 larvae
ITOT2I	state variable	#	Total number of infected L2 larvae
ITOT3I	state variable	#	Total number of infected L3 larvae
ITOT4I	state variable	#	Total number of infected L4 larvae
ITOT5I	state variable	#	Total number of infected L5 larvae
IPATCUM	output	#	Cumulative number of patches in greenhouse
IPATDAT	initial condition	-	Array containing co-ordinates of patches in greenhouse
IPATEGG	state variable	#	Number of egg batches in boxcar train per patch
IPATEGGI	state variable	#	Number of contaminated egg batches in boxcar train per patch
IPATL1	state variable	#	Number of L1 larvae in boxcar train per patch

IPATL1I	state variable	#	Number of infected L1 larvae in boxcar train per patch
IPATL2	state variable	#	Number of L2 larvae in boxcar train per patch
IPATL2I	state variable	#	Number of infected L2 larvae in boxcar train per patch
IPATL3I	state variable	#	Number of infected L3 larvae in boxcar train per patch
IPATL3R	state variable	#	Number of resistant L3 larvae in boxcar train per patch
IPATL3S	state variable	#	Number of susceptible L3 larvae in boxcar train per patch
IPATL4I	state variable	#	Number of infected L4 larvae in boxcar train per patch
IPATL4R	state variable	#	Number of resistant L4 larvae in boxcar train per patch
IPATL4S	state variable	#	Number of susceptible L4 larvae in boxcar train per patch
IPATL5ID	state variable	#	Number of lethally infected L5 larvae per boxcar
IPATL5IS	state variable	#	Number of sublethally infected L5 larvae per boxcar
IPATL5R	state variable	#	Number of resistant L5 larvae in boxcar train per patch
IPATL5S	state variable	#	Number of susceptible L5 larvae in boxcar train per patch
IPATLOW	auxiliary	-	Pointer for lowest patch number containing larvae
IPCLVS	auxiliary	-	Pointer for number of contaminated leaves
IPLANTL1	initial condition	#	Number of plants in foraging domain of L1 larvae
IPLANTL2	initial condition	#	Number of plants in foraging domain of L2 larvae
IPLANTL3	initial condition	#	Number of plants in foraging domain of L3 larvae
IPLANTL4	initial condition	#	Number of plants in foraging domain of L4 larvae
IPLANTL5	initial condition	#	Number of plants in foraging domain of L5 larvae
IPLLOW	auxiliary	-	Pointer for infected larvae with lowest rank number
IPLUP	auxiliary	-	Pointer for infected larvae with highest rank number
IPMAX	parameter	#	Maximum number of patches in greenhouse
IPOINM	parameter	#	Maximum value for contaminated leaves and infected larvae
IPPRDIF	auxiliary	# plants	Distance between contaminated leaf and egg batch
IPPRLOW	auxiliary	-	Lowest possible plant per row co-ordinate for location of infected larvae
IPPRMAX	initial condition	#	Maximum number of plants per row in bed

IPPRUP	auxiliary	-	Highest possible plant per row co-ordinate for location of infected larvae
IPSOURCE	auxiliary	#	Number of contaminated leaves in larval foraging domain
IPUP	state variable	#	Number of pupae per boxcar
IPUPI	state variable	#	Number of sublethally infected pupae per boxcar
IRA	difference	#	Integer value of number of shifted individuals to next boxcar per time step
IREDATR	difference	#	Attrition difference of individuals in boxcar per time step
IREDINFRHT	difference	#	Infection difference by horizontal transmission of individuals in boxcar per time step
IREDINFRS	difference	#	Infection difference by ingestion of sprayed polyhedra of individuals in boxcar per time step
IRL1HT	difference	#	Infection difference by horizontal transmission of L1 larvae per time step
IRL1S	difference	#	Infection difference by ingestion of sprayed polyhedra of L1 larvae per time step
IRL2HT	difference	#	Infection difference by horizontal transmission of L2 larvae per time step
IRL2S	difference	#	Infection difference by ingestion of sprayed polyhedra of L2 larvae per time step
IRL3RHT	difference	#	Infection difference by horizontal transmission of resistant L3 larvae per time step
IRL3RS	difference	#	Infection difference by ingestion of sprayed polyhedra of resistant L3 larvae per time step
IRL3SHT	difference	#	Infection difference by horizontal transmission of susceptible L3 larvae per time step
IRL3SS	difference	#	Infection difference by ingestion of sprayed polyhedra of susceptible L3 larvae per time step
IRL4RHT	difference	#	Infection difference by horizontal transmission of resistant L4 larvae per time step
IRL4RS	difference	#	Infection difference by ingestion of sprayed polyhedra of resistant L4 larvae per time step
IRL4SHT	difference	#	Infection difference by horizontal transmission of susceptible L4 larvae per time step
IRL4SS	difference	#	Infection difference by ingestion of sprayed polyhedra of susceptible L4 larvae per time step
IRL5RHTD	difference	#	Infection difference by horizontal transmission of resistant L5 larvae into lethally infected L5 larvae per time step
IRL5RHTS	difference	#	Infection difference by horizontal transmission of resistant L5 larvae into sublethally infected L5 larvae per time step
IRL5RHTSAR	difference	#	Array of infection differences by horizontal transmission of resistant L5 larvae into

			sublethally infected L5 larvae per time step
IRL5RSD	difference	#	Infection difference by ingestion of sprayed polyhedra of resistant L5 larvae into lethally infected L5 larvae per time step
IRL5RSS	difference	#	Infection difference by ingestion of sprayed polyhedra of resistant L5 larvae into sublethally infected L5 larvae per time step
IRL5RSSAR	difference	#	Array of infection differences by ingestion of sprayed polyhedra of resistant L5 larvae into sublethally infected L5 larvae per time step
IRL5SHTD	difference	#	Infection difference by horizontal transmission of susceptible L5 larvae into lethally infected L5 larvae per time step
IRL5SHTS	difference	#	Infection difference by horizontal transmission of susceptible L5 larvae into sublethally infected L5 larvae per time step
IRL5SHTSAR	difference	#	Array of infection differences by horizontal transmission of susceptible L5 larvae into sublethally infected L5 larvae per time step
IRL5SSD	difference	#	Infection difference by ingestion of sprayed polyhedra of susceptible L5 larvae into lethally infected L5 larvae per time step
IRL5SSS	difference	#	Infection difference by ingestion of sprayed polyhedra of susceptible L5 larvae into sublethally infected L5 larvae per time step
IRL5SSSAR	difference	#	Array of Infection differences by ingestion of sprayed polyhedra of susceptible L5 larvae into sublethally infected L5 larvae per time step
IROUND	function	-	Stochastically rounded real value into integer
IROWDIF	auxiliary	#	Distance between contaminated leaf and egg batch
IROWLOW	auxiliary	-	Lowest possible row co-ordinate for location of infected larvae
IROWMAX	initial condition	#	Maximum number plant rows in bed
IROWUP	auxiliary	-	Highest possible row co-ordinate for location
of			
			infected larvae
ISET	auxiliary	-	TTUTIL variable
ISOURCE	auxiliary	-	Contaminated leaf that is source of infection
by			
			horizontal transmission
ISTOCH	initial condition	-	Switch for stochasticity in egg batch size
ITASK	auxiliary	-	Task that subroutine should perform
ITOT	output	#	Total number of living individuals in and outside patches
ITOTAL	auxiliary	#	Help variable for population totals
ITOTBAT	state variable	#	Total number of egg batches in oviduct in greenhouse
ITOTBATI	state variable	#	Total number of contaminated egg batches in

			oviduct in greenhouse
ITOTEGG	output	#	Total number of egg batches in greenhouse
ITOTEGGI	output	#	Total number of contaminated egg batches in greenhouse
ITOTFEM	state variable	#	Total number of female moths in greenhouse
ITOTFEMI	state variable	#	Total number of sublethally infected female moths in greenhouse
ITOTIP	output	#	Total number of living individuals in patches
ITOTL1	output	#	Total number of L1 larvae in greenhouse
ITOTL1I	output	#	Total number of infected L1 larvae in greenhouse
ITOTL2	output	#	Total number of L2 larvae in greenhouse
ITOTL2I	output	#	Total number of infected L2 larvae in greenhouse
ITOTL3	output	#	Total number of L3 larvae in greenhouse
TOTL3I	output	#	Total number of infected L3 larvae in greenhouse
ITOTL4	output	#	Total number of L4 larvae in greenhouse
ITOTL4I	output	#	Total number of infected L4 larvae in greenhouse
ITOTL5	output	#	Total number of L5 larvae in greenhouse
ITOTL5ID	output	#	Total number of lethally infected L5 larvae in greenhouse
ITOTL5IS	output	#	Total number of sublethally infected L5 larvae in greenhouse
ITOTLIL	auxiliary	#	Total number of infected larvae per instar in greenhouse
ITOTMAL	state variable	#	Total number of male moths in greenhouse
ITOTMALI	state variable	#	Total number of sublethally infected male moths in greenhouse
ITOTOP	output	#	Total number of living individuals outside patches
ITOTPUP	state variable	#	Total number of pupae in greenhouse
ITOTPUPI	state variable	#	Total number of sublethally infected pupae in greenhouse
K	initial condition	m2.m-2	Extinction coefficient of crop for spray
LAI	state variable	m2.m-2	Leaf area index of crop in patch
LAI0	parameter	m2.m-2	Initial leaf area index of crop
LAIBED	state variable	m2.m-2	Leaf area index of crop in bed
LAIHARV	initial condition	m2.m-2	Leaf area index of crop at harvest
LAIMAX	parameter	m2.m-2	Maximum leaf area index of crop
LAIPREF	parameter	m2.m-2	Maximum leaf area index of crop to be preferred for egg deposition
LOT	auxiliary	-	Random number between 0 and 1
LPDL1	parameter	d-1	Mean number of leaf visits of L1 larvae per day
LPDL1T27	auxiliary	d-1	Leaf visit rate of L1 larvae at 27°C
LPDL2	parameter	d-1	Mean number of leaf visits of L2 larvae per day
LPDL2T27	auxiliary	d-1	Leaf visit rate of L2 larvae at 27°C

LPDL3	parameter	d-1	Mean number of leaf visits of L3 larvae per day
LPDL3T27	auxiliary	d-1	Leaf visit rate of L3 larvae at 27°C
LPDL4	parameter	d-1	Mean number of leaf visits of L4 larvae per day
LPDL4T27	auxiliary	d-1	Leaf visit rate of L4 larvae at 27°C
LPDL5	parameter	d-1	Mean number of leaf visits of L5 arvae per day
LPDL5T27	auxiliary	d-1	Leaf visit rate of L5 larvae at 27°C
LPP0	parameter	#	Initial number of leaves per plant in patch
LPPMAX	parameter	#	Maximum number of leaves per plant
OUTFL	difference	#	Outflow difference of individuals from final boxcar per time step
PDEFORL1	state variable	#	Number of plant damage events in foraging domain of L1 larvae
PDEFORL2	state variable	#	Number of plant damage events in foraging domain of L2 larvae
PDEFORL3	state variable	#	Number of plant damage events in foraging domain of L3 larvae
PDEFORL4	state variable	#	Number of plant damage events in foraging domain of L4 larvae
PDEFORL5	state variable	#	Number of plant damage events in foraging domain of L5 larvae
PDERFORL1	rate variable	# d-1	Plant damage event rate in foraging domain of L1 larvae
PDERFORL2	rate variable	# d-1	Plant damage event rate in foraging domain of L2 larvae
PDERFORL3	rate variable	# d-1	Plant damage event rate in foraging domain of L3 larvae
PDERFORL4	rate variable	# d-1	Plant damage event rate in foraging domain of L4 larvae
PDERFORL5	rate variable	# d-1	Plant damage event rate in foraging domain of L5 larvae
PDERL1	rate variable	# d-1	Plant damage rate of L1 larvae
PDERL2	rate variable	# d-1	Plant damage rate of L2 larvae
PDERL3	rate variable	# d-1	Plant damage rate of L3 larvae
PDERL4	rate variable	# d-1	Plant damage rate of L4 larvae
PDERL5	rate variable	# d-1	Plant damage rate of L5 larvae
PHD	state variable	#	Number of polyhedra per LAI stratum per patch
PHDBED	state variable	#	Number of polyhedra per LAI stratum per bed
PHDIRBED	rate variable	#.d-1	Inactivation rate of polyhedra per LAI stratum of bed
PHDIRLAR	rate variable	#.d-1	Inactivation rate of polyhedra produced in larvae
PHDL1	state variable	#	Number of polyhedra in lower canopy layer
PHDL2	state variable	#	Number of polyhedra in middle canopy layer
PHDL3	state variable	#	Number of polyhedra in upper canopy layer
PHDLARTOT	state variable	#	Total number of polyhedra produced in larvae present in the greenhouse
PHDPROD	parameter	# larvae-1	Array containing polyhedra production per

			larval instar
PHDSPR	initial condition	#.m-2 ground	Density of sprayed polyhedra of virus application
PHDTOTBED	state variable	#	Total number of polyhedra of LAI strata per bed
PL1	parameter	#-1	Infection chance per ingested polyhedron for L1 larvae
PL2	parameter	#-1	Infection chance per ingested polyhedron for L2 larvae
PL3R	parameter	#-1	Infection chance per ingested polyhedron for resistant L3 larvae
PL3S	parameter	#-1	Infection chance per ingested polyhedron for susceptible L3 larvae
PL4R	parameter	#-1	Infection chance per ingested polyhedron for resistant L4 larvae
PL4S	parameter	#-1	Infection chance per ingested polyhedron for susceptible L4 larvae
PL5R	parameter	#-1	Infection chance per ingested polyhedron for resistant L5 larvae
PL5S	parameter	#-1	Infection chance per ingested polyhedron for susceptible L5 larvae
POLDEN	state variable	#.m-2	Polyhedra density per canopy layer
RA	rate variable	#	Real value of number of shifted individuals to next boxcar per time step
RATR	rate variable	d-1	Relative attrition rate
RATRFEM	rate variable	d-1	Relative attrition rate of female moths
RATRL1	rate variable	d-1	Relative attrition rate of L1 larvae
RATRL2	rate variable	d-1	Relative attrition rate of L2 larvae
RATRL3	rate variable	d-1	Relative attrition rate of L3 larvae
RATRL4	rate variable	d-1	Relative attrition rate of L4 larvae
RATRL5	rate variable	d-1	Relative attrition rate of L5 larvae
RATRMAL	rate variable	d-1	Relative attrition rate of male moths
RATRPUP	rate variable	d-1	Relative attrition rate of pupae
REALATR	auxiliary	#	Help variable for determination of attrition difference
REALINFHT	auxiliary	#	Help variable for determination of infection difference by horizontal transmission
REALINFS	auxiliary	#	Help variable for determination of infection difference by ingestion of sprayed polyhedra
RESPHD	parameter	#.m-2	Residual density of infectious polyhedra
REST	auxiliary	-	Help variable for determination of number of leaves per canopy layer
RGRLAI	parameter	d-1	Relative growth rate of LAI
RGRLPP	parameter	d-1	Relative growth rate of leaves per plant
RINACTR	parameter	d-1	Relative polyhedra inactivation rate
RINFRHT	auxiliary	d-1	Relative infection rate by horizontal transmission
RINFRS	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra
RIRCL1	auxiliary	d-1	Relative infection rate by horizontal

			transmission per canopy layer of L1larvae
RIRCL2	auxiliary	d-1	Relative infection rate by horizontal transmission per canopy layer of L2 larvae
RIRCL3	auxiliary	d-1	Relative infection rate by horizontal transmission per canopy layer of L3 larvae
RIRCL4	auxiliary	d-1	Relative infection rate by horizontal transmission per canopy layer of L4 larvae
RIRCL5	auxiliary	d-1	Relative infection rate by horizontal transmission per canopy layer of L5 larvae
RIRL1HT	auxiliary	d-1	Relative infection rate by horizontal transmission of L1 larvae
RIRL1S	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra of L1 larvae
RIRL2HT	auxiliary	d-1	Relative infection rate by horizontal transmission of L2 larvae
RIRL2S	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra of L2 larvae
RIRL3HT	auxiliary	d-1	Relative infection rate by horizontal transmission of L3 larvae
RIRL3RS	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra of resistant L3 larvae
RIRL3SS	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra of susceptible L3 larvae
RIRL4HT	auxiliary	d-1	Relative infection rate by horizontal transmission of L4 larvae
RIRL4RS	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra of resistant L4 larvae
RIRL4SS	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra of susceptible L4 larvae
RIRL5HT	auxiliary	d-1	Relative infection rate by horizontal transmission of L5 larvae
RIRL5RS	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra of resistant L5 larvae
RIRL5SS	auxiliary	d-1	Relative infection rate by ingestion of sprayed polyhedra of susceptible L5 larvae
RLATR	difference	#	Attrition difference per time step
RLINFHT	difference	#	Infection difference by horizontal transmission of larvae in boxcar per time step
RLINFS	difference	#	Infection difference by ingestion of sprayed polyhedra of larvae in boxcar per time step
SDBAT	auxiliary	d-1	Standard deviation of development rate of egg batches in oviduct
SDBATI	auxiliary	d-1	Standard deviation of development rate of contaminated egg batches in oviduct
SDEGG	auxiliary	d-1	Standard deviation of development rate of egg batches
SDEGGI	auxiliary	d-1	Standard deviation of development rate of contaminated egg batches
SDFEM	auxiliary	d-1	Standard deviation of development rate of female moths

SDFEMI	auxiliary	d-1	Standard deviation of development rate of sublethally infected female moths
SDL1	auxiliary	d-1	Standard deviation of development rate of L1 larvae
SDL1I	auxiliary	d-1	Standard deviation of development rate of infected L1 larvae
SDL2	auxiliary	d-1	Standard deviation of development rate of L2 larvae
SDL2I	auxiliary	d-1	Standard deviation of development rate of infected L2 larvae
SDL3	auxiliary	d-1	Standard deviation of development rate of L3 larvae
SDL3I	auxiliary	d-1	Standard deviation of development rate of infected L3 larvae
SDL4	auxiliary	d-1	Standard deviation of development rate of L4 larvae
SDL4I	auxiliary	d-1	Standard deviation of development rate of infected L4 larvae
SDL5	auxiliary	d-1	Standard deviation of development rate of L5 larvae
SDL5ID	auxiliary	d-1	Standard deviation of development rate of lethally infected L5 larvae
SDL5IS	auxiliary	d-1	Standard deviation of development rate of sublethally infected L5 larvae
SDMAL	auxiliary	d-1	Standard deviation of development rate of male moths
SDMALI	auxiliary	d-1	Standard deviation of development rate of sublethally infected male moths
SDPUP	auxiliary	d-1	Standard deviation of development rate of pupae
SDPUPI	auxiliary	d-1	Standard deviation of development rate of sublethally infected pupae
SEXRAT	parameter	-	Sexratio
SUMW	auxiliary	-	Help variable for determination of weighed sum
TBATINFL	initial condition	-	Moment of inflight of fertilized female moths in greenhouse
TEMP	initial condition	degree C	Temperature
TEMPT	initial condition	degree C	Array containing hourly measured temperatures in greenhouse
TIME	initial condition	d	Time
TSPRAY	initial condition	d	Time of virus application
VERTDIST	parameter	-	Fraction larvae per canopy layer
X	parameter	-	X-axis of cumulative frequency distribution of egg batch size

Appendix III: Listing of BACSIM

```
*-----------------------------------------------------------------------*
*     BACSIM                                                            *
*                                                                       *
*     Author : F.J.J.A. Bianchi                                         *
*              Laboratoria of Virology and Theoretical Production       *
*              Ecology,                                                 *
*              Wageningen University.                                   *
*              Correspondence adress:                                   *
*              W. van der Werf                                          *
*              Laboratory of Crop and Weed Ecology                      *
*              Haarweg 333                                              *
*              6709 RZ Wageningen, The Netherlands                      *
*     Date   : 20-10-2000                                               *
*     Purpose: The program BACSIM simulates the population dynamics     *
*              of the beet armyworm (Spodoptera exigua) in greenhouse   *
*              chrysanthemum and the biological control of this pest    *
*              insect with baculoviruses. Beet armyworm larvae that     *
*              ingest a lethal dose of polyhedra become infected,       *
*              produce new polyhedra and die. At the end of the virus   *
*              infection, larvae disintegrate and spread the newly      *
*              formed virus particles. There are three major routes     *
*              of virus infection: virus infection by ingestion of      *
*              sprayed virus, virus infection by horizontal             *
*              transmission (ingestion of polyhedra originating from    *
*              a cadaver) and vertical transmission (transmission of    *
*              virus from adult to progeny). The location of egg        *
*              batches and foraging domains of larvae as well as the    *
*              location of dead infected larvae in the crop are stored *
*              using 4 co-ordinates (bed (=group of plants of the same  *
*              age), rownumber, plantnumber per row and leafnumber per  *
*              plant). Insect development is simulated for individual   *
*              patches with integer boxcar trains. Separate boxcar      *
*              trains are used for uninfected and infected insects.     *
*              The integration order is: initialization (ITASK=1),      *
*              forcing functions and auxiliary variables (ITASK=2),     *
*              differences and rate variables (ITASK=3), generation of  *
*              output (ITASK=4), state variables (ITASK=5) and shift    *
*              (ITASK=6).                                               *
*     Subroutines called:                                               *
*     from TTUTIL: FOPENS,RDSETS,RDFROM                                 *
*     own:         FILES  (file management)                             *
*                  INITIM (initialization of time statements)           *
*                  SPADIS (spatial distribution of Spodoptera exigua)   *
*                  SEDEV  (Spodoptera exigua development)               *
*                  VIRUS  (quantity and distribution of sprayed virus)  *
*                  CROP   (crop development)                            *
*                  RIRHT  (relative infection rate by horizontal       *
*                          transmission)                                *
*                  RIRSPR (relative infection rate by ingestion of     *
*                          sprayed virus)                               *
*-----------------------------------------------------------------------*
      PROGRAM BACSIM

      PARAMETER (IPMAX = 200)
      PARAMETER (ICLMAX = 3)

*     Declaration of global variables
      INTEGER   IPATCUM,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $          IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,
     $          IPATL4I,ITIL5D,ITIL5S,IDRL1I,IDRL2I,IDRL3I,IDRL4I,
     $          IDRL5ID,IMLTMAL,IMLTFEM,IMLTMALI,IMLTFEMI,IDRMAL,
     $          IDRFEM,IDRMALI,IDRFEMI,IRL1HT,IRL2HT,IRL3SHT,
     $          IRL3RHT,IRL4SHT,IRL4RHT,IRL5SHTS,IRL5RHTS,IRL5SHTD,
     $          IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,IRL4RS,
     $          IRL5SSS,IRL5RSS,IRL5SSD,IRL5RSD,ITOTMAL,ITOTFEM,
     $          ITOTMALI,ITOTFEMI,IBATIN,IBATINI,IBATOUT,IBATOUTI,
     $          IHATCHI,IPATDAT,ILPP,IPATLOW,IHARV,ITOTPUP,ITOTPUPI,
     $          IPLANTL1,IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,INPLANT
      REAL      DELT,TIME,FINTIM,FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,
     $          FCLVSL5,RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT,
     $          LAI,RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,RIRL4RS,
     $          RIRL5SS,RIRL5RS,VERTDIST,LAIBED,PHD,DELTLAI,DVRL1,
     $          DVRL2,DVRL3,DVRL4,DVRL5,PHDLARTOT,LPDL1,LPDL2,LPDL3,
     $          LPDL4,LPDL5
      DIMENSION IPATL1(IPMAX),IPATL2(IPMAX),IPATL3S(IPMAX),
     $          IPATL3R(IPMAX),IPATL4S(IPMAX),IPATL4R(IPMAX),
     $          IPATL5S(IPMAX),IPATL5R(IPMAX),IPATL1I(IPMAX),
     $          IPATL2I(IPMAX),IPATL3I(IPMAX),IPATL4I(IPMAX),
     $          ITIL5D(IPMAX),ITIL5S(IPMAX),IDRL1I(IPMAX),
     $          IDRL2I(IPMAX),IDRL3I(IPMAX),IDRL4I(IPMAX),
     $          IDRL5ID(IPMAX),FCLVSL1(IPMAX,3),FCLVSL2(IPMAX,ICLMAX),
     $          FCLVSL3(IPMAX,ICLMAX),FCLVSL4(IPMAX,ICLMAX),
     $          FCLVSL5(IPMAX,ICLMAX),RIRL1HT(IPMAX),RIRL2HT(IPMAX),
     $          RIRL3HT(IPMAX),RIRL4HT(IPMAX),RIRL5HT(IPMAX),
     $          LAI(IPMAX),IRL1HT(IPMAX),IRL2HT(IPMAX),IRL3SHT(IPMAX),
     $          IRL3RHT(IPMAX),IRL4SHT(IPMAX),IRL4RHT(IPMAX),
     $          IRL5SHTS(IPMAX),IRL5RHTS(IPMAX),IRL5SHTD(IPMAX),
     $          IRL5RHTD(IPMAX),RIRL1S(IPMAX),RIRL2S(IPMAX),
     $          RIRL3SS(IPMAX),RIRL3RS(IPMAX),RIRL4SS(IPMAX),
     $          RIRL4RS(IPMAX),RIRL5SS(IPMAX),RIRL5RS(IPMAX),
     $          IRL1S(IPMAX),IRL2S(IPMAX),IRL3SS(IPMAX),IRL3RS(IPMAX),
     $          IRL4SS(IPMAX),IRL4RS(IPMAX),IRL5SSS(IPMAX),
     $          IRL5RSS(IPMAX),IRL5SSD(IPMAX),IRL5RSD(IPMAX),
     $          VERTDIST(3,5),IHATCHI(IPMAX),IPATDAT(3,IPMAX),
     $          ILPP(IPMAX),LAIBED(IPMAX),PHD(IPMAX,60)
```

```
*     Initialize TTSELECT output file
      CALL OUTDAT(1,20,'TIME',0.)

*     TTUTIL commands
      CALL FOPENS (40,'RERUNS.LOG','NEW','DEL')
      CALL RDSETS (10,40,'RERUNS.DAT',INSETS)

*     TTUTIL loop
      DO 30 ISET = 0,INSETS

*       TTUTIL commands
        CALL RDFROM(ISET,.TRUE.)

**********************************************************
*       Initial part
**********************************************************
        ITASK = 1

*       Initialization of time variables
        CALL INITIM(ITASK,
     $                  TIME,DELT,FINTIM)

*       Initialization of SPADIS variables
        CALL SPADIS(ITASK,IPATLOW,IPATCUM,IRL1HT,IRL2HT,
     $              IRL3SHT,IRL3RHT,IRL4SHT,IRL4RHT,IRL5SHTD,
     $              IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $              IRL4RS,IRL5SSD,IRL5RSD,IDRL1I,IDRL2I,IDRL3I,
     $              IDRL4I,IDRL5ID,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $              ITIL5D,IBATOUT,IBATOUTI,IHATCHI,ILPP,LAIBED,IHARV,
     $                  FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $                  VERTDIST,IPATDAT,IPLUP,IPCLVS,ICAD,IPLANTL1,
     $                  IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,INPLANT,
     $                  PHDLARTOT)

*       Initialization of SEDEV variables
        CALL SEDEV(ITASK,TIME,DELT,IHARV,IPATDAT,RIRL1HT,RIRL2HT,
     $             RIRL3HT,RIRL4HT,RIRL5HT,RIRL1S,RIRL2S,RIRL3SS,
     $             RIRL3RS,RIRL4SS,RIRL4RS,RIRL5SS,RIRL5RS,
     $                 IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,IPATL4R,
     $                 IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $                 ITIL5S,ITIL5D,IDRL1I,IDRL2I,IDRL3I,IDRL4I,
     $                 IDRL5ID,IMLTMAL,IMLTFEM,IMLTMALI,IMLTFEMI,
     $                 IDRMAL,IDRFEM,IDRMALI,IDRFEMI,IRL1HT,IRL2HT,
     $                 IRL3SHT,IRL3RHT,IRL4SHT,IRL4RHT,IRL5SHTS,
     $                 IRL5RHTS,IRL5SHTD,IRL5RHTD,IRL1S,IRL2S,IRL3SS,
     $                 IRL3RS,IRL4SS,IRL4RS,IRL5SSS,IRL5RSS,IRL5SSD,
     $                 IRL5RSD,ITOTMAL,ITOTFEM,ITOTMALI,ITOTFEMI,
     $                 IBATIN,IBATINI,IBATOUT,IBATOUTI,IHATCHI,
     $                 IPATLOW,IPATCUM,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                 DVRL1I,DVRL2I,DVRL3I,DVRL4I,DVRL5ID,ITOTPUP,
     $                  ITOTPUPI)

*       Initialization of VIRUS variables
        CALL VIRUS(ITASK,DELT,IHARV,IPATLOW,IPATCUM,IPATDAT,
     $              LAIBED,IBATOUT,IBATOUTI,DELTLAI,TIME,
     $                  PHD,PHDLARTOT)

*       Initialization of CROP variables
        CALL CROP(ITASK,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,IPATL4R,
     $             IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $             ITIL5S,ITIL5D,DELT,IPATCUM,IPATDAT,IBATOUT,IBATOUTI,
     $             IPATLOW,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,DVRL1I,DVRL2I,
     $             DVRL3I,DVRL4I,IPLANTL1,IPLANTL2,IPLANTL3,
     $             IPLANTL4,IPLANTL5,INPLANT,
     $                 LAI,ILPP,LAIBED,IHARV,FCCUM,LPDL1,LPDL2,
     $                 LPDL3,LPDL4,LPDL5)

*       Initialization of RIRHT variables
        CALL RIRHT(ITASK,FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $              DELT,IPATCUM,VERTDIST,IPATLOW,LPDL1,LPDL2,
     $              LPDL3,LPDL4,LPDL5,
     $                  RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT)

*       Initialization of RIRS variables
        CALL RIRSPR(ITASK,LAI,PHD,IPATCUM,VERTDIST,IPATLOW,
     $               DELTLAI,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                   RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,
     $                   RIRL4RS,RIRL5SS,RIRL5RS)


*       Dynamic loop
10      DO 20 WHILE (TIME.LE.FINTIM)

********************************************************
*           forcing functions and auxiliary variables
********************************************************
            ITASK = 2

*           Calculation of forcing functions for Spodoptera exigua
*           development
            CALL SEDEV(ITASK,TIME,DELT,IHARV,IPATDAT,
     $                  RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT,
     $                  RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,RIRL4RS,
     $                  RIRL5SS,RIRL5RS,
     $                      IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                      IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,
     $                      IPATL3I,IPATL4I,ITIL5S,ITIL5D,IDRL1I,
     $                      IDRL2I,IDRL3I,IDRL4I,IDRL5ID,IMLTMAL,IMLTFEM,
     $                      IMLTMALI,IMLTFEMI,IDRMAL,IDRFEM,IDRMALI,
     $                      IDRFEMI,IRL1HT,IRL2HT,IRL3SHT,IRL3RHT,
     $                    IRL4SHT,IRL4RHT,IRL5SHTS,IRL5RHTS,IRL5SHTD,
     $                    IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                    IRL4RS,IRL5SSS,IRL5RSS,IRL5SSD,IRL5RSD,
     $                    ITOTMAL,ITOTFEM,ITOTMALI,ITOTFEMI,IBATIN,
     $                    IBATINI,IBATOUT,IBATOUTI,IHATCHI,IPATLOW,
     $                    IPATCUM,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                    DVRL1I,DVRL2I,DVRL3I,DVRL4I,DVRL5ID,ITOTPUP,
     $                    ITOTPUPI)

*         Calculation of auxiliary variables for Spodoptera exigua
*         spatial distribution
          CALL SPADIS(ITASK,IPATLOW,IPATCUM,IRL1HT,IRL2HT,
     $                 IRL3SHT,IRL3RHT,IRL4SHT,IRL4RHT,IRL5SHTD,
     $                 IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                 IRL4RS,IRL5SSD,IRL5RSD,IDRL1I,IDRL2I,IDRL3I,
     $                 IDRL4I,IDRL5ID,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $                 ITIL5D,IBATOUT,IBATOUTI,IHATCHI,ILPP,LAIBED,
     $                 IHARV,
     $                     FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $                     VERTDIST,IPATDAT,IPLUP,IPCLVS,ICAD,IPLANTL1,
     $                     IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,INPLANT,
     $                     PHDLARTOT)

*         Calculation of auxiliary variables for VIRUS variables
          CALL VIRUS(ITASK,DELT,IHARV,IPATLOW,IPATCUM,IPATDAT,
     $                LAIBED,IBATOUT,IBATOUTI,DELTLAI,TIME,
     $                    PHD,PHDLARTOT)

*         Calculation of auxiliary variables for CROP variables
          CALL CROP(ITASK,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $               IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,
     $               IPATL4I,ITIL5S,ITIL5D,DELT,IPATCUM,IPATDAT,
     $               IBATOUT,IBATOUTI,IPATLOW,DVRL1,DVRL2,DVRL3,
     $               DVRL4,DVRL5,DVRL1I,DVRL2I,DVRL3I,DVRL4I,
     $               IPLANTL1,IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,
     $               INPLANT,
     $                   LAI,ILPP,LAIBED,IHARV,FCCUM,LPDL1,LPDL2,
     $                   LPDL3,LPDL4,LPDL5)

********************************************************
*         differences and rate variables
********************************************************
          ITASK = 3

*         Inactivation rate of sprayed polyhedra in LAI strata
          CALL VIRUS(ITASK,DELT,IHARV,IPATLOW,IPATCUM,IPATDAT,
     $                LAIBED,IBATOUT,IBATOUTI,DELTLAI,TIME,
     $                    PHD,PHDLARTOT)

*         Relative infection rate by ingestion of sprayed polyhedra
```

```
            CALL RIRSPR(ITASK,LAI,PHD,IPATCUM,VERTDIST,IPATLOW,
     $                 DELTLAI,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                     RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,
     $                     RIRL4RS,RIRL5SS,RIRL5RS)

*           Relative infection rate by horizontal transmission
            CALL RIRHT(ITASK,FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $                 DELT,IPATCUM,VERTDIST,IPATLOW,LPDL1,LPDL2,
     $                 LPDL3,LPDL4,LPDL5,
     $                     RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT)

*           Development rate of Spodoptera exigua
            CALL SEDEV(ITASK,TIME,DELT,IHARV,IPATDAT,
     $                 RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT,
     $                 RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,RIRL4RS,
     $                 RIRL5SS,RIRL5RS,
     $                     IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                     IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,
     $                     IPATL3I,IPATL4I,ITIL5S,ITIL5D,IDRL1I,
     $                     IDRL2I,IDRL3I,IDRL4I,IDRL5ID,IMLTMAL,IMLTFEM,
     $                     IMLTMALI,IMLTFEMI,IDRMAL,IDRFEM,IDRMALI,
     $                     IDRFEMI,IRL1HT,IRL2HT,IRL3SHT,IRL3RHT,
     $                     IRL4SHT,IRL4RHT,IRL5SHTS,IRL5RHTS,IRL5SHTD,
     $                     IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                     IRL4RS,IRL5SSS,IRL5RSS,IRL5SSD,IRL5RSD,
     $                     ITOTMAL,ITOTFEM,ITOTMALI,ITOTFEMI,IBATIN,
     $                     IBATINI,IBATOUT,IBATOUTI,IHATCHI,IPATLOW,
     $                     IPATCUM,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                     DVRL1I,DVRL2I,DVRL3I,DVRL4I,DVRL5ID,ITOTPUP,
     $                     ITOTPUPI)

*           Calculation of auxiliary variables for CROP variables
            CALL CROP(ITASK,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,
     $                IPATL4I,ITIL5S,ITIL5D,DELT,IPATCUM,IPATDAT,
     $                IBATOUT,IBATOUTI,IPATLOW,DVRL1,DVRL2,DVRL3,
     $                DVRL4,DVRL5,DVRL1I,DVRL2I,DVRL3I,DVRL4I,
     $                IPLANTL1,IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,
     $                INPLANT,
     $                    LAI,ILPP,LAIBED,IHARV,FCCUM,LPDL1,LPDL2,
     $                    LPDL3,LPDL4,LPDL5)

*******************************************************
*           output
*******************************************************
            ITASK = 4

*           Output of SEDEV variables
            CALL SEDEV(ITASK,TIME,DELT,IHARV,IPATDAT,
     $                 RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT,
     $                 RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,RIRL4RS,
     $                 RIRL5SS,RIRL5RS,
     $                     IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                     IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,
     $                     IPATL3I,IPATL4I,ITIL5S,ITIL5D,IDRL1I,
     $                     IDRL2I,IDRL3I,IDRL4I,IDRL5ID,IMLTMAL,IMLTFEM,
     $                     IMLTMALI,IMLTFEMI,IDRMAL,IDRFEM,IDRMALI,
     $                     IDRFEMI,IRL1HT,IRL2HT,IRL3SHT,IRL3RHT,
     $                     IRL4SHT,IRL4RHT,IRL5SHTS,IRL5RHTS,IRL5SHTD,
     $                     IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                     IRL4RS,IRL5SSS,IRL5RSS,IRL5SSD,IRL5RSD,
     $                     ITOTMAL,ITOTFEM,ITOTMALI,ITOTFEMI,IBATIN,
     $                     IBATINI,IBATOUT,IBATOUTI,IHATCHI,IPATLOW,
     $                     IPATCUM,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                     DVRL1I,DVRL2I,DVRL3I,DVRL4I,DVRL5ID,ITOTPUP,
     $                     ITOTPUPI)

*           Output of SPADIS variables
            CALL SPADIS(ITASK,IPATLOW,IPATCUM,IRL1HT,IRL2HT,
     $                  IRL3SHT,IRL3RHT,IRL4SHT,IRL4RHT,IRL5SHTD,
     $                  IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                  IRL4RS,IRL5SSD,IRL5RSD,IDRL1I,IDRL2I,IDRL3I,
     $                  IDRL4I,IDRL5ID,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $                  ITIL5D,IBATOUT,IBATOUTI,IHATCHI,ILPP,LAIBED,
     $                  IHARV,
     $                      FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $                      VERTDIST,IPATDAT,IPLUP,IPCLVS,ICAD,IPLANTL1,
     $                      IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,INPLANT,
     $                      PHDLARTOT)

*           Output of VIRUS variables
            CALL VIRUS(ITASK,DELT,IHARV,IPATLOW,IPATCUM,IPATDAT,
     $                 LAIBED,IBATOUT,IBATOUTI,DELTLAI,TIME,
     $                     PHD,PHDLARTOT)

*           Output of CROP variables
            CALL CROP(ITASK,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,
     $                IPATL4I,ITIL5S,ITIL5D,DELT,IPATCUM,IPATDAT,
     $                IBATOUT,IBATOUTI,IPATLOW,DVRL1,DVRL2,DVRL3,
     $                DVRL4,DVRL5,DVRL1I,DVRL2I,DVRL3I,DVRL4I,
     $                IPLANTL1,IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,
     $                INPLANT,
     $                    LAI,ILPP,LAIBED,IHARV,FCCUM,LPDL1,LPDL2,
     $                    LPDL3,LPDL4,LPDL5)

*           Output of RIRHT variables
            CALL RIRHT(ITASK,FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $                 DELT,IPATCUM,VERTDIST,IPATLOW,LPDL1,LPDL2,
     $                 LPDL3,LPDL4,LPDL5,
     $                     RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT)

*           Output of RIRS variables
            CALL RIRSPR(ITASK,LAI,PHD,IPATCUM,VERTDIST,IPATLOW,
     $                  DELTLAI,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                      RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,
     $                      RIRL4SS,RIRL4RS,RIRL5SS,RIRL5RS)

********************************************************
*           State variables
********************************************************
            ITASK = 5

*           Time integral
            TIME = TIME + DELT

*           Crop development: LAI, leaves per plant and foliage consumption
            CALL CROP(ITASK,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,
     $                IPATL4I,ITIL5S,ITIL5D,DELT,IPATCUM,IPATDAT,
     $                IBATOUT,IBATOUTI,IPATLOW,DVRL1,DVRL2,DVRL3,
     $                DVRL4,DVRL5,DVRL1I,DVRL2I,DVRL3I,DVRL4I,
     $                IPLANTL1,IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,
     $                INPLANT,
     $                    LAI,ILPP,LAIBED,IHARV,FCCUM,LPDL1,LPDL2,
     $                    LPDL3,LPDL4,LPDL5)

*           Quantity and distribution of sprayed polyhedra in crop
            CALL VIRUS(ITASK,DELT,IHARV,IPATLOW,IPATCUM,IPATDAT,
     $                 LAIBED,IBATOUT,IBATOUTI,DELTLAI,TIME,
     $                     PHD,PHDLARTOT)

*           Spodoptera exigua development and population dynamics
            CALL SEDEV(ITASK,TIME,DELT,IHARV,IPATDAT,
     $                 RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT,
     $                 RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,RIRL4RS,
     $                 RIRL5SS,RIRL5RS,
     $                     IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                     IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,
     $                     IPATL3I,IPATL4I,ITIL5S,ITIL5D,IDRL1I,
     $                     IDRL2I,IDRL3I,IDRL4I,IDRL5ID,IMLTMAL,IMLTFEM,
     $                     IMLTMALI,IMLTFEMI,IDRMAL,IDRFEM,IDRMALI,
     $                     IDRFEMI,IRL1HT,IRL2HT,IRL3SHT,IRL3RHT,
     $                     IRL4SHT,IRL4RHT,IRL5SHTS,IRL5RHTS,IRL5SHTD,
     $                     IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                     IRL4RS,IRL5SSS,IRL5RSS,IRL5SSD,IRL5RSD,
     $                     ITOTMAL,ITOTFEM,ITOTMALI,ITOTFEMI,IBATIN,
     $                     IBATINI,IBATOUT,IBATOUTI,IHATCHI,IPATLOW,
     $                     IPATCUM,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                     DVRL1I,DVRL2I,DVRL3I,DVRL4I,DVRL5ID,ITOTPUP,
     $                     ITOTPUPI)

*           Simulation is terminated when number of infected larvae
*           or cadavers exceeds a critical value
            IF ((IPLUP.GE.950).OR.(IPCLVS.GE.950)) THEN
               TIME = FINTIM+1.
               WRITE(*,*)'number of infected larvae/cadavers too high'
               GOTO 10
            ENDIF

********************************************************
*           Shift
********************************************************
            ITASK = 6

*           Simulation is terminated when number of patches exceeds a
*           critical value
            IF (IPATCUM.GE.180) THEN
               TIME = FINTIM+1.
               WRITE(*,*)'number of patches too high'
               GOTO 10
            ENDIF

*           Shift for Spodoptera exigua development, egg batch size
*           in new patches, removal of larvae in harvested beds
*           and population totals
            CALL SEDEV(ITASK,TIME,DELT,IHARV,IPATDAT,
     $                 RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT,
     $                 RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,RIRL4RS,
     $                 RIRL5SS,RIRL5RS,
     $                     IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                     IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,
     $                     IPATL3I,IPATL4I,ITIL5S,ITIL5D,IDRL1I,
     $                     IDRL2I,IDRL3I,IDRL4I,IDRL5ID,IMLTMAL,IMLTFEM,
     $                     IMLTMALI,IMLTFEMI,IDRMAL,IDRFEM,IDRMALI,
     $                     IDRFEMI,IRL1HT,IRL2HT,IRL3SHT,IRL3RHT,
     $                     IRL4SHT,IRL4RHT,IRL5SHTS,IRL5RHTS,IRL5SHTD,
     $                     IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                     IRL4RS,IRL5SSS,IRL5RSS,IRL5SSD,IRL5RSD,
     $                     ITOTMAL,ITOTFEM,ITOTMALI,ITOTFEMI,IBATIN,
     $                     IBATINI,IBATOUT,IBATOUTI,IHATCHI,IPATLOW,
     $                     IPATCUM,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                     DVRL1I,DVRL2I,DVRL3I,DVRL4I,DVRL5ID,ITOTPUP,
     $                     ITOTPUPI)

*           Spatial distribution of new patches and removal of larvae
*           in harvested beds
            CALL SPADIS(ITASK,IPATLOW,IPATCUM,IRL1HT,IRL2HT,
     $                  IRL3SHT,IRL3RHT,IRL4SHT,IRL4RHT,IRL5SHTD,
     $                  IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                  IRL4RS,IRL5SSD,IRL5RSD,IDRL1I,IDRL2I,IDRL3I,
     $                  IDRL4I,IDRL5ID,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $                  ITIL5D,IBATOUT,IBATOUTI,IHATCHI,ILPP,LAIBED,
     $                  IHARV,
     $                      FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $                      VERTDIST,IPATDAT,IPLUP,IPCLVS,ICAD,IPLANTL1,
     $                      IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,INPLANT,
     $                      PHDLARTOT)

*           LAI and leaves per plant of new patches and newly planted beds
            CALL CROP(ITASK,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,
     $                IPATL4I,ITIL5S,ITIL5D,DELT,IPATCUM,IPATDAT,
     $                IBATOUT,IBATOUTI,IPATLOW,DVRL1,DVRL2,DVRL3,
     $                DVRL4,DVRL5,DVRL1I,DVRL2I,DVRL3I,DVRL4I,
     $                IPLANTL1,IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,
     $                INPLANT,
     $                    LAI,ILPP,LAIBED,IHARV,FCCUM,LPDL1,LPDL2,
     $                    LPDL3,LPDL4,LPDL5)

*           Quantity and distribution of sprayed polyhedra in new patches
*           and removal of sprayed polyhedra in harvested beds
            CALL VIRUS(ITASK,DELT,IHARV,IPATLOW,IPATCUM,IPATDAT,
     $                 LAIBED,IBATOUT,IBATOUTI,DELTLAI,TIME,
     $                     PHD,PHDLARTOT)

            GOTO 10
20        CONTINUE
          CLOSE (20,STATUS = 'DELETE')
30      CONTINUE
        END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE VIRUS                                                     *
*                                                                      *
* Purpose: Simulation of quantity and distribution of sprayed          *
*          virus over leaf layers.                                     *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                   class *
* ----    ---- ---------------                                  ----- *
* ITASK   I4   Task that subroutine should perform (-)            I   *
* DELT    R4   Time step (d)                                      I   *
* IHARV   I4   Flag for harvest (-)                               I   *
* IPATLOW I4   Pointer for lowest patch number containing larvae (-) I *
* IPATCUM I4   Cumulative number of patches in greenhouse (#)     I   *
* IPATDAT I4   Array containing co-ordinates of patches in            *
*              greenhouse (-)                                     I   *
* LAIBED  R4   Leaf area index of crop in bed (m2.m-2)            I   *
* IBATOUT I4   Number of batches in oviduct flowing out IBATCH        *
*              boxcar train (#)                                   I   *
* IBATOUTI                                                            *
*         I4   Number of batches in infected oviduct flowing out      *
*              IBATCHI boxcar train (#)                           I   *
* DELTLAI R4   LAI stratum (m2.m-2)                               I   *
* TIME    R4   Time (d)                                           I   *
* PHD     R4   Number of polyhedra per LAI stratum per patch (#)  O   *
* PHDLARTOT                                                           *
*         R4   Total number of polyhedra produced in larvae (#)   O   *
*                                                                     *
* Fatal error checks: none                                            *
* warnings:           none                                            *
* Subroutines called:                                                 *
* from TTUTIL:        RDINIT,RDSINT,RDFREA,RDSREA,OUTDAT              *
* File usage:         INCON.DAT, VIRUS.DAT, CROP.DAT                  *
*---------------------------------------------------------------------*
      SUBROUTINE VIRUS(ITASK,DELT,IHARV,IPATLOW,IPATCUM,IPATDAT,
     $                 LAIBED,IBATOUT,IBATOUTI,DELTLAI,TIME,
     $                     PHD,PHDLARTOT)

      PARAMETER (IPMAX = 200)
      INTEGER   IBEDMAX,IP,IHARV,IPATDAT,INSPRAY,IB,IL,IBATOUT,
     $          IBATOUTI
      REAL      RINACTR,PHDIRBED,PHD,DELT,PHDBED,TSPRAY,PHDSPR,
     $          TIME,DELTLAI,LAIMAX,K,LAIBED,FUP,FDOWN,PHDTOTBED,
     $          RESPHD,PHDIRLAR
      DIMENSION PHD(IPMAX,60),PHDBED(IPMAX,60),PHDIRBED(IPMAX,60),
     $          IPATDAT(3,IPMAX),TSPRAY(20),PHDSPR(20),LAIBED(IPMAX),
     $          PHDTOTBED(IPMAX)
      SAVE

*     Initialization of VIRUS variables
```

```
      IF (ITASK.EQ.1) THEN
*         Read initial conditions
          CALL RDINIT (30,40,'INCON.DAT')
          CALL RDSINT ('IBEDMAX',IBEDMAX)
          CALL RDSINT ('INSPRAY',INSPRAY)
          IF (INSPRAY.GT.0) THEN
              CALL RDFREA ('TSPRAY',TSPRAY(1),20,INSPRAY)
              CALL RDFREA ('PHDSPR',PHDSPR(1),20,INSPRAY)
          ENDIF
*         Read parameter values
          CALL RDINIT (30,40,'VIRUS.DAT')
          CALL RDSREA ('RINACTR',RINACTR)
          CALL RDSREA ('RESPHD',RESPHD)
          CALL RDINIT (30,40,'CROP.DAT')
          CALL RDSREA ('LAIMAX',LAIMAX)
          CALL RDSREA ('K',K)
*         Definition of LAI stratum
          DELTLAI = LAIMAX/60

*         Initial number of polyhedra per LAI stratum for each bed
          DO 20 IB = 1,IBEDMAX
              DO 10 IL = 1,60
                 PHDBED(IB,IL) = 0.
10           CONTINUE
20        CONTINUE

*         Initial number of polyhedra per LAI stratum for each patch
          DO 40 IP = 1,IPMAX
              DO 30 IL = 1,60
                  PHD(IP,IL) = 0.
30            CONTINUE
40        CONTINUE
      ENDIF

*     Calculation of VIRUS auxiliary variables
      IF (ITASK.EQ.2) THEN
*         Calculation of new polyhedra quantity per LAI stratum in each bed
*         after virus application
          DO 70 IN = 1,INSPRAY
            IF (ABS(TSPRAY(IN)-TIME).LE.(0.5*DELT)) THEN
                TSPRAY(IN) = 0.
*               Loop over beds
                DO 60 IB = 1,IBEDMAX
*                 Loop over LAI strata
                  DO 50 IL = 60,1,-1
*                   Calculation of fraction of polyhedra quantity at the top
*                   and bottom of LAI stratum
                    IF (LAIBED(IB).GT.((IL-1)*DELTLAI)) THEN
                        IF ((LAIBED(IB)-IL*DELTLAI).LT.0) THEN
                            FUP = 1.
                        ELSE
                            FUP = EXP(-1*(LAIBED(IB)-IL*DELTLAI)*K)
                        ENDIF
                        FDOWN = EXP(-1*(LAIBED(IB)-(IL-1)*DELTLAI)*K)
*                       New polyhedra quantity per LAI stratum after spray
                        PHDBED(IB,IL) = PHDBED(IB,IL) +
     $                                   (FUP-FDOWN)*PHDSPR(IN)
                    ELSE
                        PHDBED(IB,IL) = 0.
                    ENDIF
50                CONTINUE
60              CONTINUE
            ENDIF
70        CONTINUE

*         Polyhedra quantity per LAI stratum for each patch
          DO 90 IP = IPATLOW,IPATCUM
            DO 80 IL = 1,60
              PHD(IP,IL) = PHDBED(IPATDAT(1,IP),IL)
80          CONTINUE
90        CONTINUE
      ENDIF

*     Calculation of VIRUS rate variables
      IF (ITASK.EQ.3) THEN
*         Calculation of polyhedra inactivation rate per LAI stratum using
*         the relative inactivation rate of polyhedra and residual density
*         of infectious polyhedra
          DO 110 IB = 1,IBEDMAX
              DO 100 IL = 1,60
                  PHDIRBED(IB,IL)=(PHDBED(IB,IL)-RESPHD*DELTLAI)*RINACTR
                  IF (PHDIRBED(IB,IL).LT.0) PHDIRBED(IB,IL) = 0.
100           CONTINUE
110       CONTINUE

*         Calculation of inactivation rate of polyhedra produced in larvae
          PHDIRLAR = PHDLARTOT*RINACTR
      ENDIF

*     Output of VIRUS variables
      IF (ITASK.EQ.4) THEN
          CALL OUTDAT(2,20,'TIME',TIME)
          CALL OUTDAT(2,20,'PHDBED1',PHDTOTBED(1))
      ENDIF

*     Calculation of VIRUS state variables
      IF (ITASK.EQ.5) THEN
*         Calculation of polyhedra quantity per LAI stratum in each bed
          DO 130 IB = 1,IBEDMAX
              PHDTOTBED(IB) = 0.
          DO 120 IL = 1,60
             PHDBED(IB,IL)  = PHDBED(IB,IL)-PHDIRBED(IB,IL)*DELT
             PHDTOTBED(IB)  = PHDTOTBED(IB)+PHDBED(IB,IL)
120       CONTINUE
130     CONTINUE

*       Polyhedra quantity per LAI stratum for each patch
        DO 150 IP = IPATLOW,IPATCUM
         DO 140 IL = 1,60
          PHD(IP,IL) = PHDBED(IPATDAT(1,IP),IL)
140      CONTINUE
150     CONTINUE

*       Calculation of the number of infectious polyhedra produced in larvae
        PHDLARTOT = PHDLARTOT - PHDIRLAR*DELT
      ENDIF

*     Calculation of shift for VIRUS variables
      IF (ITASK.EQ.6) THEN
*       After a harvest the new plants are polyhedra free
        DO 170 IB = 1,IBEDMAX
          IF (IB.EQ.IHARV) THEN
            DO 160 IL = 1,60
              PHDBED(IB,IL) = 0.
160          CONTINUE
          ENDIF
170     CONTINUE

*       Polyhedra quantity per LAI stratum of new patches
        IF (IBATOUT.GT.0.OR.IBATOUTI.GT.0) THEN
          DO 190 IB = 1,(IBATOUT+IBATOUTI)
           DO 180 IL = 1,60
            PHD(IPATCUM-IB+1,IL) =
     $       PHDBED(IPATDAT(1,IPATCUM-IB+1),IL)
180        CONTINUE
190       CONTINUE
        ENDIF
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE RIRSPR                                                    *
*                                                                      *
* Purpose: Simulation of relative infection rate by ingestion          *
*          of sprayed polyhedra.                                       *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* ITASK    I4  Task that subroutine should perform (-)             I   *
* LAI      R4  Leaf area index of crop in patch (m2.m-2)           I   *
* PHD      R4  Number of polyhedra per LAI stratum per patch (#)   I   *
* IPATCUM I4  Cumulative number of patches in greenhouse (#)       I   *
* VERTDIST                                                             *
*          R4  Fraction larvae per canopy layer (-)                I   *
* IPATLOW I4  Pointer for lowest patch number containing larvae (-) I  *
* DELTLAI R4  LAI stratum (m2.m-2)                                 I   *
* DVRL1    R4  Development rate of L1 larvae (d-1)                 I   *
* DVRL2    R4  Development rate of L2 larvae (d-1)                 I   *
* DVRL3    R4  Development rate of L3 larvae (d-1)                 I   *
* DVRL4    R4  Development rate of L4 larvae (d-1)                 I   *
* DVRL5    R4  Development rate of L5 larvae (d-1)                 I   *
* RIRL1S   R4  Relative infection rate by ingestion of sprayed         *
*              polyhedra of L1 larvae (d-1)                        O   *
* RIRL2S   R4  Relative infection rate by ingestion of sprayed         *
*              polyhedra of L2 larvae (d-1)                        O   *
* RIRL3SS R4  Relative infection rate by ingestion of sprayed          *
*              polyhedra of susceptible L3 larvae (d-1)            O   *
* RIRL3RS R4  Relative infection rate by ingestion of sprayed          *
*              polyhedra of resistant L3 larvae (d-1)              O   *
* RIRL4SS R4  Relative infection rate by ingestion of sprayed          *
*              polyhedra of susceptible L4 larvae (d-1)            O   *
* RIRL4RS R4  Relative infection rate by ingestion of sprayed          *
*              polyhedra of resistant L4 larvae (d-1)              O   *
* RIRL5SS R4  Relative infection rate by ingestion of sprayed          *
*              polyhedra of susceptible L5 larvae (d-1)            O   *
* RIRL5RS R4  Relative infection rate by ingestion of sprayed          *
*              polyhedra of resistant L5 larvae (d-1)              O   *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
*                                                                      *
* Fatal error checks: none                                             *
* warnings:           none                                             *
* Subroutines called:                                                  *
* from TTUTIL:        RDINIT,RDSREA,OUTDAT                             *
* File usage:         SEDEV.DAT, VIRUS.DAT                             *
*----------------------------------------------------------------------*
      SUBROUTINE RIRSPR(ITASK,LAI,PHD,IPATCUM,VERTDIST,IPATLOW,
     $                  DELTLAI,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,
     $                      RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,
     $                      RIRL4RS,RIRL5SS,RIRL5RS)
```

```
      PARAMETER (IPMAX = 200)
      INTEGER   IP,IL
      REAL      LAI,PHD,POLDEN,CONSL1,CONSL2,CONSL3,CONSL4,
     $          CONSL5,PL1,PL2,PL3S,PL3R,PL4S,PL4R,PL5S,PL5R,
     $          RIRL1S,RIRL2S,RIRL3SS,RIRL3RS,RIRL4SS,RIRL4RS,
     $          RIRL5SS,RIRL5RS,VERTDIST,DELTLAI,DVRL1,DVRL2,
     $          DVRL3,DVRL4,DVRL5,CFP
      DIMENSION LAI(IPMAX),POLDEN(3),RIRL1S(IPMAX),RIRL2S(IPMAX),
     $          RIRL3SS(IPMAX),RIRL3RS(IPMAX),RIRL4SS(IPMAX),
     $          RIRL4RS(IPMAX),RIRL5SS(IPMAX),RIRL5RS(IPMAX),
     $          VERTDIST(3,5),PHD(IPMAX,60)
      SAVE

*     Initialization of RIRSPR variables
      IF (ITASK.EQ.1) THEN
*         Read parameter values
          CALL RDINIT (30,40,'SEDEV.DAT')
*         Consumption rate per larval instar
          CALL RDSREA ('CONSL1',CONSL1)
          CALL RDSREA ('CONSL2',CONSL2)
          CALL RDSREA ('CONSL3',CONSL3)
          CALL RDSREA ('CONSL4',CONSL4)
          CALL RDSREA ('CONSL5',CONSL5)
          CALL RDINIT (30,40,'VIRUS.DAT')
*         Infection chance per ingested polyhedron per larval instar
          CALL RDSREA ('PL1',PL1)
          CALL RDSREA ('PL2',PL2)
          CALL RDSREA ('PL3S',PL3S)
          CALL RDSREA ('PL3R',PL3R)
          CALL RDSREA ('PL4S',PL4S)
          CALL RDSREA ('PL4R',PL4R)
          CALL RDSREA ('PL5S',PL5S)
          CALL RDSREA ('PL5R',PL5R)
*         Conversion factor of infection chance per ingested polyhedron
*         from artificial diet to chrysanthemum
          CALL RDSREA ('CFP',CFP)
      ENDIF

*     Calculation of RIRSPR rate variables
      IF (ITASK.EQ.3) THEN
*         Calculation of virus distribution of LAI strata over canopy layers
          DO 40 IP = IPATLOW,IPATCUM
*             Calculation of location of canopy layers in canopy
              ILAYER3 = INT(LAI(IP)/DELTLAI)+1
              ILAYER2 = INT(ILAYER3*2/3)
              ILAYER1 = INT(ILAYER3/3)

*             Sprayed polyhedra quantity in each canopy layer
              PHDL1 = 0
      CALL OUTDAT(2,20,'RIRL1S',RIRL1S(1))
      CALL OUTDAT(2,20,'RIRL2S',RIRL2S(1))
      CALL OUTDAT(2,20,'RIRL3SS',RIRL3SS(1))
      CALL OUTDAT(2,20,'RIRL3RS',RIRL3RS(1))
      CALL OUTDAT(2,20,'RIRL4SS',RIRL4SS(1))
      CALL OUTDAT(2,20,'RIRL4RS',RIRL4RS(1))
      CALL OUTDAT(2,20,'RIRL5SS',RIRL5SS(1))
      CALL OUTDAT(2,20,'RIRL5RS',RIRL5RS(1))
   ENDIF

   RETURN
   END
```

```
              PHDL2 = 0
              PHDL3 = 0

              DO 10 IL = 1,ILAYER1
                PHDL1 = PHDL1 + PHD(IP,IL)
10            CONTINUE
*             Sprayed polyhedra density in lowest canopy layer
              POLDEN(1) = PHDL1/(ILAYER1*DELTLAI)
              DO 20 IL = ILAYER1+1,ILAYER2
                PHDL2 = PHDL2 + PHD(IP,IL)
20            CONTINUE
*             Sprayed polyhedra density in middle canopy layer
              POLDEN(2) = PHDL2/((ILAYER2-ILAYER1)*DELTLAI)
              DO 30 IL = ILAYER2+1,ILAYER3
                PHDL3 = PHDL3 + PHD(IP,IL)
30            CONTINUE
*             Sprayed polyhedra density in upper canopy layer
              POLDEN(3) = PHDL3/(LAI(IP)-(ILAYER2*DELTLAI))

*             Relative infection rate by ingestion of sprayed polyhedra
*             per larval instar (1/day)
              RIRL1S(IP) = CONSL1*DVRL1*PL1*POLDEN(1)*VERTDIST(1,1)*CFP+
     $                     CONSL1*DVRL1*PL1*POLDEN(2)*VERTDIST(2,1)*CFP+
     $                     CONSL1*DVRL1*PL1*POLDEN(3)*VERTDIST(3,1)*CFP
              RIRL2S(IP) = CONSL2*DVRL2*PL2*POLDEN(1)*VERTDIST(1,2)*CFP+
     $                     CONSL2*DVRL2*PL2*POLDEN(2)*VERTDIST(2,2)*CFP+
     $                     CONSL2*DVRL2*PL2*POLDEN(3)*VERTDIST(3,2)*CFP
              RIRL3SS(IP)=CONSL3*DVRL3*PL3S*POLDEN(1)*VERTDIST(1,3)*CFP+
     $                    CONSL3*DVRL3*PL3S*POLDEN(2)*VERTDIST(2,3)*CFP+
     $                    CONSL3*DVRL3*PL3S*POLDEN(3)*VERTDIST(3,3)*CFP
              RIRL3RS(IP)=CONSL3*DVRL3*PL3R*POLDEN(1)*VERTDIST(1,3)*CFP+
     $                    CONSL3*DVRL3*PL3R*POLDEN(2)*VERTDIST(2,3)*CFP+
     $                    CONSL3*DVRL3*PL3R*POLDEN(3)*VERTDIST(3,3)*CFP
              RIRL4SS(IP)=CONSL4*DVRL4*PL4S*POLDEN(1)*VERTDIST(1,4)*CFP+
     $                    CONSL4*DVRL4*PL4S*POLDEN(2)*VERTDIST(2,4)*CFP+
     $                    CONSL4*DVRL4*PL4S*POLDEN(3)*VERTDIST(3,4)*CFP
              RIRL4RS(IP)=CONSL4*DVRL4*PL4R*POLDEN(1)*VERTDIST(1,4)*CFP+
     $                    CONSL4*DVRL4*PL4R*POLDEN(2)*VERTDIST(2,4)*CFP+
     $                    CONSL4*DVRL4*PL4R*POLDEN(3)*VERTDIST(3,4)*CFP
              RIRL5SS(IP)=CONSL5*DVRL5*PL5S*POLDEN(1)*VERTDIST(1,5)*CFP+
     $                    CONSL5*DVRL5*PL5S*POLDEN(2)*VERTDIST(2,5)*CFP+
     $                    CONSL5*DVRL5*PL5S*POLDEN(3)*VERTDIST(3,5)*CFP
              RIRL5RS(IP)=CONSL5*DVRL5*PL5R*POLDEN(1)*VERTDIST(1,5)*CFP+
     $                    CONSL5*DVRL5*PL5R*POLDEN(2)*VERTDIST(2,5)*CFP+
     $                    CONSL5*DVRL5*PL5R*POLDEN(3)*VERTDIST(3,5)*CFP
40        CONTINUE
      ENDIF

*     Output of RIRSPR variables
      IF (ITASK.EQ.4) THEN
```

```
*----------------------------------------------------------------------*
* SUBROUTINE CROP                                                      *
*                                                                      *
* Purpose: Simulation of crop development: LAI, leaves per plant       *
*          and crop injury.                                            *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                     class *
* ----   ---- ---------------                                    ----- *
* ITASK   I4  Task that subroutine should perform (-)               I  *
* IPATL1  I4  Number of L1 larvae in boxcar train per patch (#)     I  *
* IPATL2  I4  Number of L2 larvae in boxcar train per patch (#)     I  *
* IPATL3S I4  Number of susceptible L3 larvae in boxcar train per      *
*             patch (#)                                             I  *
* IPATL3R I4  Number of resistant L3 larvae in boxcar train per        *
*             patch (#)                                             I  *
* IPATL4S I4  Number of susceptible L4 larvae in boxcar train per      *
*             patch (#)                                             I  *
* IPATL4R I4  Number of resistant L4 larvae in boxcar train per        *
*             patch (#)                                             I  *
* IPATL5S I4  Number of susceptible L5 larvae in boxcar train per      *
*             patch (#)                                             I  *
* IPATL5R I4  Number of resistant L5 larvae in boxcar train per        *
*             patch (#)                                             I  *
* IPATL1I I4  Number of infected L1 larvae in boxcar train per         *
*             patch (#)                                             I  *
* IPATL2I I4  Number of infected L2 larvae in boxcar train per         *
*             patch (#)                                             I  *
* IPATL3I I4  Number of infected L3 larvae in boxcar train per         *
*             patch (#)                                             I  *
* IPATL4I I4  Number of infected L4 larvae in boxcar train per         *
*             patch (#)                                             I  *
* ITIL5S  I4  Number of sublethally infected L5 larvae per boxcar      *
*             (#)                                                   I  *
* ITIL5D  I4  Number of lethally infected L5 larvae per boxcar (#)  I  *
* DELT    R4  Time step (d)                                         I  *
* IPATCUM I4  Cumulative number of patches in greenhouse (#)        I  *
* IPATDAT I4  Array containing co-ordinates of patches in              *
*             greenhouse (-)                                        I  *
* IBATOUT I4  Number of batches in oviduct flowing out IBATCH          *
*             boxcar train (#)                                      I  *
* IBATOUTI                                                             *
*         I4  Number of batches in infected oviduct flowing out        *
*             IBATCHI boxcar train (#)                              I  *
* IPATLOW I4  Pointer for lowest patch number containing larvae (-) I  *
* DVRL1   R4  Development rate of L1 larvae (d-1)                   I  *
* DVRL2   R4  Development rate of L2 larvae (d-1)                   I  *
* DVRL3   R4  Development rate of L3 larvae (d-1)                   I  *
* DVRL4   R4  Development rate of L4 larvae (d-1)                   I  *
* DVRL5   R4  Development rate of L5 larvae (d-1)                   I  *
* DVRL1I   R4  Development rate of infected L1 larvae (d-1)           I  *
* DVRL2I   R4  Development rate of infected L2 larvae (d-1)           I  *
* DVRL3I   R4  Development rate of infected L3 larvae (d-1)           I  *
* DVRL4I   R4  Development rate of infected L4 larvae (d-1)           I  *
* IPLANTL1                                                               *
*          I4  Number of plants in foraging domain of L1 larvae (#)  I  *
* IPLANTL2                                                               *
*          I4  Number of plants in foraging domain of L2 larvae (#)  I  *
* IPLANTL3                                                               *
*          I4  Number of plants in foraging domain of L3 larvae (#)  I  *
* IPLANTL4                                                               *
*          I4  Number of plants in foraging domain of L4 larvae (#)  I  *
* IPLANTL5                                                               *
*          I4  Number of plants in foraging domain of L5 larvae (#)  I  *
* INPLANT  I4  Number of plants per bed (#)                           I  *
* LAI      R4  Leaf area index of crop in patch (m2.m-2)              O  *
* ILPP     I4  Number of leaves per plant in patch (#)                O  *
* LAIBED   R4  Leaf area index of crop in bed (m2.m-2)                O  *
* IHARV    I4  Flag for harvest (-)                                   O  *
* FCCUM    R4  Cumulative foliage consumption in greenhouse (m2)      O  *
* LPDL1    R4  Mean number of leaf visits of L1 larvae per day (d-1) O  *
* LPDL2    R4  Mean number of leaf visits of L2 larvae per day (d-1) O  *
* LPDL3    R4  Mean number of leaf visits of L3 larvae per day (d-1) O  *
* LPDL4    R4  Mean number of leaf visits of L4 larvae per day (d-1) O  *
* LPDL5    R4  Mean number of leaf visits of L5 larvae per day (d-1) O  *
*                                                                        *
* Fatal error checks: IBEDMAX > 20                                       *
*                     LAIHARV > LAIMAX                                   *
* warnings:           none                                               *
* Subroutines called:                                                    *
* from TTUTIL:        RDINIT,RDFREA,RDSINT,RDSREA,OUTDAT                 *
* File usage:         INCON.DAT, CROP.DAT, SEDEV.DAT                     *
*------------------------------------------------------------------------*
      SUBROUTINE CROP(ITASK,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $                IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,
     $                IPATL4I,ITIL5S,ITIL5D,DELT,IPATCUM,IPATDAT,
     $                IBATOUT,IBATOUTI,IPATLOW,DVRL1,DVRL2,DVRL3,DVRL4,
     $                DVRL5,DVRL1I,DVRL2I,DVRL3I,DVRL4I,IPLANTL1,
     $                IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,INPLANT,
     $                    LAI,ILPP,LAIBED,IHARV,FCCUM,LPDL1,LPDL2,
     $                    LPDL3,LPDL4,LPDL5)

      PARAMETER (IPMAX = 200)
      INTEGER   IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,IPATL4R,IPATL5S,
     $          IPATL5R,IPATL1I,IPATL2I,IPATL3I,IPATL4I,ITIL5S,ITIL5D,
     $          IBEDMAX,IPATDAT,ILPPBED,ILPP,IBATOUT,IBATOUTI,IHARV,IP,
     $          IB,IPLANTL1,IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5
      REAL      LAI,CONSL1,CONSL2,CONSL3,CONSL4,CONSL5,LAIBED,LAIHARV,
     $          FCR,FCRL1,FCRL2,FCRL3,FCRL4,FCRL5,LAI0,RGRLAI,
     $          DAPLAI50,LAIMAX,RGRLPP,DAPLPP50,LPPMAX,DAP,DELT,
     $           LPP0,FCCUM,DVRL1,DVRL2,DVRL3,DVRL4,DVRL5,DVRL1I,
     $           DVRL2I,DVRL3I,DVRL4I,PDERL1,PDERL2,PDERL3,PDERL4,
     $           PDERL5,LPDL1,LPDL2,LPDL3,LPDL4,LPDL5,PDEFORL1,
     $           PDEFORL2,PDEFORL3,PDEFORL4,PDEFORL5,PDERFORL1,
     $           PDERFORL2,PDERFORL3,PDERFORL4,PDERFORL5,DPFORL1,
     $           DPFORL2,DPFORL3,DPFORL4,DPFORL5,DPPATCH,DPCUM,
     $           DPCUMO,DVRL1T27,DVRL2T27,DVRL3T27,DVRL4T27,DVRL5T27,
     $           LPDL1T27,LPDL2T27,LPDL3T27,LPDL4T27,LPDL5T27,LINT,
     $           CFDVR,DVRL1T,DVRL2T,DVRL3T,DVRL4T,DVRL5T,DPBEDO,
     $           DPBED,FCPATCH,DPTOT,DPTOTO,FCTOT
      DIMENSION LAI(IPMAX),IPATL1(IPMAX),IPATL2(IPMAX),IPATL3S(IPMAX),
     $           IPATL3R(IPMAX),IPATL4S(IPMAX),IPATL4R(IPMAX),
     $           IPATL5S(IPMAX),IPATL5R(IPMAX),IPATL1I(IPMAX),
     $           IPATL2I(IPMAX),IPATL3I(IPMAX),IPATL4I(IPMAX),
     $           ITIL5S(IPMAX),ITIL5D(IPMAX),LAIBED(IPMAX),
     $           IPATDAT(3,IPMAX),ILPPBED(IPMAX),ILPP(IPMAX),
     $           FCRL1(IPMAX),FCRL2(IPMAX),FCRL3(IPMAX),FCRL4(IPMAX),
     $           FCRL5(IPMAX),FCR(IPMAX),DAP(50),DPPATCH(IPMAX),
     $           PDERL1(IPMAX),PDERL2(IPMAX),PDERL3(IPMAX),
     $           PDERL4(IPMAX),PDERL5(IPMAX),PDERFORL1(IPMAX),
     $           PDERFORL2(IPMAX),PDERFORL3(IPMAX),PDERFORL4(IPMAX),
     $           PDERFORL5(IPMAX),PDEFORL1(IPMAX),PDEFORL2(IPMAX),
     $           PDEFORL3(IPMAX),PDEFORL4(IPMAX),PDEFORL5(IPMAX),
     $           DPFORL1(IPMAX),DPFORL2(IPMAX),DPFORL3(IPMAX),
     $           DPFORL4(IPMAX),DPFORL5(IPMAX),DVRL1T(12),DVRL2T(12),
     $           DVRL3T(12),DVRL4T(12),DVRL5T(12),DPBEDO(IPMAX),
     $           DPBED(IPMAX),FCPATCH(IPMAX)
      SAVE

*     Initialization of CROP variables
      IF (ITASK.EQ.1) THEN
*        Read initial conditions
         CALL RDINIT (30,40,'INCON.DAT')
         CALL RDFREA ('DAP',DAP(1),50,50)
         CALL RDSINT ('IBEDMAX',IBEDMAX)
*        Read parameter values
         CALL RDINIT (30,40,'CROP.DAT')
         CALL RDSREA ('RGRLAI',RGRLAI)
         CALL RDSREA ('DAPLAI50',DAPLAI50)
         CALL RDSREA ('LAI0',LAI0)
         CALL RDSREA ('LAIMAX',LAIMAX)
         CALL RDSREA ('RGRLPP',RGRLPP)
         CALL RDSREA ('DAPLPP50',DAPLPP50)
         CALL RDSREA ('LPP0',LPP0)
         CALL RDSREA ('LPPMAX',LPPMAX)
         CALL RDSREA ('LAIHARV',LAIHARV)
         CALL RDINIT (30,40,'SEDEV.DAT')
         CALL RDSREA ('CONSL1',CONSL1)
         CALL RDSREA ('CONSL2',CONSL2)
         CALL RDSREA ('CONSL3',CONSL3)
      CALL RDSREA ('CONSL4',CONSL4)
      CALL RDSREA ('CONSL5',CONSL5)
      CALL RDFREA ('DVRL1T',DVRL1T(1),12,12)
      CALL RDFREA ('DVRL2T',DVRL2T(1),12,12)
      CALL RDFREA ('DVRL3T',DVRL3T(1),12,12)
      CALL RDFREA ('DVRL4T',DVRL4T(1),12,12)
      CALL RDFREA ('DVRL5T',DVRL5T(1),12,12)
      CALL RDSREA ('CFDVR' ,CFDVR)
      CALL RDINIT (30,40,'SPADIS.DAT')
      CALL RDSREA ('LPDL1',LPDL1T27)
      CALL RDSREA ('LPDL2',LPDL2T27)
      CALL RDSREA ('LPDL3',LPDL3T27)
      CALL RDSREA ('LPDL4',LPDL4T27)
      CALL RDSREA ('LPDL5',LPDL5T27)

      IF (IBEDMAX.GT.50) THEN
         STOP'IBEDMAX value reached maximum value of 50!'
      ENDIF

      IF (LAIHARV.GE.LAIMAX) THEN
         STOP'LAIHARV value higher as maximal LAI value!'
      ENDIF

*     Initial LAI and leaves per plant for each bed
      DO 10 IB = 1,IBEDMAX
         LAIBED(IB) = LAI0 + ((LAIMAX - LAI0)/
     $                (1. + EXP(-RGRLAI*(DAP(IB) - DAPLAI50))))
         ILPPBED(IB) = NINT(LPP0 + ((LPPMAX - LPP0)/
     $                 (1. + EXP(-RGRLPP*(DAP(IB) - DAPLPP50)))))
10     CONTINUE

*     Initial LAI and leaves per plant for each patch
      DO 20 IP = 1,IPATCUM
         LAI(IP) = LAIBED(IPATDAT(1,IP))
         ILPP(IP) = ILPPBED(IPATDAT(1,IP))
20    CONTINUE

*     Initialization of foliage consumption and plant damage
*     variables for each patch
      DO 30 IP = 1,IPMAX
         FCRL1(IP) = 0.
         FCRL2(IP) = 0.
         FCRL3(IP) = 0.
         FCRL4(IP) = 0.
         FCRL5(IP) = 0.
         FCPATCH(IP) = 0.
         PDERL1(IP) = 0.
         PDERL2(IP) = 0.
         PDERL3(IP) = 0.
         PDERL4(IP) = 0.
```

```
              PDERL5(IP) = 0.
              PDERFORL1(IP) = 0.
              PDERFORL2(IP) = 0.
              PDERFORL3(IP) = 0.
              PDERFORL4(IP) = 0.
              PDERFORL5(IP) = 0.
              PDEFORL1(IP) = 0.
              PDEFORL2(IP) = 0.
              PDEFORL3(IP) = 0.
              PDEFORL4(IP) = 0.
              PDEFORL5(IP) = 0.
              DPFORL1(IP) = 0.
              DPFORL2(IP) = 0.
              DPFORL3(IP) = 0.
              DPFORL4(IP) = 0.
              DPFORL5(IP) = 0.
              DPPATCH(IP) = 0.
30        CONTINUE
          FCCUM = 0.
      ENDIF

*     Calculation of CROP auxiliary variables
      IF (ITASK.EQ.2) THEN
*         Calculation of leaf visit rate of larvae
          DVRL1T27  = LINT(DVRL1T, 12, 27.) * CFDVR
          DVRL2T27  = LINT(DVRL2T, 12, 27.) * CFDVR
          DVRL3T27  = LINT(DVRL3T, 12, 27.) * CFDVR
          DVRL4T27  = LINT(DVRL4T, 12, 27.) * CFDVR
          DVRL5T27  = LINT(DVRL5T, 12, 27.) * CFDVR
          LPDL1 = LPDL1T27 * DVRL1/DVRL1T27
          LPDL2 = LPDL2T27 * DVRL2/DVRL2T27
          LPDL3 = LPDL3T27 * DVRL3/DVRL3T27
          LPDL4 = LPDL4T27 * DVRL4/DVRL4T27
          LPDL5 = LPDL5T27 * DVRL5/DVRL5T27

          IHARV = 0
*         Trigger for harvest
          DO 40 IB = 1,IBEDMAX
              IF (LAIBED(IB).GE.LAIHARV.AND.IHARV.EQ.0) THEN
                 IHARV = IB
                 WRITE (*,*)'Harvest of bed',IB
              ENDIF
40        CONTINUE
      ENDIF

*     Calculation of CROP rate variables
      IF (ITASK.EQ.3) THEN

          DO 50 IP = IPATLOW,IPATCUM
*             Calculation of foliage consumption rate per patch
```

```
              IF (((DVRL1I/DVRL1)).LE.1.) THEN
                  FCRL1(IP)=IPATL1(IP)*CONSL1*DVRL1+
     $                      IPATL1I(IP)*(CONSL1*(DVRL1I/DVRL1)+
     $                      CONSL2*(1.-(DVRL1I/DVRL1)))*DVRL1
              ELSE
                  FCRL1(IP)=(IPATL1(IP)+IPATL1I(IP))*CONSL1*DVRL1
              ENDIF
              IF (((DVRL2I/DVRL2)).LE.1.) THEN
                  FCRL2(IP)=IPATL2(IP)*CONSL2*DVRL2+
     $                      IPATL2I(IP)*(CONSL2*(DVRL2I/DVRL2)+
     $                      CONSL3*(1.-(DVRL2I/DVRL2)))*DVRL2
              ELSE
                  FCRL2(IP)=(IPATL2(IP)+IPATL2I(IP))*CONSL2*DVRL2
              ENDIF
              IF (((DVRL3I/DVRL3)).LE.1.) THEN
                  FCRL3(IP)=(IPATL3S(IP)+IPATL3R(IP))*CONSL3*DVRL3+
     $                      IPATL3I(IP)*(CONSL3*(DVRL3I/DVRL3)+
     $                      CONSL4*(1.-(DVRL3I/DVRL3)))*DVRL3
              ELSE
                  FCRL3(IP)=(IPATL3S(IP)+IPATL3R(IP)+IPATL3I(IP))*
     $                      CONSL3*DVRL3
              ENDIF
              IF (((DVRL4I/DVRL4)).LE.1.) THEN
                  FCRL4(IP)=(IPATL4S(IP)+IPATL4R(IP))*CONSL4*DVRL4+
     $                      IPATL4I(IP)*(CONSL4*(DVRL4I/DVRL4)+
     $                      CONSL5*(1.-(DVRL4I/DVRL4)))*DVRL4
              ELSE
                  FCRL4(IP)=(IPATL4S(IP)+IPATL4R(IP)+IPATL4I(IP))*
     $                      CONSL4*DVRL4
              ENDIF
              FCRL5(IP)=(IPATL5S(IP)+IPATL5R(IP)+ITIL5S(IP)+ITIL5D(IP))*
     $                  CONSL5*DVRL5
              FCR(IP)=FCRL1(IP)+FCRL2(IP)+FCRL3(IP)+FCRL4(IP)+FCRL5(IP)

*             Calculation of plant damage event rate of
*             populations of L1 to L5 larvae per patch
              PDERL1(IP)=(IPATL1(IP)+IPATL1I(IP)*
     $                   (MIN(1.,(DVRL1I/DVRL1))))*LPDL1*0.14
              PDERL2(IP)=(IPATL2(IP)+IPATL2I(IP)*
     $                   (MIN(1.,(DVRL2I/DVRL2)))+IPATL1I(IP)*
     $                   (1.-MIN(1.,(DVRL1I/DVRL1))))*LPDL2*0.25
              PDERL3(IP)=(IPATL3S(IP)+IPATL3R(IP)+
     $                   IPATL3I(IP)*(MIN(1.,(DVRL3I/DVRL3)))+
     $                   IPATL2I(IP)*(1.-MIN(1.,(DVRL2I/DVRL2))))*
     $                   LPDL3*0.4
              PDERL4(IP)=(IPATL4S(IP)+IPATL4R(IP)+
     $                   IPATL4I(IP)*(MIN(1.,(DVRL4I/DVRL4)))+
     $                   IPATL3I(IP)*(1.-MIN(1.,(DVRL3I/DVRL3))))*
     $                   LPDL4*0.4
              PDERL5(IP)=(IPATL5S(IP)+IPATL5R(IP)+ITIL5D(IP)+
     $                    ITIL5S(IP)+IPATL4I(IP)*
     $                    (1.-MIN(1.,(DVRL4I/DVRL4))))*0.6*LPDL5

*              Calculation of plant damage event rate in foraging
*              domains of L1 to L5 larvae per patch
               PDERFORL1(IP)=(PDERL1(IP))+
     $                      (PDERL2(IP)*(REAL(IPLANTL1)/REAL(IPLANTL2)))+
     $                      (PDERL3(IP)*(REAL(IPLANTL1)/REAL(IPLANTL3)))+
     $                      (PDERL4(IP)*(REAL(IPLANTL1)/REAL(IPLANTL4)))+
     $                      (PDERL5(IP)*(REAL(IPLANTL1)/REAL(IPLANTL5)))
               PDERFORL2(IP)=(PDERL2(IP)*
     $                      (1.-REAL(IPLANTL1)/REAL(IPLANTL2)))+
     $                      (PDERL3(IP)*(REAL(IPLANTL2-IPLANTL1)/
     $                      REAL(IPLANTL3)))+
     $                      (PDERL4(IP)*(REAL(IPLANTL2-IPLANTL1)/
     $                      REAL(IPLANTL4)))+
     $                      (PDERL5(IP)*(REAL(IPLANTL2-IPLANTL1)/
     $                      REAL(IPLANTL5)))
               PDERFORL3(IP)=(PDERL3(IP)*
     $                      (1.-REAL(IPLANTL2)/REAL(IPLANTL3)))+
     $                      (PDERL4(IP)*(REAL(IPLANTL3-IPLANTL2)/
     $                      REAL(IPLANTL4)))+
     $                      (PDERL5(IP)*(REAL(IPLANTL3-IPLANTL2)/
     $                      REAL(IPLANTL5)))
               PDERFORL4(IP)=(PDERL4(IP)*
     $                      (1.-REAL(IPLANTL3)/REAL(IPLANTL4)))+
     $                      (PDERL5(IP)*(REAL(IPLANTL4-IPLANTL3)/
     $                      REAL(IPLANTL5)))
               PDERFORL5(IP)=(PDERL5(IP)*
     $                      (1.-REAL(IPLANTL4)/REAL(IPLANTL5)))
50        CONTINUE
      ENDIF

*     Output of CROP variables
      IF (ITASK.EQ.4) THEN
          CALL OUTDAT(2,20,'LAI1',LAI(1))
          CALL OUTDAT(2,20,'LAIBED1',LAIBED(1))
          CALL OUTDAT(2,20,'ILPP1',REAL(ILPP(1)))
          CALL OUTDAT(2,20,'ILPPBED1',REAL(ILPPBED(1)))
          CALL OUTDAT(2,20,'FCCUM',FCCUM)
          CALL OUTDAT(2,20,'DPCUMO',DPCUMO)
      ENDIF

*     Calculation of CROP state variables
      IF (ITASK.EQ.5) THEN
*         LAI and leaves per plant for each bed
          DO 60 IB = 1,IBEDMAX
              DAP(IB) = DAP(IB) + DELT
              LAIBED(IB) = LAI0 + ((LAIMAX - LAI0)/
     $                    (1. + EXP(-RGRLAI*(DAP(IB) - DAPLAI50))))
```

```
              ILPPBED(IB) = NINT(LPP0 + ((LPPMAX - LPP0)/
     $                     (1. + EXP(-RGRLPP*(DAP(IB) - DAPLPP50)))))
60        CONTINUE

          DO 70 IP = IPATLOW,IPATCUM
*             Calculation of LAI and leaves per plant for each patch
              LAI(IP) = LAIBED(IPATDAT(1,IP))
              ILPP(IP) = ILPPBED(IPATDAT(1,IP))

*             Calculation of area foliage consumption for each patch
              FCPATCH(IP) = FCPATCH(IP) + FCR(IP)*DELT

*             Calculation of plant damage events in foraging
*             domains of L1 to L5 larvae per patch
              PDEFORL1(IP) = PDEFORL1(IP) + PDERFORL1(IP)*DELT
              PDEFORL2(IP) = PDEFORL2(IP) + PDERFORL2(IP)*DELT
              PDEFORL3(IP) = PDEFORL3(IP) + PDERFORL3(IP)*DELT
              PDEFORL4(IP) = PDEFORL4(IP) + PDERFORL4(IP)*DELT
              PDEFORL5(IP) = PDEFORL5(IP) + PDERFORL5(IP)*DELT

*             Calculation of damaged plants in foraging
*             domains of L1 to L5 larvae per patch
              DPFORL1(IP) = IPLANTL1*
     $                      (1.0-EXP(-PDEFORL1(IP)/IPLANTL1))
              DPFORL2(IP) = (IPLANTL2-IPLANTL1)*(1.0-EXP(-PDEFORL2(IP)/
     $                      (IPLANTL2-IPLANTL1)))
              DPFORL3(IP) = (IPLANTL3-IPLANTL2)*(1.0-EXP(-PDEFORL3(IP)/
     $                      (IPLANTL3-IPLANTL2)))
              DPFORL4(IP) = (IPLANTL4-IPLANTL3)*(1.0-EXP(-PDEFORL4(IP)/
     $                      (IPLANTL4-IPLANTL3)))
              DPFORL5(IP) = (IPLANTL5-IPLANTL4)*(1.0-EXP(-PDEFORL5(IP)/
     $                      (IPLANTL5-IPLANTL4)))

*             Calculation of damaged plants in patch
              DPPATCH(IP) =  DPFORL1(IP)+DPFORL2(IP)+DPFORL3(IP)+
     $                       DPFORL4(IP)+DPFORL5(IP)
70        CONTINUE

          DPCUM = 0.
          FCCUM = 0.
*         Calculation of cumulative foliage consumption and number of
*         damaged plants in greenhouse
          DO 80 IP = 1,IPATCUM
              DPCUM = DPCUM + DPPATCH(IP)
              FCCUM = FCCUM + FCPATCH(IP)
80        CONTINUE

          DPTOT = 0.
          FCTOT = 0.
*         Calculation of instantaneous foliage consumption and number of
*         damaged plants in greenhouse
          DO 90 IP = IPATLOW,IPATCUM
              DPTOT = DPTOT + DPPATCH(IP)
              FCTOT = FCTOT + FCPATCH(IP)
90        CONTINUE

*         Calculation of cumulative number of damaged plants per bed
*         and in greenhouse corrected for overlapping patches
          DPCUMO = 0.
          DO 110 IB = 1,IBEDMAX
              DPBED(IB) = 0.
              DPBEDO(IB) = 0.
              DO 100 IP = 1,IPATCUM
                  IF (IPATDAT(1,IP).EQ.IB) THEN
                      DPBED(IB) = DPBED(IB) + DPPATCH(IP)
                  ENDIF
100           CONTINUE
              DPBEDO(IB) = INPLANT*(1.-EXP(-DPBED(IB)/INPLANT))
              DPCUMO = DPCUMO + DPBEDO(IB)
110       CONTINUE

*         Calculation of instantaneous number of damaged plants per bed
*         and in greenhouse corrected for overlapping patches
          DPTOTO = 0.
          DO 130 IB = 1,IBEDMAX
              DPBED(IB) = 0.
              DPBEDO(IB) = 0.
              DO 120 IP = IPATLOW,IPATCUM
                  IF (IPATDAT(1,IP).EQ.IB) THEN
                      DPBED(IB) = DPBED(IB) + DPPATCH(IP)
                  ENDIF
120           CONTINUE
              DPBEDO(IB) = INPLANT*(1.-EXP(-DPBED(IB)/INPLANT))
              DPTOTO = DPTOTO + DPBEDO(IB)
130       CONTINUE
      ENDIF

*     Calculation of shift for CROP variables
      IF (ITASK.EQ.6) THEN
*         Determination of LAI and leaves per plant of newly planted bed
          IF (IHARV.NE.0) THEN
              DAP(IHARV) = 0.
              LAIBED(IHARV) = LAI0 + ((LAIMAX - LAI0)/
     $                    (1. + EXP(-RGRLAI*(DAP(IHARV) - DAPLAI50))))
              ILPPBED(IHARV) = NINT(LPP0 + ((LPPMAX - LPP0)/
     $                     (1. + EXP(-RGRLPP*(DAP(IHARV) - DAPLPP50)))))
          ENDIF

*         LAI and leaves per plant of new patches
          IF (IBATOUT.GT.0.OR.IBATOUTI.GT.0) THEN
```

```
              DO 140 IB = 1,(IBATOUT+IBATOUTI)
                  ILPP(IPATCUM-IB+1) = ILPPBED(IPATDAT(1,IPATCUM-IB+1))
                  LAI(IPATCUM-IB+1) = LAIBED(IPATDAT(1,IPATCUM-IB+1))
140           CONTINUE
          ENDIF
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE RIRHT                                                     *
*                                                                      *
* Purpose: Simulation of relative infection rate by                    *
*          horizontal transmission.                                    *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* ITASK    I4  Task that subroutine should perform (-)             I   *
* FCLVSL1 R4   Fraction contaminated leaves in L1 foraging domain      *
*              (-)                                                 I   *
* FCLVSL2 R4   Fraction contaminated leaves in L2 foraging domain      *
*              (-)                                                 I   *
* FCLVSL3 R4   Fraction contaminated leaves in L3 foraging domain      *
*              (-)                                                 I   *
* FCLVSL4 R4   Fraction contaminated leaves in L4 foraging domain      *
*              (-)                                                 I   *
* FCLVSL5 R4   Fraction contaminated leaves in L5 foraging domain      *
*              (-)                                                 I   *
* DELT     R4  Time step (d)                                       I   *
* IPATCUM I4   Cumulative number of patches in greenhouse (#)      I   *
* VERTDIST                                                             *
*          R4  Fraction larvae per canopy layer (-)                I   *
* IPATLOW I4   Pointer for lowest patch number containing larvae (-) I *
* LPDL1    R4  Mean number of leaf visits of L1 larvae per day (d-1) I *
* LPDL2    R4  Mean number of leaf visits of L2 larvae per day (d-1) I *
* LPDL3    R4  Mean number of leaf visits of L3 larvae per day (d-1) I *
* LPDL4    R4  Mean number of leaf visits of L4 larvae per day (d-1) I *
* LPDL5    R4  Mean number of leaf visits of L5 larvae per day (d-1) I *
* RIRL1HT R4   Relative infection rate by horizontal transmission      *
*              of L1 larvae (d-1)                                  O   *
* RIRL2HT R4   Relative infection rate by horizontal transmission      *
*              of L2 larvae (d-1)                                  O   *
* RIRL3HT R4   Relative infection rate by horizontal transmission      *
*              of L3 larvae (d-1)                                  O   *
* RIRL4HT R4   Relative infection rate by horizontal transmission      *
*              of L4 larvae (d-1)                                  O   *
* RIRL5HT R4   Relative infection rate by horizontal transmission      *
*              of L5 larvae (d-1)                                  O   *
*                                                                      *
* Fatal error checks: none                                             *
* warnings:           none                                             *
* Subroutines called:                                                  *
* from TTUTIL:        RDINIT,RDSREA,OUTDAT                             *
* File usage:         SPADIS.DAT                                       *
*----------------------------------------------------------------------*
      SUBROUTINE RIRHT(ITASK,FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $                 DELT,IPATCUM,VERTDIST,IPATLOW,LPDL1,LPDL2,
     $                 LPDL3,LPDL4,LPDL5,
     $                     RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT)

      PARAMETER (IPMAX = 200)
      PARAMETER (ICLMAX = 3)
      INTEGER   ICL,IP
      REAL      RIRCL1,RIRCL2,RIRCL3,RIRCL4,RIRCL5,LPDL1,LPDL2,
     $          LPDL3,LPDL4,LPDL5,RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,
     $          RIRL5HT,FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $          DELT,VERTDIST
      DIMENSION RIRCL1(IPMAX,4),RIRCL2(IPMAX,4),RIRCL3(IPMAX,4),
     $          RIRCL4(IPMAX,4),RIRCL5(IPMAX,4),RIRL1HT(IPMAX),
     $          RIRL2HT(IPMAX),RIRL3HT(IPMAX),RIRL4HT(IPMAX),
     $          RIRL5HT(IPMAX),FCLVSL1(IPMAX,ICLMAX),
     $          FCLVSL2(IPMAX,ICLMAX),FCLVSL3(IPMAX,ICLMAX),
     $          FCLVSL4(IPMAX,ICLMAX),FCLVSL5(IPMAX,ICLMAX),
     $          VERTDIST(3,5)
      SAVE

*     Initialization of RIRHT variables
      IF (ITASK.EQ.1) THEN
*         Read parameter values
          CALL RDINIT (30,40,'SPADIS.DAT')
          CALL RDSREA ('LPDL1',LPDL1)
          CALL RDSREA ('LPDL2',LPDL2)
          CALL RDSREA ('LPDL3',LPDL3)
          CALL RDSREA ('LPDL4',LPDL4)
          CALL RDSREA ('LPDL5',LPDL5)
      ENDIF

*     Calculation of RIRHT variables
      IF (ITASK.EQ.3) THEN
*       Loop over patches
        DO 20 IP = IPATLOW,IPATCUM
*           Calculation of relative infection rate by horizontal
*           transmission per canopy layer for each larval instar.
*           Loop over canopy layers
            DO 10 ICL = 1,3
              RIRCL1(IP,ICL) = (1./DELT)*(1.-(1.-FCLVSL1(IP,ICL))
     $                              **(LPDL1*DELT))
              RIRCL2(IP,ICL) = (1./DELT)*(1.-(1.-FCLVSL2(IP,ICL))
     $                              **(LPDL2*DELT))
              RIRCL3(IP,ICL) = (1./DELT)*(1.-(1.-FCLVSL3(IP,ICL))
     $                              **(LPDL3*DELT))
              RIRCL4(IP,ICL) = (1./DELT)*(1.-(1.-FCLVSL4(IP,ICL))
     $                              **(LPDL4*DELT))
              RIRCL5(IP,ICL) = (1./DELT)*(1.-(1.-FCLVSL5(IP,ICL))
     $                              **(LPDL5*DELT))
10          CONTINUE

*           Relative infection rate by horizontal transmission
*           per larval instar (1/day)
            RIRL1HT(IP) = RIRCL1(IP,1)*VERTDIST(1,1) +
     $                    RIRCL1(IP,2)*VERTDIST(2,1) +
     $                    RIRCL1(IP,3)*VERTDIST(3,1)
            RIRL2HT(IP) = RIRCL2(IP,1)*VERTDIST(1,2) +
     $                    RIRCL2(IP,2)*VERTDIST(2,2) +
     $                    RIRCL2(IP,3)*VERTDIST(3,2)
            RIRL3HT(IP) = RIRCL3(IP,1)*VERTDIST(1,3) +
     $                    RIRCL3(IP,2)*VERTDIST(2,3) +
     $                    RIRCL3(IP,3)*VERTDIST(3,3)
            RIRL4HT(IP) = RIRCL4(IP,1)*VERTDIST(1,4) +
     $                    RIRCL4(IP,2)*VERTDIST(2,4) +
     $                    RIRCL4(IP,3)*VERTDIST(3,4)
            RIRL5HT(IP) = RIRCL5(IP,1)*VERTDIST(1,5) +
     $                    RIRCL5(IP,2)*VERTDIST(2,5) +
     $                    RIRCL5(IP,3)*VERTDIST(3,5)
20        CONTINUE
      ENDIF

*     Output of RIRHT variables
      IF (ITASK.EQ.4) THEN
         CALL OUTDAT(2,20,'RIRL1HT',RIRL1HT(1))
         CALL OUTDAT(2,20,'RIRL2HT',RIRL2HT(1))
         CALL OUTDAT(2,20,'RIRL3HT',RIRL3HT(1))
         CALL OUTDAT(2,20,'RIRL4HT',RIRL4HT(1))
         CALL OUTDAT(2,20,'RIRL5HT',RIRL5HT(1))
      ENDIF

      RETURN
      END
```

```
*-----------------------------------------------------------------*
* SUBROUTINE MATING                                               *
*                                                                 *
* Purpose: Simulation of mating and initialization of new egg batches  *
*          in female oviduct.                                     *
*                                                                 *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time) *
* name    type meaning (unit)                               class *
* ----    ---- ---------------                              ----- *
* IMLTFEM I4  Emergence difference of female moths per time step  *
*             (#)                                            I   *
* IMLTFEMI                                                        *
*         I4  Emergence difference of sublethally infected female *
*             moths per time step (#)                        I   *
* IMLTMAL I4  Emergence difference of male moths per time step (#)  I   *
* IMLTMALI                                                        *
*         I4  Emergence difference of sublethally infected male   *
*             moths per time step (#)                        I   *
* ITOTMAL I4  Number of male moths in boxcar train (#)       I   *
* ITOTMALI                                                        *
*         I4  Number of sublethally infected male moths in boxcar *
*             train (#)                                      I   *
* IBATNR  I4  Number of deposited egg batches per female moth (#)  I   *
* IBATIN  I4  Number of batches in oviduct flowing in the IBATCH  *
*             boxcar train (#)                               O   *
* IBATINI I4  Number of infected batches in oviduct flowing in the *
*             IBATCHI boxcar train (#)                       O   *
*                                                                 *
* Fatal error checks: none                                        *
* warnings:           none                                        *
* Subroutines called: none                                        *
* File usage:         none                                        *
*-----------------------------------------------------------------*
      SUBROUTINE MATING(IMLTFEM,IMLTFEMI,IMLTMAL,IMLTMALI,
     $                  ITOTMAL,ITOTMALI,IBATNR,
     $                      IBATIN,IBATINI)

      INTEGER IMLTFEM,IMLTFEMI,ITOTMAL,ITOTMALI,IBATIN,
     $        IBATINI,I,IBATNR
      REAL    FINFMAL,LOT
      SAVE

*     Calculation of fraction infected male moths
      IF (ITOTMAL.EQ.0.AND.ITOTMALI.EQ.0) THEN
          FINFMAL = 0.
      ELSE
          FINFMAL = (ITOTMALI + IMLTMALI)/
     $              (ITOTMAL + ITOTMALI + IMLTMAL + IMLTMALI)
      ENDIF
```

```
*     Mating of uninfected female moths with uninfected or infected male moths.
*     All females mate as long as there are males in the greenhouse.
*     If there are no male moths in the greenhouse, emerged female moths
*     cannot mate. If male moths emerge later, the earlier emerged female
*     moths still will not mate.
      IBATIN = 0
      IBATINI = 0
      IF (((ITOTMAL + ITOTMALI + IMLTMAL + IMLTMALI).GT.0).AND.
     $    (IMLTFEM.GT.0)) THEN
*         Loop over uninfected female moths
          DO 10 I = 1,IMLTFEM
            LOT = UNIFL()
*             Mating of uninfected female moths with infected male moth
              IF (LOT.LE.FINFMAL) THEN
                  IBATINI = IBATINI + IBATNR
*             Mating of uninfected female moths with uninfected male moth
              ELSE
                  IBATIN = IBATIN + IBATNR
              ENDIF
10        CONTINUE
      ENDIF

*     Mating of infected female moth with male moth
      IF ((ITOTMAL + ITOTMALI + IMLTMAL + IMLTMALI).GT.0) THEN
*         Loop over infected female moths
          DO 20 I = 1,IMLTFEMI
            IBATINI = IBATINI + IBATNR
20        CONTINUE
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE SEDEV                                                     *
*                                                                      *
* Purpose: Simulation of Spodoptera exigua development and             *
*          population dynamics                                         *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                     class *
* ----   ---- ---------------                                    ----- *
* ITASK   I4  Task that subroutine should perform (-)               I  *
* TIME    R4  Time (d)                                              I  *
* DELT    R4  Time step (d)                                         I  *
* IHARV   I4  Flag for harvest (-)                                  I  *
* IPATDAT I4  Array containing co-ordinates of patches in              *
*             greenhouse (-)                                        I  *
* RIRL1HT R4  Relative infection rate by horizontal transmission       *
*             of L1 larvae (d-1)                                    I  *
* RIRL2HT R4  Relative infection rate by horizontal transmission       *
*             of L2 larvae (d-1)                                    I  *
* RIRL3HT R4  Relative infection rate by horizontal transmission       *
*             of L3 larvae (d-1)                                    I  *
* RIRL4HT R4  Relative infection rate by horizontal transmission       *
*             of L4 larvae (d-1)                                    I  *
* RIRL5HT R4  Relative infection rate by horizontal transmission       *
*             of L5 larvae (d-1)                                    I  *
* RIRL1S  R4  Relative infection rate by ingestion of sprayed          *
*             polyhedra of L1 larvae (d-1)                          I  *
* RIRL2S  R4  Relative infection rate by ingestion of sprayed          *
*             polyhedra of L2 larvae (d-1)                          I  *
* RIRL3SS R4  Relative infection rate by ingestion of sprayed          *
*             polyhedra of susceptible L3 larvae (d-1)              I  *
* RIRL3RS R4  Relative infection rate by ingestion of sprayed          *
*             polyhedra of resistant L3 larvae (d-1)                I  *
* RIRL4SS R4  Relative infection rate by ingestion of sprayed          *
*             polyhedra of susceptible L4 larvae (d-1)              I  *
* RIRL4RS R4  Relative infection rate by ingestion of sprayed          *
*             polyhedra of resistant L4 larvae (d-1)                I  *
* RIRL5SS R4  Relative infection rate by ingestion of sprayed          *
*             polyhedra of susceptible L5 larvae (d-1)              I  *
* RIRL5RS R4  Relative infection rate by ingestion of sprayed          *
*             polyhedra of resistant L5 larvae (d-1)                I  *
* IPATL1  I4  Number of L1 larvae in boxcar train per patch (#)     O  *
* IPATL2  I4  Number of L2 larvae in boxcar train per patch (#)     O  *
* IPATL3S I4  Number of susceptible L3 larvae in boxcar train per      *
*             patch (#)                                             O  *
* IPATL3R I4  Number of resistant L3 larvae in boxcar train per        *
*             patch (#)                                             O  *
* IPATL4S I4  Number of susceptible L4 larvae in boxcar train per      *
*             patch (#)                                             O  *
* IPATL4R I4  Number of resistant L4 larvae in boxcar train per        *
*             patch (#)                                             O *
* IPATL5S I4  Number of susceptible L5 larvae in boxcar train per     *
*             patch (#)                                             O *
* IPATL5R I4  Number of resistant L5 larvae in boxcar train per       *
*             patch (#)                                             O *
* IPATL1I I4  Number of infected L1 larvae in boxcar train per        *
*             patch (#)                                             O *
* IPATL2I I4  Number of infected L2 larvae in boxcar train per        *
*             patch (#)                                             O *
* IPATL3I I4  Number of infected L3 larvae in boxcar train per        *
*             patch (#)                                             O *
* IPATL4I I4  Number of infected L4 larvae in boxcar train per        *
*             patch (#)                                             O *
* ITIL5S  I4  Number of sublethally infected L5 larvae per boxcar     *
*             (#)                                                   O *
* ITIL5D  I4  Number of lethally infected L5 larvae per boxcar (#)  O *
* IDRL1I  I4  Death difference of infected L1 larvae per time step    *
*             (#)                                                   O *
* IDRL2I  I4  Death difference of infected L2 larvae per time step    *
*             (#)                                                   O *
* IDRL3I  I4  Death difference of infected L3 larvae per time step    *
*             (#)                                                   O *
* IDRL4I  I4  Death difference of infected L4 larvae per time step    *
*             (#)                                                   O *
* IDRL5ID I4  Death difference of lethally infected L5 larvae per     *
*             time step (#)                                         O *
* IMLTMAL I4  Emergence difference of male moths per time step (#)  O *
* IMLTFEM I4  Emergence difference of female moths per time step      *
*             (#)                                                   O *
* IMLTMALI                                                            *
*         I4  Emergence difference of sublethally infected male       *
*             moths per time step (#)                               O *
* IMLTFEMI                                                            *
*         I4  Emergence difference of sublethally infected female     *
*             moths per time step (#)                               O *
* IDRMAL  I4  Death difference of male moths per time step (#)      O *
* IDRFEM  I4  Death difference of female moths per time step (#)    O *
* IDRMALI I4  Death difference of sublethally infected male moths     *
*             per time step (#)                                     O *
* IDRFEMI I4  Death difference of sublethally infected female         *
*             moths per time step (#)                               O *
* IRL1HT  I4  Infection difference by horizontal transmission of      *
*             L1 larvae per timestep (#)                            O *
* IRL2HT  I4  Infection difference by horizontal transmission of      *
*             L2 larvae per timestep (#)                            O *
* IRL3SHT I4  Infection difference by horizontal transmission of      *
*             susceptible L3 larvae per timestep (#)                O *
* IRL3RHT I4  Infection difference by horizontal transmission of      *
*             resistant L3 larvae per timestep (#)                  O *
* IRL4SHT I4  Infection difference by horizontal transmission of      *
*               susceptible L4 larvae per timestep (#)            O  *
* IRL4RHT I4    Infection difference by horizontal transmission of   *
*               resistant L4 larvae per timestep (#)              O  *
* IRL5SHTS                                                           *
*         I4    Infection difference by horizontal transmission of   *
*               susceptible L5 larvae into sublethally infected L5   *
*               larvae per timestep (#)                           O  *
* IRL5RHTS                                                           *
*         I4    Infection difference by horizontal transmission of   *
*               resistant L5 larvae into sublethally infected L5     *
*               larvae per timestep (#)                           O  *
* IRL5SHTD                                                           *
*         I4    Infection difference by horizontal transmission of   *
*               susceptible L5 larvae into lethally infected L5      *
*               larvae per timestep (#)                           O  *
* IRL5RHTD                                                           *
*         I4    Infection difference by horizontal transmission of   *
*               resistant L5 larvae into lethally infected L5 larvae *
*               per timestep (#)                                  O  *
* IRL1S   I4    Infection difference by ingestion of sprayed         *
*               polyhedra of L1 larvae per timestep (#)           O  *
* IRL2S   I4    Infection difference by ingestion of sprayed         *
*               polyhedra of L2 larvae per timestep (#)           O  *
* IRL3SS  I4    Infection difference by ingestion of sprayed         *
*               polyhedra of susceptible L3 larvae per timestep (#) O *
* IRL3RS  I4    Infection difference by ingestion of sprayed         *
*               polyhedra of resistant L3 larvae per timestep (#) O  *
* IRL4SS  I4    Infection difference by ingestion of sprayed         *
*               polyhedra of susceptible L4 larvae per timestep (#) O *
* IRL4RS  I4    Infection difference by ingestion of sprayed         *
*               polyhedra of resistant L4 larvae per timestep (#) O  *
* IRL5SSS I4    Infection difference by ingestion of sprayed         *
*               polyhedra of susceptible L5 larvae into sublethally  *
*               infected L5 larvae per timestep (#)               O  *
* IRL5RSS I4    Infection difference by ingestion of sprayed         *
*               polyhedra of resistant L5 larvae into sublethally    *
*               infected L5 larvae per timestep (#)               O  *
* IRL5SSD I4    Infection difference by ingestion of sprayed         *
*               polyhedra of susceptible L5 larvae into lethally     *
*               infected L5 larvae per timestep (#)               O  *
* IRL5RSD I4    Infection difference by ingestion of sprayed         *
*               polyhedra of resistant L5 larvae into lethally       *
*               infected L5 larvae per timestep (#)               O  *
* ITOTMAL I4    Number of male moths in boxcar train (#)          O  *
* ITOTFEM I4    Number of female moths in boxcar train (#)        O  *
* ITOTMALI                                                           *
*         I4    Number of sublethally infected male moths in boxcar  *
*               train (#)                                         O  *
* ITOTFEMI                                                           *
*         I4    Number of sublethally infected female moths in       *
*              boxcar train (#)                                      O  *
* IBATIN   I4  Number of batches in oviduct flowing in the IBATCH       *
*              boxcar train (#)                                      O  *
* IBATINI  I4  Number of infected batches in oviduct flowing in the     *
*              IBATCHI boxcar train (#)                              O  *
* IBATOUT  I4  Number of batches in oviduct flowing out IBATCH          *
*              boxcar train (#)                                      O  *
* IBATOUTI                                                              *
*          I4  Number of batches in infected oviduct flowing out        *
*              IBATCHI boxcar train (#)                              O  *
* IHATCHI  I4  Hatch difference of infected L1 larvae per patch per     *
*              time step (#)                                         O  *
* IPATLOW  I4  Pointer for lowest patch number containing larvae (-) O  *
* IPATCUM  I4  Cumulative number of patches in greenhouse (#)        O  *
* DVRL1    R4  Development rate of L1 larvae (d-1)                   O  *
* DVRL2    R4  Development rate of L2 larvae (d-1)                   O  *
* DVRL3    R4  Development rate of L3 larvae (d-1)                   O  *
* DVRL4    R4  Development rate of L4 larvae (d-1)                   O  *
* DVRL5    R4  Development rate of L5 larvae (d-1)                   O  *
* DVRL1I   R4  Development rate of infected L1 larvae (d-1)          O  *
* DVRL2I   R4  Development rate of infected L2 larvae (d-1)          O  *
* DVRL3I   R4  Development rate of infected L3 larvae (d-1)          O  *
* DVRL4I   R4  Development rate of infected L4 larvae (d-1)          O  *
* DVRL5ID  R4  Development rate of lethally infected L5 larvae (d-1) O  *
* ITOTPUP  I4  Number of pupea in boxcar train (#)                   O  *
* ITOTPUPI                                                              *
*          I4  Number of sublethally infected pupea in boxcar train     *
*              (#)                                                   O  *
*                                                                       *
* Fatal error checks: Check for individuals in non-existing patches     *
*                     FSURV < 0, FSURV > 1                              *
*                     IPATCUM >IPMAX                                    *
* warnings:           none                                              *
* Subroutines called:                                                   *
* from TTUTIL:        RDINIT,RDSINT,RDSREA,RDAREA,RDFINT,RDFREA,        *
*                     OUTDAT                                            *
* own:                IBOXCAR,IBOXPUP,IBOXPATL5,IBOXPAT,IBOXPATL2       *
*                     IBOXPTIS,IBOXPTI,MATING,ACCOUNT,DEPOSIT           *
* File usage:         INCON.DAT, SEDEV.DAT, VIRUS.DAT                   *
*-----------------------------------------------------------------------*
      SUBROUTINE SEDEV(ITASK,TIME,DELT,IHARV,IPATDAT,RIRL1HT,
     $                 RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT,RIRL1S,RIRL2S,
     $                 RIRL3SS,RIRL3RS,RIRL4SS,RIRL4RS,RIRL5SS,RIRL5RS,
     $                   IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,IPATL4R,
     $                   IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,
     $                   IPATL4I,ITIL5S,ITIL5D,IDRL1I,IDRL2I,IDRL3I,
     $                   IDRL4I,IDRL5ID,IMLTMAL,IMLTFEM,IMLTMALI,
     $                   IMLTFEMI,IDRMAL,IDRFEM,IDRMALI,IDRFEMI,IRL1HT,
     $                   IRL2HT,IRL3SHT,IRL3RHT,IRL4SHT,IRL4RHT,
     $                   IRL5SHTS,IRL5RHTS,IRL5SHTD,IRL5RHTD,IRL1S,
     $                   IRL2S,IRL3SS,IRL3RS,IRL4SS,IRL4RS,IRL5SSS,
     $                   IRL5RSS,IRL5SSD,IRL5RSD,ITOTMAL,ITOTFEM,
     $                   ITOTMALI,ITOTFEMI,IBATIN,IBATINI,IBATOUT,
     $                   IBATOUTI,IHATCHI,IPATLOW,IPATCUM,DVRL1,
     $                   DVRL2,DVRL3,DVRL4,DVRL5,DVRL1I,DVRL2I,DVRL3I,
     $                   DVRL4I,DVRL5ID,ITOTPUP,ITOTPUPI)

      PARAMETER (IPMAX = 200)
      INTEGER   ITASK,IPATCUM,IP,IPATL1,IPATL2,IPATL3S,IPATL3R,IPATL4S,
     $          IPATL4R,IPATL5S,IPATL5R,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $          ITIL5S,ITIL5D,ITOTBAT,ITOTBATI,IDRL1I,IDRL2I,IDRL3I,
     $          IDRL4I,IDRL5ID,IMLTMAL,IMLTFEM,IMLTMALI,IMLTFEMI,
     $          IDRMAL,IDRMALI,IDRFEMI,IBATCH,IBATCHI,IEGG,IL1,IL2,
     $          IL3,IL3S,IL3R,IL4,IL4S,IL4R,IL5,IL5S,IL5R,IPUP,IMAL,
     $          IFEM,IEGGI,IL1I,IL2I,IL3I,IL4I,IPUPI,IMALI,IFEMI,IL5IS,
     $          IL5ID,IPATEGG,IPATEGGI,IRL1HT,IRL2HT,IRL3SHT,IRL3RHT,
     $          IRL4SHT,IRL4RHT,IRL5SHTS,IRL5RHTS,IRL5SHTD,IRL5RHTD,
     $          IATREGG,IATRL1,IATRL2,IATRL3S,IATRL3R,IATRL4S,IATRL4R,
     $          IATRL5S,IATRL5R,IATRPUP,IATRMAL,IATRFEM,IATRBAT,
     $          IATRL5IS,IATRPUPI,IATRMALI,IATRFEMI,IATRBATI,IHATCH,
     $          IMLTL2,IMLTL3S,IMLTL3R,IMLTL4S,IMLTL4R,IMLTL5S,IMLTL5R,
     $          IMLTPUPS,IMLTPUPR,IHATCHI,IMLTPUPI,IMOULTP,IMOULTPI,
     $          ITOTPUP,ITOTPUPI,ITOTMAL,ITOTFEM,ITOTMALI,ITOTFEMI,
     $          IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,IRL4RS,IRL5SSD,IRL5RSD,
     $          IRL5SSS,IRL5RSS,IBATIN,IBATINI,IBATOUT,IBATOUTI,IHARV,
     $          IPATDAT,IBATNR,I,ICTEMP,ICBATSIZE,INBATINFL,ISTOCH
      REAL      RIRL1HT,RIRL2HT,RIRL3HT,RIRL4HT,RIRL5HT,LINT,CDSBATCH,
     $          CDSBATCHI,CDSEGG,CDSL1,CDSL2,CDSL3S,CDSL4S,CDSL5S,
     $          CDSL3R,CDSL4R,CDSL5R,CDSPUP,CDSMAL,CDSFEM,CDSEGGI,
     $          CDSL1I,CDSL2I,CDSL3I,CDSL4I,CDSL5IS,CDSL5ID,CDSPUPI,
     $          CDSMALI,CDSFEMI,DVRBATT,DVREGGT,DVRL1T,DVRL2T,DVRL3T,
     $          DVRL4T,DVRL5T,DVRPUPT,DVRMALT,DVRFEMT,DVREGGIT,DVRL1IT,
     $          DVRL2IT,DVRL3IT,DVRL4IT,DVRL5IDT,DVRL5IST,DVRPUPIT,
     $          DVRMALIT,DVRFEMIT,SDEGG,SDL1,SDL2,SDL3,SDL4,SDL5,
     $          SDPUP,SDMAL,SDFEM,SDBAT,DVRBATCH,DVREGG,DVRL1,DVRL2,
     $          DVRL3,DVRL4,DVRL5,DVRPUP,DVRMAL,DVRFEM,DVREGGI,DVRL1I,
     $          DVRL2I,DVRL3I,DVRL4I,DVRL5ID,DVRL5IS,DVRPUPI,DVRMALI,
     $          DVRFEMI,DVRBATCHI,DVRBATIT,CVBATCH,CVEGG,CVL1,CVL2,
     $          CVL3,CVL4,CVL5,CVPUP,CVMAL,CVFEM,RIRL1S,RIRL2S,RIRL3RS,
     $          RIRL4SS,RIRL4RS,RIRL5SS,RIRL5RS,SEXRAT,X,FX,RATRL1,
     $          RATRL2,RATRL3,RATRL4,RATRL5,RATRPUP,FCBATS,FCBATL,
     $          FCEGGS,FCEGGL,FINFL5S,SDEGGI,SDL1I,SDL2I,SDL3I,SDL4I,
     $          SDL5IS,SDL5ID,SDPUPI,SDFEMI,SDMALI,SDBATI,TEMPT,
     $          FSURVFEM,FSURVMAL,FSURVL1,FSURVL2,FSURVL3,FSURVL4,
     $          FSURVL5,FSURVPUP,CFDVR,TBATINFL,TIME
      DIMENSION IPATL1(IPMAX),IPATL2(IPMAX),IPATL3S(IPMAX),
     $          IPATL3R(IPMAX),IPATL4S(IPMAX),IPATL4R(IPMAX),
     $          IPATL5S(IPMAX),IPATL5R(IPMAX),IPATL1I(IPMAX),
     $          IPATL2I(IPMAX),IPATL3I(IPMAX),IPATL4I(IPMAX),
     $          ITIL5S(IPMAX),ITIL5D(IPMAX),IDRL1I(IPMAX),
     $          IDRL2I(IPMAX),IDRL3I(IPMAX),IDRL4I(IPMAX),
     $          IDRL5ID(IPMAX),RIRL1HT(IPMAX),RIRL2HT(IPMAX),
     $          RIRL3HT(IPMAX),RIRL4HT(IPMAX),RIRL5HT(IPMAX),
     $          IBATCH(0:4),IBATCHI(0:4),IEGG(IPMAX,0:4),
     $          IL1(IPMAX,0:4),IL2(IPMAX,0:4),IL3(IPMAX,0:4),
     $          IL3S(IPMAX,0:4),IL3R(IPMAX,0:4),IL4(IPMAX,0:4),
     $          IL4S(IPMAX,0:4),IL4R(IPMAX,0:4),IL5(IPMAX,0:4),
     $          IL5S(IPMAX,0:4),IL5R(IPMAX,0:4),IPUP(0:4),IMAL(0:2),
     $          IFEM(0:2),IEGGI(IPMAX,0:4),IL1I(IPMAX,0:4),
     $          IL2I(IPMAX,0:4),IL3I(IPMAX,0:4),IL4I(IPMAX,0:4),
     $          IPUPI(0:4),IMALI(0:2),IFEMI(0:2),IL5IS(IPMAX,0:4),
     $          IL5ID(IPMAX,0:4),IPATEGG(IPMAX),IPATEGGI(IPMAX),
     $          IRL1HT(IPMAX),IRL2HT(IPMAX),IRL3SHT(IPMAX),
     $          IRL3RHT(IPMAX),IRL4SHT(IPMAX),IRL4RHT(IPMAX),
     $          IRL5SHTS(IPMAX),IRL5RHTS(IPMAX),IRL5SHTD(IPMAX),
     $          IRL5RHTD(IPMAX),IATREGG(IPMAX),IATRL1(IPMAX),
     $          IATRL2(IPMAX),IATRL3S(IPMAX),IATRL3R(IPMAX),
     $          IATRL4S(IPMAX),IATRL4R(IPMAX),IATRL5S(IPMAX),
     $          IATRL5R(IPMAX),IATRL5IS(IPMAX),IHATCH(IPMAX),
     $          IMLTL2(IPMAX),IMLTL3S(IPMAX),IMLTL3R(IPMAX),
     $          IMLTL4S(IPMAX),IMLTL4R(IPMAX),IMLTL5S(IPMAX),
     $          IMLTL5R(IPMAX),IMLTPUPS(IPMAX),IMLTPUPR(IPMAX),
     $          IHATCHI(IPMAX),IMLTPUPI(IPMAX),CDSEGG(IPMAX),
     $          CDSL1(IPMAX),CDSL2(IPMAX),CDSL3S(IPMAX),
     $          CDSL4S(IPMAX),CDSL5S(IPMAX),CDSL3R(IPMAX),
     $          CDSL4R(IPMAX),CDSL5R(IPMAX),CDSEGGI(IPMAX),
     $          CDSL1I(IPMAX),CDSL2I(IPMAX),CDSL3I(IPMAX),
     $          CDSL4I(IPMAX),CDSL5IS(IPMAX),CDSL5ID(IPMAX),
     $          DVRBATT(12),DVREGGT(12),DVRL1T(12),DVRL2T(12),
     $          DVRL3T(12),DVRL4T(12),DVRL5T(12),DVRPUPT(12),
     $          DVRMALT(12),DVRFEMT(12),DVREGGIT(12),DVRL1IT(12),
     $          DVRL2IT(12),DVRL3IT(12),DVRL4IT(12),DVRL5IDT(12),
     $          DVRL5IST(12),DVRPUPIT(12),DVRMALIT(12),DVRFEMIT(12)
      DIMENSION DVRBATIT(12),SDEGG(12),SDL1(12),SDL2(12),SDL3(12),
     $          SDL4(12),SDL5(12),SDPUP(12),SDMAL(12),SDFEM(12),
     $          SDBAT(12),RIRL1S(IPMAX),RIRL2S(IPMAX),RIRL3SS(IPMAX),
     $          RIRL3RS(IPMAX),RIRL4SS(IPMAX),RIRL4RS(IPMAX),
     $          RIRL5SS(IPMAX),RIRL5RS(IPMAX),IRL1S(IPMAX),
     $          IRL2S(IPMAX),IRL3SS(IPMAX),IRL3RS(IPMAX),
     $          IRL4SS(IPMAX),IRL4RS(IPMAX),IRL5SSD(IPMAX),
     $          IRL5RSD(IPMAX),IRL5SSS(IPMAX),IRL5RSS(IPMAX),
     $          IPATDAT(3,IPMAX),X(110),FX(110),IRL5SHTSAR(IPMAX,0:4),
     $          IRL5RHTSAR(IPMAX,0:4),IRL5SSSAR(IPMAX,0:4),
     $          IRL5RSSAR(IPMAX,0:4),SDEGGI(12),SDL1I(12),SDL2I(12),
     $          SDL3I(12),SDL4I(12),SDL5ID(12),SDL5IS(12),SDPUPI(12),
     $          SDFEMI(12),SDMALI(12),SDBATI(12),TEMPT(1920),
     $          TBATINFL(50)
      SAVE

*     Initialization of SEDEV variables
```

```
      IF (ITASK.EQ.1) THEN
*         Read initial conditions
          CALL RDINIT (30,40,'INCON.DAT')
          CALL RDSINT ('ISTOCH',ISTOCH)
          CALL RDSINT ('INBATINFL',INBATINFL)
          CALL RDAREA ('TBATINFL',TBATINFL,50,50)
*         temperature switch
          CALL RDSINT ('ICTEMP',ICTEMP)
*         Constant temperature in greenhouse when ICTEMP is set to 1
          CALL RDSREA ('TEMP',TEMP)
*         Measured temperatures in greenhouse when ICTEMP is set to 0
          CALL RDAREA ('TEMPT',TEMPT,1920,1920)
*         Uninfected insects
          CALL RDFINT ('IEGG',IEGG(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL1',IL1(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL2',IL2(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL3',IL3(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL4',IL4(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL5',IL5(1,0),1000,5*IPMAX)
          CALL RDFINT ('IPUP',IPUP(0),5,5)
          CALL RDFINT ('IMAL ',IMAL(0),3,3)
          CALL RDFINT ('IFEM ',IFEM(0),3,3)
          CALL RDFINT ('IBATCH',IBATCH(0),5,5)
*         Infected insects
          CALL RDFINT ('IEGGI',IEGGI(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL1I',IL1I(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL2I',IL2I(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL3I',IL3I(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL4I',IL4I(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL5ID',IL5ID(1,0),1000,5*IPMAX)
          CALL RDFINT ('IL5IS',IL5IS(1,0),1000,5*IPMAX)
          CALL RDFINT ('IPUPI',IPUPI(0),5,5)
          CALL RDFINT ('IMALI',IMALI(0),3,3)
          CALL RDFINT ('IFEMI',IFEMI(0),3,3)
          CALL RDFINT ('IBATCHI',IBATCHI(0),5,5)
*         Read parameter values
          CALL RDINIT (30,40,'SEDEV.DAT')
*         Development rates of uninfected insects as function of temperature
          CALL RDFREA ('DVREGGT',DVREGGT(1),12,12)
          CALL RDFREA ('DVRL1T',DVRL1T(1),12,12)
          CALL RDFREA ('DVRL2T',DVRL2T(1),12,12)
          CALL RDFREA ('DVRL3T',DVRL3T(1),12,12)
          CALL RDFREA ('DVRL4T',DVRL4T(1),12,12)
          CALL RDFREA ('DVRL5T',DVRL5T(1),12,12)
          CALL RDFREA ('DVRPUPT',DVRPUPT(1),12,12)
          CALL RDFREA ('DVRMALT',DVRMALT(1),12,12)
          CALL RDFREA ('DVRFEMT',DVRFEMT(1),12,12)
          CALL RDFREA ('DVRBATT',DVRBATT(1),12,12)
*         Conversion factor of larval development rate from artificial diet
*         to chrysanthemum
```

```
      CALL RDSREA ('CFDVR' ,CFDVR)
*     Standard deviation of development rates of insects as function
*     of temperature
      CALL RDFREA ('SDEGG',SDEGG(1),12,12)
      CALL RDFREA ('SDL1',SDL1(1),12,12)
      CALL RDFREA ('SDL2',SDL2(1),12,12)
      CALL RDFREA ('SDL3',SDL3(1),12,12)
      CALL RDFREA ('SDL4',SDL4(1),12,12)
      CALL RDFREA ('SDL5',SDL5(1),12,12)
      CALL RDFREA ('SDPUP',SDPUP(1),12,12)
      CALL RDFREA ('SDMAL',SDMAL(1),12,12)
      CALL RDFREA ('SDFEM',SDFEM(1),12,12)
      CALL RDFREA ('SDBAT',SDBAT(1),12,12)
*     Fraction survival of development stages in absence of virus
      CALL RDSREA ('FSURVL1' ,FSURVL1)
      CALL RDSREA ('FSURVL2' ,FSURVL2)
      CALL RDSREA ('FSURVL3' ,FSURVL3)
      CALL RDSREA ('FSURVL4' ,FSURVL4)
      CALL RDSREA ('FSURVL5' ,FSURVL5)
      CALL RDSREA ('FSURVPUP',FSURVPUP)
      CALL RDSREA ('FSURVFEM' ,FSURVFEM)
      CALL RDSREA ('FSURVMAL' ,FSURVMAL)
*     Sexratio of insects
      CALL RDSREA ('SEXRAT',SEXRAT)
*     Number of deposited egg batches per female moth
      CALL RDSINT ('IBATNR',IBATNR)
*     Fraction of infected L5 larvae that will become sublethally infected
      CALL RDSREA ('FINFL5S',FINFL5S)
*     Critical egg batch size to be a large egg batch
      CALL RDSINT ('ICBATSIZE',ICBATSIZE)
*     Fraction of small egg batches that are contaminated by sublethally
*     infected adults
      CALL RDSREA ('FCBATS',FCBATS)
*     Fraction of large egg batches that are contaminated by sublethally
*     infected adults
      CALL RDSREA ('FCBATL',FCBATL)
*     Maximal fraction contaminated eggs in small egg batches
      CALL RDSREA ('FCEGGS',FCEGGS)
*     Maximal fraction contaminated eggs in large egg batches
      CALL RDSREA ('FCEGGL',FCEGGL)
*     Cumulative egg batch size distribution
      CALL RDAREA ('FX',FX,110,110)
*     Egg batch size
      CALL RDAREA ('X',X,110,110)
      CALL RDINIT (30,40,'VIRUS.DAT')
*     Development rates of infected insects as function of temperature
      CALL RDFREA ('DVREGGIT',DVREGGIT(1),12,12)
      CALL RDFREA ('DVRL1IT',DVRL1IT(1),12,12)
      CALL RDFREA ('DVRL2IT',DVRL2IT(1),12,12)
      CALL RDFREA ('DVRL3IT',DVRL3IT(1),12,12)
```

```
          CALL RDFREA ('DVRL4IT',DVRL4IT(1),12,12)
          CALL RDFREA ('DVRL5IDT',DVRL5IDT(1),12,12)
          CALL RDFREA ('DVRL5IST',DVRL5IST(1),12,12)
          CALL RDFREA ('DVRPUPIT',DVRPUPIT(1),12,12)
          CALL RDFREA ('DVRMALIT',DVRMALIT(1),12,12)
          CALL RDFREA ('DVRFEMIT',DVRFEMIT(1),12,12)
          CALL RDFREA ('DVRBATIT',DVRBATIT(1),12,12)
*         Standard deviation of development rates of infected insects
*         as function of temperature
          CALL RDFREA ('SDEGGI',SDEGGI(1),12,12)
          CALL RDFREA ('SDL1I',SDL1I(1),12,12)
          CALL RDFREA ('SDL2I',SDL2I(1),12,12)
          CALL RDFREA ('SDL3I',SDL3I(1),12,12)
          CALL RDFREA ('SDL4I',SDL4I(1),12,12)
          CALL RDFREA ('SDL5ID',SDL5ID(1),12,12)
          CALL RDFREA ('SDL5IS',SDL5IS(1),12,12)
          CALL RDFREA ('SDPUPI',SDPUPI(1),12,12)
          CALL RDFREA ('SDMALI',SDMALI(1),12,12)
          CALL RDFREA ('SDFEMI',SDFEMI(1),12,12)
          CALL RDFREA ('SDBATI',SDBATI(1),12,12)

*         Division of populations L3, L4 and L5 larvae in suseptible
*         and resistant sub-populations.
          DO 20 IP = 1,IPMAX
              DO 10 I = 0,4
                  IL3S(IP,I) = IROUND(IL3(IP,I)*0.5)
                  IL3R(IP,I) = IL3(IP,I) - IL3S(IP,I)
                  IL4S(IP,I) = IROUND(IL4(IP,I)*0.5)
                  IL4R(IP,I) = IL4(IP,I) - IL4S(IP,I)
                  IL5S(IP,I) = IROUND(IL5(IP,I)*0.5)
                  IL5R(IP,I) = IL5(IP,I) - IL5S(IP,I)
10            CONTINUE
*             Initialization of death rates
              IDRL1I(IP) = 0
              IDRL2I(IP) = 0
              IDRL3I(IP) = 0
              IDRL4I(IP) = 0
              IDRL5ID(IP) = 0
20        CONTINUE

*         Check for individuals in non-existing patches
          DO 40 IP = IPATCUM+1,IPMAX
              DO 30 I = 0,4
              IF ((IEGG(IP,I).NE.0).OR.
     $            (IL1(IP,I).NE.0).OR.
     $            (IL2(IP,I).NE.0).OR.
     $            (IL3S(IP,I).NE.0).OR.
     $            (IL3R(IP,I).NE.0).OR.
     $            (IL4S(IP,I).NE.0).OR.
     $            (IL4R(IP,I).NE.0).OR.
     $            (IL5S(IP,I).NE.0).OR.
     $            (IL5R(IP,I).NE.0).OR.
     $            (IEGGI(IP,I).NE.0).OR.
     $            (IL1I(IP,I).NE.0).OR.
     $            (IL2I(IP,I).NE.0).OR.
     $            (IL3I(IP,I).NE.0).OR.
     $            (IL4I(IP,I).NE.0).OR.
     $            (IL5ID(IP,I).NE.0).OR.
     $            (IL5IS(IP,I).NE.0))
     $             STOP'Population initialization error!'
30            CONTINUE
40        CONTINUE

*         Check for impossible input data for attrition calculation
          IF(FSURVL1.LE.0.OR.FSURVL1.GT.1.)
     $    STOP'Attrition initialization error for L1 larvae!'
          IF(FSURVL2.LE.0.OR.FSURVL2.GT.1.)
     $    STOP'Attrition initialization error for L2 larvae!'
          IF(FSURVL3.LE.0.OR.FSURVL3.GT.1.)
     $    STOP'Attrition initialization error for L3 larvae!'
          IF(FSURVL4.LE.0.OR.FSURVL4.GT.1.)
     $    STOP'Attrition initialization error for L4 larvae!'
          IF(FSURVL5.LE.0.OR.FSURVL5.GT.1.)
     $    STOP'Attrition initialization error for L5 larvae!'
          IF(FSURVPUP.LE.0.OR.FSURVPUP.GT.1.)
     $    STOP'Attrition initialization error for pupae!'
          IF(FSURVMAL.LE.0.OR.FSURVMAL.GT.1.)
     $    STOP'Attrition initialization error for male moths!'
          IF(FSURVFEM.LE.0.OR.FSURVFEM.GT.1.)
     $    STOP'Attrition initialization error for female moths!'
      ENDIF

*     Calculation of SEDEV forcing functions
      IF (ITASK.EQ.2) THEN
*         Reading measured greenhouse temperatures
          IF (ICTEMP.NE.1) THEN
              TEMP = LINT(TEMPT, 1920, TIME)
          ENDIF

*         Development rates of uninfected insects
          DVREGG   = LINT(DVREGGT,  12, TEMP)
          DVRL1    = LINT(DVRL1T,   12, TEMP) * CFDVR
          DVRL2    = LINT(DVRL2T,   12, TEMP) * CFDVR
          DVRL3    = LINT(DVRL3T,   12, TEMP) * CFDVR
          DVRL4    = LINT(DVRL4T,   12, TEMP) * CFDVR
          DVRL5    = LINT(DVRL5T,   12, TEMP) * CFDVR
          DVRPUP   = LINT(DVRPUPT,  12, TEMP)
          DVRMAL   = LINT(DVRMALT,  12, TEMP)
          DVRFEM   = LINT(DVRFEMT,  12, TEMP)
          DVRBATCH = LINT(DVRBATT,  12, TEMP)
```

```
*     Development rates of infected insects
      DVREGGI   = LINT(DVREGGIT, 12, TEMP)
      DVRL1I    = LINT(DVRL1IT,  12, TEMP)
      DVRL2I    = LINT(DVRL2IT,  12, TEMP)
      DVRL3I    = LINT(DVRL3IT,  12, TEMP)
      DVRL4I    = LINT(DVRL4IT,  12, TEMP)
      DVRL5ID   = LINT(DVRL5IDT, 12, TEMP)
      DVRL5IS   = LINT(DVRL5IST, 12, TEMP)
      DVRPUPI   = LINT(DVRPUPIT, 12, TEMP)
      DVRMALI   = LINT(DVRMALIT, 12, TEMP)
      DVRFEMI   = LINT(DVRFEMIT, 12, TEMP)
      DVRBATCHI = LINT(DVRBATIT, 12, TEMP)
*     Coefficient of variation of uninfected insects
      CVEGG   = LINT(SDEGG,   12, TEMP) * DVREGG
      CVL1    = LINT(SDL1,    12, TEMP) * DVRL1
      CVL2    = LINT(SDL2,    12, TEMP) * DVRL2
      CVL3    = LINT(SDL3,    12, TEMP) * DVRL3
      CVL4    = LINT(SDL4,    12, TEMP) * DVRL4
      CVL5    = LINT(SDL5,    12, TEMP) * DVRL5
      CVPUP   = LINT(SDPUP,   12, TEMP) * DVRPUP
      CVMAL   = LINT(SDMAL,   12, TEMP) * DVRMAL
      CVFEM   = LINT(SDFEM,   12, TEMP) * DVRFEM
      CVBATCH = LINT(SDBAT,   12, TEMP) * DVRBATCH
*     Coefficient of variation of infected insects
      CVEGGI   = LINT(SDEGGI,   12, TEMP) * DVREGGI
      CVL1I    = LINT(SDL1I,    12, TEMP) * DVRL1I
      CVL2I    = LINT(SDL2I,    12, TEMP) * DVRL2I
      CVL3I    = LINT(SDL3I,    12, TEMP) * DVRL3I
      CVL4I    = LINT(SDL4I,    12, TEMP) * DVRL4I
      CVL5ID   = LINT(SDL5ID,   12, TEMP) * DVRL5ID
      CVL5IS   = LINT(SDL5IS,   12, TEMP) * DVRL5IS
      CVPUPI   = LINT(SDPUPI,   12, TEMP) * DVRPUPI
      CVMALI   = LINT(SDMALI,   12, TEMP) * DVRMALI
      CVFEMI   = LINT(SDFEMI,   12, TEMP) * DVRFEMI
      CVBATCHI = LINT(SDBATI,   12, TEMP) * DVRBATCHI

*     Initialization of cyclic development stages of uninfected
*     and infected insects
      IF (TIME.LE.(0.5*DELT)) THEN
          DO 50 IP = 1,IPMAX
              CDSEGG(IP)  = 0.125-0.5*CVEGG**2
              CDSL1(IP)   = 0.125-0.5*CVL1**2
              CDSL2(IP)   = 0.125-0.5*CVL2**2
              CDSL3R(IP)  = 0.125-0.5*CVL3**2
              CDSL3S(IP)  = 0.125-0.5*CVL3**2
              CDSL4R(IP)  = 0.125-0.5*CVL4**2
              CDSL4S(IP)  = 0.125-0.5*CVL4**2
              CDSL5R(IP)  = 0.125-0.5*CVL5**2
              CDSL5S(IP)  = 0.125-0.5*CVL5**2
              CDSEGGI(IP) = 0.125-0.5*CVEGGI**2
```

```
                  CDSL1I(IP)  = 0.125-0.5*CVL1I**2
                  CDSL2I(IP)  = 0.125-0.5*CVL2I**2
                  CDSL3I(IP)  = 0.125-0.5*CVL3I**2
                  CDSL4I(IP)  = 0.125-0.5*CVL4I**2
                  CDSL5IS(IP)  = 0.125-0.5*CVL5IS**2
                  CDSL5ID(IP)  = 0.125-0.5*CVL5ID**2
50            CONTINUE
              CDSPUP     = 0.25-0.5*CVPUP**2
              CDSMAL     = 0.25-0.5*CVMAL**2
              CDSFEM     = 0.25-0.5*CVFEM**2
              CDSBATCH   = 0.125-0.5*CVBATCH**2
              CDSPUPI    = 0.25-0.5*CVPUPI**2
              CDSMALI    = 0.25-0.5*CVMALI**2
              CDSFEMI    = 0.25-0.5*CVFEMI**2
              CDSBATCHI  = 0.125-0.5*CVBATCHI**2
          ENDIF

*         Relative attrition rates of uninfected and infected insects
          RATRL1  = -1*LOG(FSURVL1)*DVRL1
          RATRL2  = -1*LOG(FSURVL2)*DVRL2
          RATRL3  = -1*LOG(FSURVL3)*DVRL3
          RATRL4  = -1*LOG(FSURVL4)*DVRL4
          RATRL5  = -1*LOG(FSURVL5)*DVRL5
          RATRPUP = -1*LOG(FSURVPUP)*DVRPUP
          RATRFEM = -1*LOG(FSURVFEM)*DVRFEM
          RATRMAL = -1*LOG(FSURVMAL)*DVRMAL
      ENDIF

*     Calculation of SEDEV rate variables
      IF (ITASK.EQ.3) THEN
*         Calculation of rate variables of uninfected insects.
*         Development of adult moths and pupae is simulated at the greenhouse
*         level
          CALL IBOXCAR(1,ITASK,DELT,IBATIN,4,DVRBATCH,
     $                 CVBATCH,CDSBATCH,0.,
     $                 IBATCH,ITOTBAT,IBATOUT,IATRBAT)
          CALL IBOXCAR(2,ITASK,DELT,IMLTMAL,2,DVRMAL,
     $                 CVMAL,CDSMAL,RATRMAL,
     $                 IMAL,ITOTMAL,IDRMAL,IATRMAL)
          CALL IBOXCAR(3,ITASK,DELT,IMLTFEM,2,DVRFEM,
     $                 CVFEM,CDSFEM,RATRFEM,
     $                 IFEM,ITOTFEM,IDRFEM,IATRFEM)
          CALL IBOXPUP(1,ITASK,DELT,IMOULTP,4,DVRPUP,CVPUP,CDSPUP,
     $                 SEXRAT,RATRPUP,
     $                 IPUP,ITOTPUP,IMLTMAL,IMLTFEM,IATRPUP)

*         Development of eggs and larvae is simulated per patch
          DO 60 IP = IPATLOW,IPATCUM
              CALL IBOXPATL5(1,ITASK,DELT,IMLTL5S(IP),4,DVRL5,CVL5,
     $                       CDSL5S(IP),IP,RATRL5,RIRL5HT(IP),
     $                       RIRL5SS(IP),FINFL5S,
     $                       IL5S,IPATL5S(IP),IMLTPUPS(IP),IATRL5S(IP),
     $                       IRL5SHTS(IP),IRL5SHTD(IP),IRL5SSS(IP),
     $                       IRL5SSD(IP),IRL5SHTSAR,IRL5SSSAR)
              CALL IBOXPATL5(2,ITASK,DELT,IMLTL5R(IP),4,DVRL5,CVL5,
     $                       CDSL5R(IP),IP,RATRL5,RIRL5HT(IP),
     $                       RIRL5RS(IP),FINFL5S,
     $                       IL5R,IPATL5R(IP),IMLTPUPR(IP),IATRL5R(IP),
     $                       IRL5RHTS(IP),IRL5RHTD(IP),IRL5RSS(IP),
     $                       IRL5RSD(IP),IRL5RHTSAR,IRL5RSSAR)
              CALL IBOXPAT(1,ITASK,DELT,IMLTL4S(IP),4,DVRL4,CVL4,
     $                     CDSL4S(IP),IP,RATRL4,RIRL4HT(IP),
     $                     RIRL4SS(IP),
     $                     IL4S,IPATL4S(IP),IMLTL5S(IP),IATRL4S(IP),
     $                     IRL4SHT(IP),IRL4SS(IP))
              CALL IBOXPAT(2,ITASK,DELT,IMLTL4R(IP),4,DVRL4,CVL4,
     $                     CDSL4R(IP),IP,RATRL4,RIRL4HT(IP),
     $                     RIRL4RS(IP),
     $                     IL4R,IPATL4R(IP),IMLTL5R(IP),IATRL4R(IP),
     $                     IRL4RHT(IP),IRL4RS(IP))
              CALL IBOXPAT(3,ITASK,DELT,IMLTL3S(IP),4,DVRL3,CVL3,
     $                     CDSL3S(IP),IP,RATRL3,RIRL3HT(IP),
     $                     RIRL3SS(IP),
     $                     IL3S,IPATL3S(IP),IMLTL4S(IP),IATRL3S(IP),
     $                     IRL3SHT(IP),IRL3SS(IP))
              CALL IBOXPAT(4,ITASK,DELT,IMLTL3R(IP),4,DVRL3,CVL3,
     $                     CDSL3R(IP),IP,RATRL3,RIRL3HT(IP),
     $                     RIRL3RS(IP),
     $                     IL3R,IPATL3R(IP),IMLTL4R(IP),IATRL3R(IP),
     $                     IRL3RHT(IP),IRL3RS(IP))
              CALL IBOXPATL2(ITASK,DELT,IMLTL2(IP),4,DVRL2,CVL2,
     $                       CDSL2(IP),IP,RATRL2,RIRL2HT(IP),
     $                       RIRL2S(IP),
     $                       IL2,IPATL2(IP),IMLTL3S(IP),IMLTL3R(IP),
     $                       IATRL2(IP),IRL2HT(IP),IRL2S(IP))
              CALL IBOXPAT(5,ITASK,DELT,IHATCH(IP),4,DVRL1,CVL1,
     $                     CDSL1(IP),IP,RATRL1,RIRL1HT(IP),
     $                     RIRL1S(IP),
     $                     IL1,IPATL1(IP),IMLTL2(IP),IATRL1(IP),
     $                     IRL1HT(IP),IRL1S(IP))
              CALL IBOXPAT(6,ITASK,DELT,0,4,DVREGG,CVEGG,
     $                     CDSEGG(IP),IP,0.,0.,
     $                     0.,
     $                     IEGG,IPATEGG(IP),IHATCH(IP),IATREGG(IP),
     $                     0,0)
60        CONTINUE

*         Calculation of rate variables of infected insects.
*         Development of adult moths and pupae is simulated at the greenhouse
*         level
```

```
          CALL IBOXCAR(4,ITASK,DELT,IBATINI,4,DVRBATCHI,
     $                 CVBATCHI,CDSBATCHI,0.,
     $                 IBATCHI,ITOTBATI,IBATOUTI,IATRBATI)
          CALL IBOXCAR(5,ITASK,DELT,IMLTMALI,2,DVRMALI,
     $                 CVMALI,CDSMALI,RATRMAL,
     $                 IMALI,ITOTMALI,IDRMALI,IATRMALI)
          CALL IBOXCAR(6,ITASK,DELT,IMLTFEMI,2,DVRFEMI,
     $                 CVFEMI,CDSFEMI,RATRFEM,
     $                 IFEMI,ITOTFEMI,IDRFEMI,IATRFEMI)
          CALL IBOXPUP(2,ITASK,DELT,IMOULTPI,4,DVRPUPI,CVPUPI,CDSPUPI,
     $                 SEXRAT,RATRPUP,
     $                 IPUPI,ITOTPUPI,IMLTMALI,IMLTFEMI,IATRPUPI)

*         Development of contaminated eggs and larvae is simulated per patch
          DO 70 IP = IPATLOW,IPATCUM
              CALL IBOXPTIS(ITASK,DELT,IRL5SHTSAR,IRL5RHTSAR,
     $                     IRL5SSSAR,IRL5RSSAR,
     $                     4,DVRL5IS,CVL5IS,CDSL5IS(IP),IP,RATRL5,
     $                     IL5IS,ITIL5S(IP),IMLTPUPI(IP),IATRL5IS(IP))
              CALL IBOXPTI(1,ITASK,DELT,IRL5SHTD(IP)+IRL5RHTD(IP)+
     $                     IRL5SSD(IP)+IRL5RSD(IP),
     $                     4,DVRL5ID,CVL5ID,CDSL5ID(IP),IP,RATRL5,
     $                     IL5ID,ITIL5D(IP),IDRL5ID(IP))
              CALL IBOXPTI(2,ITASK,DELT,IRL4SHT(IP)+IRL4RHT(IP)+
     $                     IRL4SS(IP)+IRL4RS(IP),
     $                     4,DVRL4I,CVL4I,CDSL4I(IP),IP,RATRL4,
     $                     IL4I,IPATL4I(IP),IDRL4I(IP))
              CALL IBOXPTI(3,ITASK,DELT,IRL3SHT(IP)+IRL3RHT(IP)+
     $                     IRL3SS(IP)+IRL3RS(IP),
     $                     4,DVRL3I,CVL3I,CDSL3I(IP),IP,RATRL3,
     $                     IL3I,IPATL3I(IP),IDRL3I(IP))
              CALL IBOXPTI(4,ITASK,DELT,IRL2HT(IP)+IRL2S(IP),
     $                     4,DVRL2I,CVL2I,CDSL2I(IP),IP,RATRL2,
     $                     IL2I,IPATL2I(IP),IDRL2I(IP))
              CALL IBOXPTI(5,ITASK,DELT,IHATCHI(IP)+IRL1HT(IP)
     $                     +IRL1S(IP),
     $                     4,DVRL1I,CVL1I,CDSL1I(IP),IP,RATRL1,
     $                     IL1I,IPATL1I(IP),IDRL1I(IP))
              CALL IBOXPTI(6,ITASK,DELT,0,
     $                     4,DVREGGI,CVEGGI,CDSEGGI(IP),IP,0.,
     $                     IEGGI,IPATEGGI(IP),IHATCHI(IP))
70        CONTINUE

*         Auxiliary variables for transfer of L5 to pupea from patch level
*         to greenhouse level
          IMOULTP = 0
          IMOULTPI = 0

*         Newly moulted pupea are simulated at the greenhouse level
          DO 80 IP = IPATLOW,IPATCUM
```

```
              IMOULTP = IMOULTP + IMLTPUPS(IP) + IMLTPUPR(IP)
              IMOULTPI = IMOULTPI + IMLTPUPI(IP)
80        CONTINUE

*         Mating of new emerged female moths and initialization of new
*         egg batches in female oviduct.
          CALL MATING(IMLTFEM,IMLTFEMI,IMLTMAL,IMLTMALI,
     $                ITOTMAL,ITOTMALI,IBATNR,
     $                IBATIN,IBATINI)
      ENDIF

*     Output of SEDEV variables
      IF (ITASK.EQ.4) THEN
*         Calculation of shift for uninfected insects.
*         Development of adult moths and pupae is simulated at the greenhouse
*         level
          CALL IBOXCAR(1,ITASK,DELT,IBATIN,4,DVRBATCH,
     $                 CVBATCH,CDSBATCH,0.,
     $                 IBATCH,ITOTBAT,IBATOUT,IATRBAT)
          CALL IBOXCAR(2,ITASK,DELT,IMLTMAL,2,DVRMAL,
     $                 CVMAL,CDSMAL,RATRMAL,
     $                 IMAL,ITOTMAL,IDRMAL,IATRMAL)
          CALL IBOXCAR(3,ITASK,DELT,IMLTFEM,2,DVRFEM,
     $                 CVFEM,CDSFEM,RATRFEM,
     $                 IFEM,ITOTFEM,IDRFEM,IATRFEM)
          CALL IBOXPUP(1,ITASK,DELT,IMOULTP,4,DVRPUP,CVPUP,CDSPUP,
     $                 SEXRAT,RATRPUP,
     $                 IPUP,ITOTPUP,IMLTMAL,IMLTFEM,IATRPUP)

*         Development of eggs and larvae is simulated per patch
          DO 90 IP = IPATLOW,IPATCUM
              CALL IBOXPATL5(1,ITASK,DELT,IMLTL5S(IP),4,DVRL5,CVL5,
     $                       CDSL5S(IP),IP,RATRL5,RIRL5HT(IP),
     $                       RIRL5SS(IP),FINFL5S,
     $                       IL5S,IPATL5S(IP),IMLTPUPS(IP),IATRL5S(IP),
     $                       IRL5SHTS(IP),IRL5SHTD(IP),IRL5SSS(IP),
     $                       IRL5SSD(IP),IRL5SHTSAR,IRL5SSSAR)
              CALL IBOXPATL5(2,ITASK,DELT,IMLTL5R(IP),4,DVRL5,CVL5,
     $                       CDSL5R(IP),IP,RATRL5,RIRL5HT(IP),
     $                       RIRL5RS(IP),FINFL5S,
     $                       IL5R,IPATL5R(IP),IMLTPUPR(IP),IATRL5R(IP),
     $                       IRL5RHTS(IP),IRL5RHTD(IP),IRL5RSS(IP),
     $                       IRL5RSD(IP),IRL5RHTSAR,IRL5RSSAR)
              CALL IBOXPAT(1,ITASK,DELT,IMLTL4S(IP),4,DVRL4,CVL4,
     $                     CDSL4S(IP),IP,RATRL4,RIRL4HT(IP),
     $                     RIRL4SS(IP),
     $                     IL4S,IPATL4S(IP),IMLTL5S(IP),IATRL4S(IP),
     $                     IRL4SHT(IP),IRL4SS(IP))
              CALL IBOXPAT(2,ITASK,DELT,IMLTL4R(IP),4,DVRL4,CVL4,
     $                     CDSL4R(IP),IP,RATRL4,RIRL4HT(IP),
     $                    RIRL4RS(IP),
     $                    IL4R,IPATL4R(IP),IMLTL5R(IP),IATRL4R(IP),
     $                    IRL4RHT(IP),IRL4RS(IP))
           CALL IBOXPAT(3,ITASK,DELT,IMLTL3S(IP),4,DVRL3,CVL3,
     $                    CDSL3S(IP),IP,RATRL3,RIRL3HT(IP),
     $                    RIRL3SS(IP),
     $                    IL3S,IPATL3S(IP),IMLTL4S(IP),IATRL3S(IP),
     $                    IRL3SHT(IP),IRL3SS(IP))
           CALL IBOXPAT(4,ITASK,DELT,IMLTL3R(IP),4,DVRL3,CVL3,
     $                    CDSL3R(IP),IP,RATRL3,RIRL3HT(IP),
     $                    RIRL3RS(IP),
     $                    IL3R,IPATL3R(IP),IMLTL4R(IP),IATRL3R(IP),
     $                    IRL3RHT(IP),IRL3RS(IP))
           CALL IBOXPATL2(ITASK,DELT,IMLTL2(IP),4,DVRL2,CVL2,
     $                      CDSL2(IP),IP,RATRL2,RIRL2HT(IP),
     $                      RIRL2S(IP),
     $                      IL2,IPATL2(IP),IMLTL3S(IP),IMLTL3R(IP),
     $                      IATRL2(IP),IRL2HT(IP),IRL2S(IP))
           CALL IBOXPAT(5,ITASK,DELT,IHATCH(IP),4,DVRL1,CVL1,
     $                    CDSL1(IP),IP,RATRL1,RIRL1HT(IP),
     $                    RIRL1S(IP),
     $                    IL1,IPATL1(IP),IMLTL2(IP),IATRL1(IP),
     $                    IRL1HT(IP),IRL1S(IP))
           CALL IBOXPAT(6,ITASK,DELT,0,4,DVREGG,CVEGG,
     $                    CDSEGG(IP),IP,0.,0.,
     $                    0.,
     $                    IEGG,IPATEGG(IP),IHATCH(IP),IATREGG(IP),
     $                    0,0)
90       CONTINUE

*        Calculation of shift for infected insects.
*        Development of adult moths and pupae is simulated at the greenhouse
*        level
         CALL IBOXCAR(4,ITASK,DELT,IBATINI,4,DVRBATCHI,
     $                 CVBATCHI,CDSBATCHI,0.,
     $                 IBATCHI,ITOTBATI,IBATOUTI,IATRBATI)
         CALL IBOXCAR(5,ITASK,DELT,IMLTMALI,2,DVRMALI,
     $                 CVMALI,CDSMALI,RATRMAL,
     $                 IMALI,ITOTMALI,IDRMALI,IATRMALI)
         CALL IBOXCAR(6,ITASK,DELT,IMLTFEMI,2,DVRFEMI,
     $                 CVFEMI,CDSFEMI,RATRFEM,
     $                 IFEMI,ITOTFEMI,IDRFEMI,IATRFEMI)
         CALL IBOXPUP(2,ITASK,DELT,IMOULTPI,4,DVRPUPI,CVPUPI,CDSPUPI,
     $                 SEXRAT,RATRPUP,
     $                 IPUPI,ITOTPUPI,IMLTMALI,IMLTFEMI,IATRPUPI)

*        Development of contaminated eggs and larvae is simulated per patch
         DO 100 IP = IPATLOW,IPATCUM
           CALL IBOXPTIS(ITASK,DELT,IRL5SHTSAR,IRL5RHTSAR,
     $                    IRL5SSSAR,IRL5RSSAR,
     $                  4,DVRL5IS,CVL5IS,CDSL5IS(IP),IP,RATRL5,
     $                  IL5IS,ITIL5S(IP),IMLTPUPI(IP),IATRL5IS(IP))
             CALL IBOXPTI(1,ITASK,DELT,IRL5SHTD(IP)+IRL5RHTD(IP)+
     $                  IRL5SSD(IP)+IRL5RSD(IP),
     $                  4,DVRL5ID,CVL5ID,CDSL5ID(IP),IP,RATRL5,
     $                  IL5ID,ITIL5D(IP),IDRL5ID(IP))
             CALL IBOXPTI(2,ITASK,DELT,IRL4SHT(IP)+IRL4RHT(IP)+
     $                  IRL4SS(IP)+IRL4RS(IP),
     $                  4,DVRL4I,CVL4I,CDSL4I(IP),IP,RATRL4,
     $                  IL4I,IPATL4I(IP),IDRL4I(IP))
             CALL IBOXPTI(3,ITASK,DELT,IRL3SHT(IP)+IRL3RHT(IP)+
     $                  IRL3SS(IP)+IRL3RS(IP),
     $                  4,DVRL3I,CVL3I,CDSL3I(IP),IP,RATRL3,
     $                  IL3I,IPATL3I(IP),IDRL3I(IP))
             CALL IBOXPTI(4,ITASK,DELT,IRL2HT(IP)+IRL2S(IP),
     $                  4,DVRL2I,CVL2I,CDSL2I(IP),IP,RATRL2,
     $                  IL2I,IPATL2I(IP),IDRL2I(IP))
             CALL IBOXPTI(5,ITASK,DELT,IHATCHI(IP)+IRL1HT(IP)
     $                  +IRL1S(IP),
     $                  4,DVRL1I,CVL1I,CDSL1I(IP),IP,RATRL1,
     $                  IL1I,IPATL1I(IP),IDRL1I(IP))
             CALL IBOXPTI(6,ITASK,DELT,0,
     $                  4,DVREGGI,CVEGGI,CDSEGGI(IP),IP,0.,
     $                  IEGGI,IPATEGGI(IP),IHATCHI(IP))
100     CONTINUE

*        Calculation of insect population totals
         CALL ACCOUNT(IPATCUM,IPATEGG,IPATL1,IPATL2,IPATL3S,IPATL3R,
     $                IPATL4S,IPATL4R,IPATL5S,IPATL5R,IPATEGGI,IPATL1I,
     $                IPATL2I,IPATL3I,IPATL4I,ITIL5S,ITIL5D,ITOTPUP,
     $                ITOTMAL,ITOTFEM,ITOTPUPI,ITOTMALI,ITOTFEMI,
     $                   IPATLOW)
         CALL OUTDAT(2,20,'IPATCH',REAL(IPATCUM-IPATLOW+1))
      ENDIF

*     Calculation of SEDEV state variables
      IF (ITASK.EQ.5) THEN
*        Calculation of state variables of uninfected insects.
*        Development of adult moths and pupae is simulated at the greenhouse
*        level

*        Inflight of fertilized female moths from outside the greenhouse
*        that deposit egg batches in the crop
         DO 110 IN = 1,50
            IF (ABS(TBATINFL(IN)-TIME).LE.(0.5*DELT)) THEN
               TBATINFL(IN) = 0.
               IBATCH(0) = IBATCH(0)+INBATINFL
            ENDIF
110      CONTINUE
```

```
          CALL IBOXCAR(1,ITASK,DELT,IBATIN,4,DVRBATCH,
     $                 CVBATCH,CDSBATCH,0.,
     $                 IBATCH,ITOTBAT,IBATOUT,IATRBAT)
          CALL IBOXCAR(2,ITASK,DELT,IMLTMAL,2,DVRMAL,
     $                 CVMAL,CDSMAL,RATRMAL,
     $                 IMAL,ITOTMAL,IDRMAL,IATRMAL)
          CALL IBOXCAR(3,ITASK,DELT,IMLTFEM,2,DVRFEM,
     $                 CVFEM,CDSFEM,RATRFEM,
     $                 IFEM,ITOTFEM,IDRFEM,IATRFEM)
          CALL IBOXPUP(1,ITASK,DELT,IMOULTP,4,DVRPUP,CVPUP,CDSPUP,
     $                 SEXRAT,RATRPUP,
     $                 IPUP,ITOTPUP,IMLTMAL,IMLTFEM,IATRPUP)

*         Development of eggs and larvae is simulated per patch
          DO 120 IP = IPATLOW,IPATCUM
             CALL IBOXPATL5(1,ITASK,DELT,IMLTL5S(IP),4,DVRL5,CVL5,
     $                       CDSL5S(IP),IP,RATRL5,RIRL5HT(IP),
     $                       RIRL5SS(IP),FINFL5S,
     $                       IL5S,IPATL5S(IP),IMLTPUPS(IP),IATRL5S(IP),
     $                       IRL5SHTS(IP),IRL5SHTD(IP),IRL5SSS(IP),
     $                       IRL5SSD(IP),IRL5SHTSAR,IRL5SSSAR)
             CALL IBOXPATL5(2,ITASK,DELT,IMLTL5R(IP),4,DVRL5,CVL5,
     $                       CDSL5R(IP),IP,RATRL5,RIRL5HT(IP),
     $                       RIRL5RS(IP),FINFL5S,
     $                       IL5R,IPATL5R(IP),IMLTPUPR(IP),IATRL5R(IP),
     $                       IRL5RHTS(IP),IRL5RHTD(IP),IRL5RSS(IP),
     $                       IRL5RSD(IP),IRL5RHTSAR,IRL5RSSAR)
             CALL IBOXPAT(1,ITASK,DELT,IMLTL4S(IP),4,DVRL4,CVL4,
     $                    CDSL4S(IP),IP,RATRL4,RIRL4HT(IP),
     $                    RIRL4SS(IP),
     $                    IL4S,IPATL4S(IP),IMLTL5S(IP),IATRL4S(IP),
     $                    IRL4SHT(IP),IRL4SS(IP))
             CALL IBOXPAT(2,ITASK,DELT,IMLTL4R(IP),4,DVRL4,CVL4,
     $                    CDSL4R(IP),IP,RATRL4,RIRL4HT(IP),
     $                    RIRL4RS(IP),
     $                    IL4R,IPATL4R(IP),IMLTL5R(IP),IATRL4R(IP),
     $                    IRL4RHT(IP),IRL4RS(IP))
             CALL IBOXPAT(3,ITASK,DELT,IMLTL3S(IP),4,DVRL3,CVL3,
     $                    CDSL3S(IP),IP,RATRL3,RIRL3HT(IP),
     $                    RIRL3SS(IP),
     $                    IL3S,IPATL3S(IP),IMLTL4S(IP),IATRL3S(IP),
     $                    IRL3SHT(IP),IRL3SS(IP))
             CALL IBOXPAT(4,ITASK,DELT,IMLTL3R(IP),4,DVRL3,CVL3,
     $                    CDSL3R(IP),IP,RATRL3,RIRL3HT(IP),
     $                    RIRL3RS(IP),
     $                    IL3R,IPATL3R(IP),IMLTL4R(IP),IATRL3R(IP),
     $                    IRL3RHT(IP),IRL3RS(IP))
             CALL IBOXPATL2(ITASK,DELT,IMLTL2(IP),4,DVRL2,CVL2,
     $                      CDSL2(IP),IP,RATRL2,RIRL2HT(IP),
     $                      RIRL2S(IP),
     $                       IL2,IPATL2(IP),IMLTL3S(IP),IMLTL3R(IP),
     $                       IATRL2(IP),IRL2HT(IP),IRL2S(IP))
              CALL IBOXPAT(5,ITASK,DELT,IHATCH(IP),4,DVRL1,CVL1,
     $                     CDSL1(IP),IP,RATRL1,RIRL1HT(IP),
     $                     RIRL1S(IP),
     $                     IL1,IPATL1(IP),IMLTL2(IP),IATRL1(IP),
     $                     IRL1HT(IP),IRL1S(IP))
              CALL IBOXPAT(6,ITASK,DELT,0,4,DVREGG,CVEGG,
     $                     CDSEGG(IP),IP,0.,0.,
     $                     0.,
     $                     IEGG,IPATEGG(IP),IHATCH(IP),IATREGG(IP),
     $                     0,0)
120       CONTINUE

*         Calculation of state variables of infected insects.
*         Development of adult moths and pupae is simulated at the greenhouse
*         level
          CALL IBOXCAR(4,ITASK,DELT,IBATINI,4,DVRBATCHI,
     $                 CVBATCHI,CDSBATCHI,0.,
     $                 IBATCHI,ITOTBATI,IBATOUTI,IATRBATI)
          CALL IBOXCAR(5,ITASK,DELT,IMLTMALI,2,DVRMALI,
     $                 CVMALI,CDSMALI,RATRMAL,
     $                 IMALI,ITOTMALI,IDRMALI,IATRMALI)
          CALL IBOXCAR(6,ITASK,DELT,IMLTFEMI,2,DVRFEMI,
     $                 CVFEMI,CDSFEMI,RATRFEM,
     $                 IFEMI,ITOTFEMI,IDRFEMI,IATRFEMI)
          CALL IBOXPUP(2,ITASK,DELT,IMOULTPI,4,DVRPUPI,CVPUPI,CDSPUPI,
     $                 SEXRAT,RATRPUP,
     $                 IPUPI,ITOTPUPI,IMLTMALI,IMLTFEMI,IATRPUPI)

*         Development of contaminated eggs and larvae is simulated per patch
          DO 130 IP = IPATLOW,IPATCUM
              CALL IBOXPTIS(ITASK,DELT,IRL5SHTSAR,IRL5RHTSAR,
     $                     IRL5SSSAR,IRL5RSSAR,
     $                     4,DVRL5IS,CVL5IS,CDSL5IS(IP),IP,RATRL5,
     $                     IL5IS,ITIL5S(IP),IMLTPUPI(IP),IATRL5IS(IP))
              CALL IBOXPTI(1,ITASK,DELT,IRL5SHTD(IP)+IRL5RHTD(IP)+
     $                     IRL5SSD(IP)+IRL5RSD(IP),
     $                     4,DVRL5ID,CVL5ID,CDSL5ID(IP),IP,RATRL5,
     $                     IL5ID,ITIL5D(IP),IDRL5ID(IP))
              CALL IBOXPTI(2,ITASK,DELT,IRL4SHT(IP)+IRL4RHT(IP)+
     $                     IRL4SS(IP)+IRL4RS(IP),
     $                     4,DVRL4I,CVL4I,CDSL4I(IP),IP,RATRL4,
     $                     IL4I,IPATL4I(IP),IDRL4I(IP))
              CALL IBOXPTI(3,ITASK,DELT,IRL3SHT(IP)+IRL3RHT(IP)+
     $                     IRL3SS(IP)+IRL3RS(IP),
     $                     4,DVRL3I,CVL3I,CDSL3I(IP),IP,RATRL3,
     $                     IL3I,IPATL3I(IP),IDRL3I(IP))
              CALL IBOXPTI(4,ITASK,DELT,IRL2HT(IP)+IRL2S(IP),
     $                     4,DVRL2I,CVL2I,CDSL2I(IP),IP,RATRL2,
     $                     IL2I,IPATL2I(IP),IDRL2I(IP))
              CALL IBOXPTI(5,ITASK,DELT,IHATCHI(IP)+IRL1HT(IP)
     $                     +IRL1S(IP),
     $                     4,DVRL1I,CVL1I,CDSL1I(IP),IP,RATRL1,
     $                     IL1I,IPATL1I(IP),IDRL1I(IP))
              CALL IBOXPTI(6,ITASK,DELT,0,
     $                     4,DVREGGI,CVEGGI,CDSEGGI(IP),IP,0.,
     $                     IEGGI,IPATEGGI(IP),IHATCHI(IP))
130       CONTINUE
      ENDIF

*     Calculation of shift for SEDEV variables
      IF (ITASK.EQ.6) THEN
*         All larvae are removed from bed that is harvested
          IF (IHARV.GT.0) THEN
              DO 150 IP = IPATLOW,IPATCUM
                  IF (IPATDAT(1,IP).EQ.IHARV) THEN
                       DO 140 IR = 0,4
                            IEGG(IP,IR) = 0
                            IL1(IP,IR) = 0
                            IL2(IP,IR) = 0
                            IL3S(IP,IR) = 0
                            IL3R(IP,IR) = 0
                            IL4S(IP,IR) = 0
                            IL4R(IP,IR) = 0
                            IL5S(IP,IR) = 0
                            IL5R(IP,IR) = 0
                            IEGGI(IP,IR) = 0
                            IL1I(IP,IR) = 0
                            IL2I(IP,IR) = 0
                            IL3I(IP,IR) = 0
                            IL4I(IP,IR) = 0
                            IL5ID(IP,IR) = 0
                            IL5IS(IP,IR) = 0
140                   CONTINUE
                  ENDIF
150           CONTINUE
          ENDIF

*         patch integral is raised with number of new patches
          IPATCUM = IPATCUM + IBATOUT + IBATOUTI
*         The program is terminated if actual number of patches exceeds
*         maximal number of patches
          IF(IPATCUM.GE.IPMAX) STOP'Maximum number of patches reached!'

*         Determination of egg batch size and fraction of infected
*         eggs in contaminated egg batch.
          IF (IBATOUT.GT.0.OR.IBATOUTI.GT.0) THEN
              CALL DEPOSIT(IBATOUT,IBATOUTI,FX,X,ICBATSIZE,
     $                     IPATCUM,FCBATS,FCBATL,FCEGGS,FCEGGL,ISTOCH,
     $                         IEGG,IEGGI)
          ENDIF

*         Calculation of shift for uninfected insects.
*         Development of adult moths and pupae is simulated at the greenhouse
*         level
          CALL IBOXCAR(1,ITASK,DELT,IBATIN,4,DVRBATCH,
     $                 CVBATCH,CDSBATCH,0.,
     $                 IBATCH,ITOTBAT,IBATOUT,IATRBAT)
          CALL IBOXCAR(2,ITASK,DELT,IMLTMAL,2,DVRMAL,
     $                 CVMAL,CDSMAL,RATRMAL,
     $                 IMAL,ITOTMAL,IDRMAL,IATRMAL)
          CALL IBOXCAR(3,ITASK,DELT,IMLTFEM,2,DVRFEM,
     $                 CVFEM,CDSFEM,RATRFEM,
     $                 IFEM,ITOTFEM,IDRFEM,IATRFEM)
          CALL IBOXPUP(1,ITASK,DELT,IMOULTP,4,DVRPUP,CVPUP,CDSPUP,
     $                 SEXRAT,RATRPUP,
     $                 IPUP,ITOTPUP,IMLTMAL,IMLTFEM,IATRPUP)

*         Development of eggs and larvae is simulated per patch
          DO 160 IP = IPATLOW,IPATCUM
              CALL IBOXPATL5(1,ITASK,DELT,IMLTL5S(IP),4,DVRL5,CVL5,
     $                       CDSL5S(IP),IP,RATRL5,RIRL5HT(IP),
     $                       RIRL5SS(IP),FINFL5S,
     $                       IL5S,IPATL5S(IP),IMLTPUPS(IP),IATRL5S(IP),
     $                       IRL5SHTS(IP),IRL5SHTD(IP),IRL5SSS(IP),
     $                       IRL5SSD(IP),IRL5SHTSAR,IRL5SSSAR)
              CALL IBOXPATL5(2,ITASK,DELT,IMLTL5R(IP),4,DVRL5,CVL5,
     $                       CDSL5R(IP),IP,RATRL5,RIRL5HT(IP),
     $                       RIRL5RS(IP),FINFL5S,
     $                       IL5R,IPATL5R(IP),IMLTPUPR(IP),IATRL5R(IP),
     $                       IRL5RHTS(IP),IRL5RHTD(IP),IRL5RSS(IP),
     $                       IRL5RSD(IP),IRL5RHTSAR,IRL5RSSAR)
              CALL IBOXPAT(1,ITASK,DELT,IMLTL4S(IP),4,DVRL4,CVL4,
     $                     CDSL4S(IP),IP,RATRL4,RIRL4HT(IP),
     $                     RIRL4SS(IP),
     $                     IL4S,IPATL4S(IP),IMLTL5S(IP),IATRL4S(IP),
     $                     IRL4SHT(IP),IRL4SS(IP))
              CALL IBOXPAT(2,ITASK,DELT,IMLTL4R(IP),4,DVRL4,CVL4,
     $                     CDSL4R(IP),IP,RATRL4,RIRL4HT(IP),
     $                     RIRL4RS(IP),
     $                     IL4R,IPATL4R(IP),IMLTL5R(IP),IATRL4R(IP),
     $                     IRL4RHT(IP),IRL4RS(IP))
              CALL IBOXPAT(3,ITASK,DELT,IMLTL3S(IP),4,DVRL3,CVL3,
     $                     CDSL3S(IP),IP,RATRL3,RIRL3HT(IP),
     $                     RIRL3SS(IP),
     $                     IL3S,IPATL3S(IP),IMLTL4S(IP),IATRL3S(IP),
     $                     IRL3SHT(IP),IRL3SS(IP))
              CALL IBOXPAT(4,ITASK,DELT,IMLTL3R(IP),4,DVRL3,CVL3,
     $                     CDSL3R(IP),IP,RATRL3,RIRL3HT(IP),
     $                    RIRL3RS(IP),
     $                    IL3R,IPATL3R(IP),IMLTL4R(IP),IATRL3R(IP),
     $                    IRL3RHT(IP),IRL3RS(IP))
            CALL IBOXPATL2(ITASK,DELT,IMLTL2(IP),4,DVRL2,CVL2,
     $                     CDSL2(IP),IP,RATRL2,RIRL2HT(IP),
     $                     RIRL2S(IP),
     $                     IL2,IPATL2(IP),IMLTL3S(IP),IMLTL3R(IP),
     $                     IATRL2(IP),IRL2HT(IP),IRL2S(IP))
            CALL IBOXPAT(5,ITASK,DELT,IHATCH(IP),4,DVRL1,CVL1,
     $                   CDSL1(IP),IP,RATRL1,RIRL1HT(IP),
     $                   RIRL1S(IP),
     $                   IL1,IPATL1(IP),IMLTL2(IP),IATRL1(IP),
     $                   IRL1HT(IP),IRL1S(IP))
            CALL IBOXPAT(6,ITASK,DELT,0,4,DVREGG,CVEGG,
     $                   CDSEGG(IP),IP,0.,0.,
     $                   0.,
     $                   IEGG,IPATEGG(IP),IHATCH(IP),IATREGG(IP),
     $                   0,0)
160       CONTINUE

*         Calculation of shift for infected insects.
*         Development of adult moths and pupae is simulated at the greenhouse
*         level
          CALL IBOXCAR(4,ITASK,DELT,IBATINI,4,DVRBATCHI,
     $                 CVBATCHI,CDSBATCHI,0.,
     $                 IBATCHI,ITOTBATI,IBATOUTI,IATRBATI)
          CALL IBOXCAR(5,ITASK,DELT,IMLTMALI,2,DVRMALI,
     $                 CVMALI,CDSMALI,RATRMAL,
     $                 IMALI,ITOTMALI,IDRMALI,IATRMALI)
          CALL IBOXCAR(6,ITASK,DELT,IMLTFEMI,2,DVRFEMI,
     $                 CVFEMI,CDSFEMI,RATRFEM,
     $                 IFEMI,ITOTFEMI,IDRFEMI,IATRFEMI)
          CALL IBOXPUP(2,ITASK,DELT,IMOULTPI,4,DVRPUPI,CVPUPI,CDSPUPI,
     $                 SEXRAT,RATRPUP,
     $                 IPUPI,ITOTPUPI,IMLTMALI,IMLTFEMI,IATRPUPI)

*         Development of contaminated eggs and larvae is simulated per patch
          DO 170 IP = IPATLOW,IPATCUM
            CALL IBOXPTIS(ITASK,DELT,IRL5SHTSAR,IRL5RHTSAR,
     $                    IRL5SSSAR,IRL5RSSAR,
     $                    4,DVRL5IS,CVL5IS,CDSL5IS(IP),IP,RATRL5,
     $                    IL5IS,ITIL5S(IP),IMLTPUPI(IP),IATRL5IS(IP))
            CALL IBOXPTI(1,ITASK,DELT,IRL5SHTD(IP)+IRL5RHTD(IP)+
     $                    IRL5SSD(IP)+IRL5RSD(IP),
     $                    4,DVRL5ID,CVL5ID,CDSL5ID(IP),IP,RATRL5,
     $                    IL5ID,ITIL5D(IP),IDRL5ID(IP))
            CALL IBOXPTI(2,ITASK,DELT,IRL4SHT(IP)+IRL4RHT(IP)+
     $                    IRL4SS(IP)+IRL4RS(IP),
     $                    4,DVRL4I,CVL4I,CDSL4I(IP),IP,RATRL4,
     $                    IL4I,IPATL4I(IP),IDRL4I(IP))
```

```
            CALL IBOXPTI(3,ITASK,DELT,IRL3SHT(IP)+IRL3RHT(IP)+
     $                   IRL3SS(IP)+IRL3RS(IP),
     $                   4,DVRL3I,CVL3I,CDSL3I(IP),IP,RATRL3,
     $                   IL3I,IPATL3I(IP),IDRL3I(IP))
            CALL IBOXPTI(4,ITASK,DELT,IRL2HT(IP)+IRL2S(IP),
     $                   4,DVRL2I,CVL2I,CDSL2I(IP),IP,RATRL2,
     $                   IL2I,IPATL2I(IP),IDRL2I(IP))
            CALL IBOXPTI(5,ITASK,DELT,IHATCHI(IP)+IRL1HT(IP)
     $                   +IRL1S(IP),
     $                   4,DVRL1I,CVL1I,CDSL1I(IP),IP,RATRL1,
     $                   IL1I,IPATL1I(IP),IDRL1I(IP))
            CALL IBOXPTI(6,ITASK,DELT,0,
     $                   4,DVREGGI,CVEGGI,CDSEGGI(IP),IP,0.,
     $                   IEGGI,IPATEGGI(IP),IHATCHI(IP))
170        CONTINUE
      ENDIF

      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* SUBROUTINE INITIM                                                     *
*                                                                       *
* Purpose: Initialization of time variables.                            *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
* name   type meaning (unit)                                      class *
* ----   ---- ---------------                                     ----- *
* ITASK   I4  Task that subroutine should perform (-)               I   *
* TIME    R4  Time (d)                                              O   *
* DELT    R4  Time step (d)                                         O   *
* FINTIM  R4  Time when simulation is terminated (d)                O   *
*                                                                       *
* Fatal error checks: none                                              *
* warnings:           none                                              *
* Subroutines called:                                                   *
* from TTUTIL:        RDINIT,RDSREA                                     *
* File usage:         INCON.DAT                                         *
*-----------------------------------------------------------------------*
      SUBROUTINE INITIM(ITASK,
     $                     TIME,DELT,FINTIM)

*     Initialization of INITIM variables
      IF (ITASK.EQ.1) THEN
*        Read initial conditions
         CALL RDINIT (30,40,'INCON.DAT')
         CALL RDSREA ('DELT',DELT)
         CALL RDSREA ('FINTIM',FINTIM)
         TIME = 0.
      ENDIF

      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* SUBROUTINE IBOXPAT                                                    *
*                                                                       *
* Purpose: Simulation of development, infection and attrition of       *
*          uninfected insects in patches.                               *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                     class *
* ----   ---- ---------------                                    ----- *
* ICOUNT  I4  Counter for boxcar train number (-)                  I  *
* ITASK   I4  Task that subroutine should perform (-)              I  *
* DELT    R4  Time step (d)                                        I  *
* IAIN    I4  Number of individuals flowing in boxcar train (#)    I  *
* IN      I4  Number of boxcars per boxcar train (#)               I  *
* DVR     R4  Development rate (d-1)                               I  *
* CV      R4  Coefficient of variation (-)                         I  *
* CDS     R4  Cyclic development stage (-)                         I  *
* IP      I4  Pointer for patch number (-)                         I  *
* RATR    R4  Relative attrition rate (d-1)                        I  *
* RINFRHT R4  Relative infection rate by horizontal transmission      *
*             (d-1)                                                I  *
* RINFRS  R4  Relative infection rate by ingestion of sprayed         *
*             polyhedra (d-1)                                      I  *
* IA      I4  Number of individuals in boxcar (#)                  O  *
* IATOT   I4  Number of individuals in boxcar train (#)            O  *
* IOUTFL  I4  Outflow difference of individuals from final boxcar     *
*             per timestep (#)                                     O  *
* IATR    I4  Attrition difference of individuals in boxcar train     *
*             per timestep (#)                                     O  *
* INFRHT  I4  Infection difference by horizontal transmission of      *
*             larvae per boxcar train per timestep (#)             O  *
* INFRS   I4  Infection difference by ingestion of sprayed            *
*             polyhedra of individuals per boxcar train per           *
*             timestep (#)                                         O  *
*                                                                     *
* Fatal error checks: DVR*DELT > F/IN                                 *
* warnings:           IA < 0                                          *
* Subroutines or functions called:                                    *
* own:                IROUND,DRWMORT                                  *
* File usage:         none                                            *
*---------------------------------------------------------------------*
      SUBROUTINE IBOXPAT(ICOUNT,ITASK,DELT,IAIN,IN,DVR,CV,CDS,
     $                   IP,RATR,RINFRHT,RINFRS,
     $                       IA,IATOT,IOUTFL,IATR,INFRHT,INFRS)

      PARAMETER (IPMAX = 200)
      INTEGER   IN,IK,IP,IA,IOUTFL,IAIN,IRA,ICOUNT,ITASK,IATOT,
     $          IATR,INFRHT,IR,INTINFHT,INTATR,IATRRED,
     $          INFRREDHT,INFRREDS,INFRS
      REAL      F,OUTFL,RA,CDS,RINFRHT,RINFRS,REALINFHT,
     $          REALATR,REALINFS,CV
      DIMENSION IRA(IPMAX,0:4),IA(IPMAX,0:4),RA(IPMAX,0:4),
     $          IATRRED(6,IPMAX,0:4),INFRREDHT(6,IPMAX,0:4),
     $          INFRREDS(6,IPMAX,0:4),REALINFHT(0:4),
     $          REALINFS(0:4),REALATR(0:4)
      SAVE

*     Calculation of IBOXPAT rate variables
      IF (ITASK.EQ.3) THEN
*        Calculation of infection and attrition differences of larvae
         DO 10 IK = 0,IN
            REALINFHT(IK) = RINFRHT*DELT*IA(IP,IK)
            INTINFHT = IROUND(REALINFHT(IK))
            INFRREDHT(ICOUNT,IP,IK) = INTINFHT

            REALINFS(IK) = RINFRS*DELT*IA(IP,IK)
            INTINFS = IROUND(REALINFS(IK))
            INFRREDS(ICOUNT,IP,IK) = INTINFS

            REALATR(IK) = RATR*DELT*IA(IP,IK)
            INTATR = IROUND(REALATR(IK))
            IATRRED(ICOUNT,IP,IK) = INTATR
10       CONTINUE

*        Mortality in boxcar cannot be higher as boxcar contents
         DO 20 IK = 0,IN
           IF ((IATRRED(ICOUNT,IP,IK) + INFRREDHT(ICOUNT,IP,IK) +
     $       INFRREDS(ICOUNT,IP,IK)).GT.IA(IP,IK)) THEN

*             Drawing of process responsible for death of larvae
              CALL DRWMORT(IA(IP,IK),REALATR(IK),REALINFS(IK),
     $                     REALINFHT(IK),
     $                     IATRRED(ICOUNT,IP,IK),INFRREDS(ICOUNT,IP,IK),
     $                     INFRREDHT(ICOUNT,IP,IK))
           ENDIF
20       CONTINUE

*        Calculation of number of larvae that become infected or die
*        by attrition
         IATR = 0
         INFRHT = 0
         INFRS = 0
         DO 30 IK = 0,IN
            IATR = IATR + IATRRED(ICOUNT,IP,IK)
            INFRHT = INFRHT + INFRREDHT(ICOUNT,IP,IK)
            INFRS = INFRS + INFRREDS(ICOUNT,IP,IK)
30       CONTINUE

*        Calculation of outflow in final boxcar
         OUTFL = MIN( (DVR*IA(IP,IN)/(1./IN-CDS)),
     $            (IA(IP,IN)/DELT)) * DELT - IATRRED(ICOUNT,IP,IN) -
     $            INFRREDHT(ICOUNT,IP,IN) - INFRREDS(ICOUNT,IP,IN)
         IOUTFL = IROUND(OUTFL)

*        When all individuals in final boxcar become infected or die
*        by attrition, outflow is set to 0.
         IF (IOUTFL.LT.0) THEN
            IOUTFL = 0
         ENDIF
      ENDIF

*     Output of IBOXPAT variables
      IF (ITASK.EQ.4) THEN
*        Total number in boxcar
         IATOT = IA(IP,0)
         DO 70 IR = 1,IN
            IATOT = IATOT + IA(IP,IR)
70       CONTINUE
      ENDIF

*     Calculation of IBOXPAT state variables
      IF (ITASK.EQ.5) THEN
*        Cyclic development stage integral
         CDS = CDS + DELT*DVR

*        Calculation of new states in boxcars
         IA(IP,0) = IA(IP,0) + IAIN - INFRREDHT(ICOUNT,IP,0) -
     $              IATRRED(ICOUNT,IP,0) - INFRREDS(ICOUNT,IP,0)
         DO 40 IK = 1,IN-1
            IA(IP,IK) = IA(IP,IK) - INFRREDHT(ICOUNT,IP,IK) -
     $                  IATRRED(ICOUNT,IP,IK) - INFRREDS(ICOUNT,IP,IK)
40       CONTINUE

*        Calculation of new state in final boxcar. The contents of this boxcar
*        is reduced with the outflow that takes place in the timestep.
         IA(IP,IN) = IA(IP,IN) - IOUTFL - INFRREDHT(ICOUNT,IP,IN) -
     $               IATRRED(ICOUNT,IP,IN) - INFRREDS(ICOUNT,IP,IN)

*        Check for negative boxcar contents
         IF (IA(IP,0).LT.0.OR.IA(IP,1).LT.0.OR.IA(IP,2).LT.0.OR.
     $       IA(IP,3).LT.0.OR.IA(IP,4).LT.0.) THEN
            WRITE (*,50)
50          FORMAT('IBOXPAT content negative!')
         ENDIF
      ENDIF

*     Calculation of shift for IBOXPAT variables
      IF (ITASK.EQ.6) THEN
*        Calculation of F
         F = 1. - IN*CV*CV
```

```
*         Check on parameter values
          IF (DVR*DELT.GE.F/IN) THEN
              STOP'DVR*DELT.GE.F/IN!'
          ENDIF

*         The shift takes place immediatly after the new states have been
*         calculated. The contents of the different boxcars are shifted.
          IF (CDS.GE.F/IN) THEN
              RA(IP,IN) = F*IA(IP,IN-1)
              IRA(IP,IN) = IROUND(RA(IP,IN))
              IA(IP,IN) = IA(IP,IN) + IRA(IP,IN)
              DO 60 IR = IN-1,2,-1
                  RA(IP,IR) = F*IA(IP,IR-1)
                  IRA(IP,IR) = IROUND(RA(IP,IR))
                  IA(IP,IR) = IA(IP,IR) - IRA(IP,IR+1) + IRA(IP,IR)
60            CONTINUE

              IRA(IP,1) = IA(IP,0)
              IA(IP,1) = IA(IP,1) - IRA(IP,2) + IRA(IP,1)

*             The contents of the inflow boxcar is set to zero
              IA(IP,0) = 0

*             Cyclic development stage is updated after shift
              CDS = CDS - F/IN
          ENDIF
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE IBOXPATL2                                                 *
*                                                                      *
* Purpose: Simulation of development, infection and attrition of       *
*          uninfected L2 larvae in patches.                            *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* ITASK    I4  Task that subroutine should perform (-)              I  *
* DELT     R4  Time step (d)                                        I  *
* IAIN     I4  Number of individuals flowing in boxcar train (#)    I  *
* IN       I4  Number of boxcars per boxcar train (#)               I  *
* DVR      R4  Development rate (d-1)                               I  *
* CV       R4  Coefficient of variation (-)                         I  *
* CDS      R4  Cyclic development stage (-)                         I  *
* IP       I4  Pointer for patch number (-)                         I  *
* RATR     R4  Relative attrition rate (d-1)                        I  *
* RINFRHT R4   Relative infection rate by horizontal transmission      *
*              (d-1)                                                I  *
* RINFRS   R4  Relative infection rate by ingestion of sprayed         *
*              polyhedra (d-1)                                      I  *
* IA       I4  Number of individuals in boxcar (#)                  O  *
* IATOT    I4  Number of individuals in boxcar train (#)            O  *
* IOUTFLS I4   Outflow difference of susceptible individuals from      *
*              final boxcar per timestep (#)                        O  *
* IOUTFLR I4   Outflow difference of resistant individuals from        *
*              final boxcar per timestep (#)                        O  *
* IATR     I4  Attrition difference of individuals in boxcar train     *
*              per timestep (#)                                     O  *
* INFRHT   I4  Infection difference by horizontal transmission of      *
*              larvae per boxcar train per timestep (#)             O  *
* INFRS    I4  Infection difference by ingestion of sprayed            *
*              polyhedra of individuals per boxcar train per           *
*              timestep (#)                                         O  *
*                                                                      *
* Fatal error checks: DVR*DELT > F/IN                                  *
* warnings:           IA < 0                                           *
* Subroutines or functions called:                                     *
* own:                IROUND,DRWMORT                                   *
* File usage:         none                                             *
*----------------------------------------------------------------------*
      SUBROUTINE IBOXPATL2(ITASK,DELT,IAIN,IN,DVR,CV,CDS,
     $                     IP,RATR,RINFRHT,RINFRS,
     $                         IA,IATOT,IOUTFLS,IOUTFLR,IATR,INFRHT,
     $                         INFRS)

      PARAMETER (IPMAX = 200)
      INTEGER   IN,IK,IP,IA,IOUTFLS,IOUTFLR,IAIN,IRA,ITASK,IATOT,
     $          IATR,INFRHT,IR,INTINFHT,INTATR,IATRRED,
     $          INFRREDHT,INFRREDS,INFRS
      REAL      F,RA,CDS,RINFRHT,RINFRS,REALINFHT,
     $          REALATR,REALINFS,CV
      DIMENSION IRA(IPMAX,0:4),IA(IPMAX,0:4),RA(IPMAX,0:4),
     $          IATRRED(IPMAX,0:4),INFRREDHT(IPMAX,0:4),
     $          INFRREDS(IPMAX,0:4),REALINFHT(0:4),REALINFS(0:4),
     $          REALATR(0:4)
      SAVE

*     Calculation of IBOXPAT rate variables
      IF (ITASK.EQ.3) THEN
*         Calculation of infection and attrition differences of larvae
          DO 10 IK = 0,IN
             REALINFHT(IK) = RINFRHT*DELT*IA(IP,IK)
             INTINFHT = IROUND(REALINFHT(IK))
             INFRREDHT(IP,IK) = INTINFHT

             REALINFS(IK) = RINFRS*DELT*IA(IP,IK)
             INTINFS = IROUND(REALINFS(IK))
             INFRREDS(IP,IK) = INTINFS

             REALATR(IK) = RATR*DELT*IA(IP,IK)
             INTATR = IROUND(REALATR(IK))
             IATRRED(IP,IK) = INTATR
10        CONTINUE

*         Mortality in boxcar cannot be higher as boxcar contents
          DO 20 IK = 0,IN
             IF ((IATRRED(IP,IK) + INFRREDHT(IP,IK) +
     $          INFRREDS(IP,IK)).GT.IA(IP,IK)) THEN

*               Drawing of process responsible for death of larvae
                CALL DRWMORT(IA(IP,IK),REALATR(IK),REALINFS(IK),
     $                       REALINFHT(IK),
     $                       IATRRED(IP,IK),INFRREDS(IP,IK),
     $                       INFRREDHT(IP,IK))
             ENDIF

20        CONTINUE

*         Calculation of number of larvae that become infected or die
*         by attrition
          IATR = 0
          INFRHT = 0
          INFRS = 0
          DO 30 IK = 0,IN
             IATR = IATR + IATRRED(IP,IK)
             INFRHT = INFRHT + INFRREDHT(IP,IK)
             INFRS = INFRS + INFRREDS(IP,IK)
30        CONTINUE
```

```
*         Calculation of outflow in final boxcar
          OUTFL = MIN( (DVR*IA(IP,IN)/(1./IN-CDS)),
     $              (IA(IP,IN)/DELT)) * DELT - IATRRED(IP,IN) -
     $              INFRREDHT(IP,IN) - INFRREDS(IP,IN)
          IOUTFL = IROUND(OUTFL)

*         When all individuals in final boxcar become infected or die
*         by attrition, outflow is set to 0.
          IF (IOUTFL.LT.0) THEN
              IOUTFL = 0
          ENDIF

*         Second instar larvae that turn third instar larvae are divided
*         in a susceptible and resistant population
          IOUTFLS = IROUND(0.5*IOUTFL)
          IOUTFLR = IOUTFL - IOUTFLS
      ENDIF

*     Output of IBOXPAT variables
      IF (ITASK.EQ.4) THEN
*         Total number in boxcar
          IATOT = IA(IP,0)
          DO 70 IR = 1,IN
              IATOT = IATOT + IA(IP,IR)
70        CONTINUE
      ENDIF

*     Calculation of IBOXPAT state variables
      IF (ITASK.EQ.5) THEN
*         Cyclic development stage integral
          CDS = CDS + DELT*DVR

*         Calculation of new states in boxcars
          IA(IP,0) = IA(IP,0) + IAIN - INFRREDHT(IP,0) -
     $              IATRRED(IP,0) - INFRREDS(IP,0)
          DO 40 IK = 1,IN-1
              IA(IP,IK) = IA(IP,IK) - INFRREDHT(IP,IK) -
     $                    IATRRED(IP,IK) - INFRREDS(IP,IK)
40        CONTINUE

*         Calculation of new state in final boxcar. The contents of this boxcar
*         is reduced with the outflow that takes place in the timestep.
          IA(IP,IN) = IA(IP,IN) - IOUTFLS - IOUTFLR - INFRREDHT(IP,IN) -
     $              IATRRED(IP,IN) - INFRREDS(IP,IN)

*         Check for negative boxcar contents
          IF (IA(IP,0).LT.0.OR.IA(IP,1).LT.0.OR.IA(IP,2).LT.0.OR.
     $        IA(IP,3).LT.0.OR.IA(IP,4).LT.0.) THEN
              WRITE (*,50)
```

```
50          FORMAT('IBOXPATL2 content negative!')
         ENDIF
      ENDIF

*     Calculation of shift for IBOXPAT variables
      IF (ITASK.EQ.6) THEN
*        Calculation of F
         F = 1. - IN*CV*CV

*        Check on parameter values
         IF (DVR*DELT.GE.F/IN) THEN
            STOP'DVR*DELT.GE.F/IN!'
         ENDIF

*        The shift takes place immediatly after the new states have been
*        calculated. The contents of the different boxcars are shifted.
         IF (CDS.GE.F/IN) THEN
            RA(IP,IN) = F*IA(IP,IN-1)
            IRA(IP,IN) = IROUND(RA(IP,IN))
            IA(IP,IN) = IA(IP,IN) + IRA(IP,IN)
            DO 60 IR = IN-1,2,-1
               RA(IP,IR) = F*IA(IP,IR-1)
               IRA(IP,IR) = IROUND(RA(IP,IR))
               IA(IP,IR) = IA(IP,IR) - IRA(IP,IR+1) + IRA(IP,IR)
60          CONTINUE

            IRA(IP,1) = IA(IP,0)
            IA(IP,1) = IA(IP,1) - IRA(IP,2) + IRA(IP,1)

*           The contents of the inflow boxcar is set to zero
            IA(IP,0) = 0

*           Cyclic development stage is updated after shift
            CDS = CDS - F/IN
         ENDIF
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* FUNCTION IROUND                                                      *
*                                                                      *
* Purpose: Stochastic rounding of real values into integers.           *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                     class *
* ----   ---- ---------------                                    ----- *
* IROUND  I4  Stochasticcally rounded real value into integer (-)   O  *
* VALUE   R4  Help variable for stochastic rounding (-)             I  *
*                                                                      *
* Fatal error checks: none                                             *
* warnings:           none                                             *
* Subroutines called: none                                             *
* File usage:         none                                             *
*----------------------------------------------------------------------*
      FUNCTION IROUND(VALUE)

      INTEGER IROUND
      REAL    VALUE
      SAVE

      IROUND = INT(VALUE + UNIFL())

      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* SUBROUTINE IBOXPUP                                                    *
*                                                                       *
* Purpose: Simulation of development and attrition of                   *
*          uninfected and infected pupae.                               *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
* name   type meaning (unit)                                      class *
* ----   ---- ---------------                                     ----- *
* ICOUNT  I4  Counter for boxcar train number (-)                   I   *
* ITASK   I4  Task that subroutine should perform (-)               I   *
* DELT    R4  Time step (d)                                         I   *
* IAIN    I4  Number of individuals flowing in boxcar train (#)     I   *
* IN      I4  Number of boxcars per boxcar train (#)                I   *
* DVR     R4  Development rate (d-1)                                I   *
* CV      R4  Coefficient of variation (-)                          I   *
* CDS     R4  Cyclic development stage (-)                          I   *
* SEXRAT  R4  Sexratio (-)                                          I   *
* RATR    R4  Relative attrition rate (d-1)                         I   *
* IA      I4  Number of individuals in boxcar (#)                   O   *
* IATOT   I4  Number of individuals in boxcar train (#)             O   *
* IOUTFLMAL                                                             *
*         I4  Outflow difference of pupea developing into male          *
*             moths in final boxcar per timestep (#)                O   *
* IOUTFLFEM                                                             *
*         I4  Outflow difference of pupea developing into female        *
*             moths in final boxcar per timestep (#)                O   *
* IATR    I4  Attrition difference of individuals in boxcar train       *
*             per timestep (#)                                      O   *
*                                                                       *
* Fatal error checks: DVR*DELT > F/IN                                   *
* warnings:           IA < 0                                            *
* Subroutines or functions called:                                      *
* own:                IROUND                                            *
* File usage:         none                                              *
*-----------------------------------------------------------------------*
      SUBROUTINE IBOXPUP(ICOUNT,ITASK,DELT,IAIN,IN,DVR,CV,CDS,
     $                   SEXRAT,RATR,
     $                       IA,IATOT,IOUTFLMAL,IOUTFLFEM,IATR)

      PARAMETER (IPMAX = 200)
      INTEGER   IN,IK,IA,IOUTFL,IOUTFLMAL,IOUTFLFEM,IRA,
     $          ICOUNT,ITASK,IAIN,IATOT,IATR,IR
      REAL      F,OUTFL,RA,CDS,SEXRAT,CV,RATR
      DIMENSION IA(0:4),RA(0:4),IRA(0:4),IATRRED(IPMAX,0:4),
     $          IOUTFL(2)
      SAVE

*     Calculation of rate variables
      IF (ITASK.EQ.3) THEN
```

```
*         Calculation of infection and attrition differences of larvae
          IATR = 0
          DO 10 IK = 0,IN
              REALATR = RATR*DELT*IA(IK)
              INTATR = IROUND(REALATR)
              IATR = INTATR + IATR
              IATRRED(ICOUNT,IK) = INTATR
10        CONTINUE

*         Calculation of outflow in final boxcar
          OUTFL = MIN( (DVR*IA(IN)/(1./IN-CDS)),
     $             (IA(IN)/DELT)) * DELT - IATRRED(ICOUNT,IN)
          IOUTFL(ICOUNT) = IROUND(OUTFL)

*         When all individuals in final boxcar become infected or die
*         by attrition, outflow is set to 0.
          IF (IOUTFL(ICOUNT).LT.0) IOUTFL(ICOUNT) = 0

*         Sexratio is fraction males
          IOUTFLMAL = IROUND(SEXRAT*IOUTFL(ICOUNT))
          IOUTFLFEM = IOUTFL(ICOUNT) - IOUTFLMAL
      ENDIF

*     Output of IBOXPUP variables
      IF (ITASK.EQ.4) THEN
*         Total number in boxcar
          IATOT = IA(0)
          DO 50 IR = 1,IN
              IATOT = IATOT + IA(IR)
50        CONTINUE
      ENDIF

*     Calculation of IBOXPUP state variables
      IF (ITASK.EQ.5) THEN
*         Cyclic development stage integral
          CDS = CDS + DELT*DVR

*         Calculation of new states in boxcars
          IA(0) = IA(0) + IAIN - IATRRED(ICOUNT,0)
          DO 20 IK = 1,IN-1
              IA(IK) = IA(IK) - IATRRED(ICOUNT,IK)
20        CONTINUE

*         Calculation of new state in final boxcar. The contents of this boxcar
*         is reduced with the outflow that takes place in the timestep.
          IA(IN) = IA(IN) - IOUTFLFEM - IOUTFLMAL - IATRRED(ICOUNT,IN)

*         Check for negative boxcar contents
          IF (IA(0).LT.0.OR.IA(1).LT.0.OR.IA(2).LT.0.OR.
     $        IA(3).LT.0.OR.IA(4).LT.0.) THEN
```

```
              WRITE (*,30)
30            FORMAT('IBOXPUP content negative!')
          ENDIF
      ENDIF

*     Calculation of shift for IBOXPUP variables
      IF (ITASK.EQ.6) THEN
*         Calculation of F
          F = 1. - IN*CV*CV

*         Check on parameter values
          IF (DVR*DELT.GE.F/IN) THEN
              STOP'DVR*DELT.GE.F/IN!'
          ENDIF

*         The shift takes place immediatly after the new states have been
*         calculated. The contents of the different boxcars are shifted.
          IF (CDS.GE.F/IN) THEN
              RA(IN) = F*IA(IN-1)
              IRA(IN) = IROUND(RA(IN))
              IA(IN) = IA(IN) + IRA(IN)
              DO 40 IR = IN-1,2,-1
                  RA(IR) = F*IA(IR-1)
                  IRA(IR) = IROUND(RA(IR))
                  IA(IR) = IA(IR) - IRA(IR+1) + IRA(IR)
40            CONTINUE

              IRA(1) = IA(0)
              IA(1) = IA(1) - IRA(2) + IRA(1)

*             The contents of the inflow boxcar is set to zero
              IA(0) = 0

*             Cyclic development stage is updated after shift
              CDS = CDS - F/IN
          ENDIF
      ENDIF

      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* SUBROUTINE IBOXCAR                                                    *
*                                                                       *
* Purpose: Simulation of development and attrition                     *
*          of uninfected and infected adults and oviduct.              *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                     class *
* ----   ---- ---------------                                    ----- *
* ICOUNT  I4  Counter for boxcar train number (-)                  I   *
* ITASK   I4  Task that subroutine should perform (-)              I   *
* DELT    R4  Time step (d)                                        I   *
* IAIN    I4  Number of individuals flowing in boxcar train (#)    I   *
* IN      I4  Number of boxcars per boxcar train (#)               I   *
* DVR     R4  Development rate (d-1)                               I   *
* CV      R4  Coefficient of variation (-)                         I   *
* CDS     R4  Cyclic development stage (-)                         I   *
* RATR    R4  Relative attrition rate (d-1)                        I   *
* IA      I4  Number of individuals in boxcar (#)                  O   *
* IATOT   I4  Number of individuals in boxcar train (#)            O   *
* IOUTFL  I4  Outflow difference of individuals from final boxcar      *
*             per timestep (#)                                     O   *
* IATR    I4  Attrition difference of individuals in boxcar train      *
*             per timestep (#)                                     O   *
*                                                                       *
* Fatal error checks: DVR*DELT > F/IN                                   *
* warnings:           IA < 0                                            *
* Subroutines or functions called:                                      *
* own:                IROUND                                            *
* File usage:         none                                              *
*-----------------------------------------------------------------------*
      SUBROUTINE IBOXCAR(ICOUNT,ITASK,DELT,IAIN,IN,DVR,CV,CDS,
     $                   RATR,
     $                       IA,IATOT,IOUTFL,IATR)

      PARAMETER (IPMAX = 200)
      INTEGER   IN,IK,IA,IOUTFL,IAIN,IRA,ICOUNT,ITASK,IATOT,
     $          IATR,IR,IATRRED
      REAL      F,OUTFL,RA,CDS,CV
      DIMENSION IRA(0:4),IA(0:4),RA(0:4),IATRRED(IPMAX,0:4)
      SAVE

*     Calculation of IBOXCAR rate variables
      IF (ITASK.EQ.3) THEN
*         Calculation of attrition rates of adults and oviduct
          IATR = 0
          DO 10 IK = 0,IN
             REALATR = RATR*DELT*IA(IK)
             INTATR = IROUND(REALATR)
             IATR = INTATR + IATR
```

```
                IATRRED(ICOUNT,IK) = INTATR
10          CONTINUE

*           Calculation of outflow in final boxcar
            OUTFL = MIN( (DVR*IA(IN)/(1./IN-CDS)),
     $                (IA(IN)/DELT)) * DELT - IATRRED(ICOUNT,IN)
            IOUTFL = IROUND(OUTFL)

*           When all individuals in final boxcar become infected or die
*           by attrition, outflow is set to 0.
            IF (IOUTFL.LT.0) THEN
                IOUTFL = 0
            ENDIF
        ENDIF

*       Output of IBOXCAR variables
        IF (ITASK.EQ.4) THEN
*           Total number in boxcar
            IATOT = IA(0)
            DO 50 IR = 1,IN
                IATOT = IATOT + IA(IR)
50          CONTINUE
        ENDIF

*       Calculation of IBOXCAR state variables
        IF (ITASK.EQ.5) THEN
*           Cyclic development stage integral
            CDS = CDS + DELT*DVR

*           Calculation of new states in boxcars
            IA(0) = IA(0) + IAIN - IATRRED(ICOUNT,0)
            DO 20 IK = 1,IN-1
                IA(IK) = IA(IK) - IATRRED(ICOUNT,IK)
20          CONTINUE

*           Calculation of new state in final boxcar. The contents of this boxcar
*           is reduced with the outflow that takes place in the timestep.
            IA(IN) = IA(IN) - IOUTFL - IATRRED(ICOUNT,IN)

*           Check for negative boxcar contents
            IF (IA(0).LT.0.OR.IA(1).LT.0.OR.IA(2).LT.0.OR.
     $          IA(3).LT.0.OR.IA(4).LT.0.) THEN
                WRITE (*,30)
30              FORMAT('IBOXCAR content negative!')
            ENDIF
        ENDIF

*       Calculation of shift for IBOXCAR variables
        IF (ITASK.EQ.6) THEN
*           Calculation of F
```

```
          F = 1. - IN*CV*CV

*         Check on parameter values
          IF (DVR*DELT.GE.F/IN) THEN
              STOP'DVR*DELT.GE.F/IN!'
          ENDIF

*         The shift takes place immediatly after the new states have been
*         calculated. The contents of the different boxcars are shifted.
          IF (CDS.GE.F/IN) THEN
              RA(IN) = F*IA(IN-1)
              IRA(IN) = IROUND(RA(IN))
              IA(IN) = IA(IN) + IRA(IN)
              DO 40 IR = IN-1,2,-1
                  RA(IR) = F*IA(IR-1)
                  IRA(IR) = IROUND(RA(IR))
                  IA(IR) = IA(IR) - IRA(IR+1) + IRA(IR)
40            CONTINUE

              IRA(1) = IA(0)
              IA(1) = IA(1) - IRA(2) + IRA(1)

*             The contents of the inflow boxcar is set to zero
              IA(0) = 0

*             Cyclic development stage is updated after shift
              CDS = CDS - F/IN
          ENDIF
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE IBOXPTI                                                   *
*                                                                      *
* Purpose: Simulation of development, attrition and mortality          *
*          of infected insects in patches.                             *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                    class *
* ----   ---- ---------------                                   ----- *
* ICOUNT  I4  Counter for boxcar train number (-)                  I  *
* ITASK   I4  Task that subroutine should perform (-)              I  *
* DELT    R4  Time step (d)                                        I  *
* IAIN    I4  Number of individuals flowing in boxcar train (#)    I  *
* IN      I4  Number of boxcars per boxcar train (#)               I  *
* DVR     R4  Development rate (d-1)                               I  *
* CV      R4  Coefficient of variation (-)                         I  *
* CDS     R4  Cyclic development stage (-)                         I  *
* IP      I4  Pointer for patch number (-)                         I  *
* RATR    R4  Relative attrition rate (d-1)                        I  *
* IA      I4  Number of individuals in boxcar (#)                  O  *
* IATOT   I4  Number of individuals in boxcar train (#)            O  *
* IOUTFL  I4  Outflow difference of individuals from final boxcar     *
*             per timestep (#)                                     O  *
*                                                                      *
* Fatal error checks: DVR*DELT > F/IN                                  *
* warnings:           IA < 0                                           *
* Subroutines or functions called:                                     *
* own:                IROUND                                           *
* File usage:         none                                             *
*----------------------------------------------------------------------*
      SUBROUTINE IBOXPTI(ICOUNT,ITASK,DELT,IAIN,IN,DVR,CV,CDS,IP,RATR,
     $                        IA,IATOT,IOUTFL)

      PARAMETER (IPMAX = 200)
      INTEGER   IN,IP,IA,IOUTFL,IRA,ITASK,IAIN,IATOT,IR,IK,
     $          IATRRED,INTATR
      REAL      F,OUTFL,RA,CDS,CV,REALATR
      DIMENSION IRA(IPMAX,0:4),IA(IPMAX,0:4),RA(IPMAX,0:4),
     $          IATRRED(6,IPMAX,0:4),REALATR(0:4)
      SAVE

*     Calculation of IBOXPTI rate variables
      IF (ITASK.EQ.3) THEN

*        Calculation of attrition rates of infected larvae
         DO 10 IK = 0,IN
            REALATR(IK) = RATR*DELT*IA(IP,IK)
            INTATR = IROUND(REALATR(IK))
            IATRRED(ICOUNT,IP,IK) = INTATR
10       CONTINUE
```

```
*          Calculation of outflow in final boxcar
           OUTFL = MIN( (DVR*IA(IP,IN)/(1./IN-CDS)),
     $              (IA(IP,IN)/DELT)) * DELT - IATRRED(ICOUNT,IP,IN)
           IOUTFL = IROUND(OUTFL)

*          When all individuals in final boxcar become infected or die
*          by attrition, outflow is set to 0.
           IF (IOUTFL.LT.0) THEN
              IOUTFL = 0
           ENDIF
        ENDIF

*       Output of IBOXPTI variables
        IF (ITASK.EQ.4) THEN
*          Total number in boxcar
           IATOT = IA(IP,0)
           DO 50 IR = 1,IN
              IATOT = IATOT + IA(IP,IR)
50         CONTINUE
        ENDIF

*       Calculation of IBOXPTI state variables
        IF (ITASK.EQ.5) THEN
*          Cyclic development stage integral
           CDS = CDS + DELT*DVR

*          Calculation of new states in boxcars
           IA(IP,0) = IA(IP,0) + IAIN - IATRRED(ICOUNT,IP,0)

           DO 20 IK = 1,IN-1
              IA(IP,IK) = IA(IP,IK) - IATRRED(ICOUNT,IP,IK)
20         CONTINUE

*          Calculation of new state in final boxcar. The contents of this boxcar
*          is reduced with the outflow that takes place in the timestep.
*          Check for negative boxcar contents.
           IA(IP,IN) = IA(IP,IN) - IOUTFL - IATRRED(ICOUNT,IP,IN)
           IF (IA(IP,0).LT.0.OR.IA(IP,1).LT.0.OR.IA(IP,2).LT.0.OR.
     $        IA(IP,3).LT.0.OR.IA(IP,4).LT.0.) THEN
              WRITE (*,30)
30            FORMAT('IBOXPTI content negative!')
           ENDIF
        ENDIF

*       Calculation of shift for IBOXPTI variables
        IF (ITASK.EQ.6) THEN
*          Calculation of F
           F = 1. - IN*CV*CV
```

```
*        Check on parameter values
         IF (DVR*DELT.GE.F/IN) THEN
             STOP'DVR*DELT.GE.F/IN!'
         ENDIF

*        The shift takes place immediatly after the new states have been
*        calculated. The contents of the different boxcars are shifted.
         IF (CDS.GE.F/IN) THEN
             RA(IP,IN)  = F*IA(IP,IN-1)
             IRA(IP,IN)  = IROUND(RA(IP,IN))
             IA(IP,IN)  = IA(IP,IN)  + IRA(IP,IN)
             DO 40 IR = IN-1,2,-1
                 RA(IP,IR)  = F*IA(IP,IR-1)
                 IRA(IP,IR)  = IROUND(RA(IP,IR))
                 IA(IP,IR)  = IA(IP,IR)  - IRA(IP,IR+1)  + IRA(IP,IR)
40           CONTINUE

             IRA(IP,1)  = IA(IP,0)
             IA(IP,1)  = IA(IP,1)  - IRA(IP,2)  + IRA(IP,1)

*            The contents of the inflow boxcar is set to zero
             IA(IP,0)  = 0

*            Cyclic development stage is updated after shift
             CDS = CDS - F/IN
         ENDIF
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE IBOXPATL5                                                 *
*                                                                      *
* Purpose: Simulation of development, infection and attrition of       *
*          uninfected L5 larvae in patches. A fraction of the infected *
*          L5 larvae will die due to virus infection (INFRHTD)         *
*          and the rest will develop in infected pupae (INFRHTS).      *
*
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* ICOUNT   I4  Counter for boxcar train number (-)                 I   *
* ITASK    I4  Task that subroutine should perform (-)             I   *
* DELT     R4  Time step (d)                                       I   *
* IAIN     I4  Number of individuals flowing in boxcar train (#)   I   *
* IN       I4  Number of boxcars per boxcar train (#)              I   *
* DVR      R4  Development rate (d-1)                              I   *
* CV       R4  Coefficient of variation (-)                        I   *
* CDS      R4  Cyclic development stage (-)                        I   *
* IP       I4  Pointer for patch number (-)                        I   *
* RATR     R4  Relative attrition rate (d-1)                       I   *
* RINFRHT R4  Relative infection rate by horizontal transmission       *
*              (d-1)                                               I   *
* RINFRS   R4  Relative infection rate by ingestion of sprayed         *
*              polyhedra (d-1)                                     I   *
* FINFL5S R4  Fraction of infected L5 larvae that will become          *
*              sublethally infected (-)                            I   *
* IA       I4  Number of individuals in boxcar (#)                 O   *
* IATOT    I4  Number of individuals in boxcar train (#)           O   *
* IOUTFL   I4  Outflow difference of individuals from final boxcar     *
*              per timestep (#)                                    O   *
* IATR     I4  Attrition difference of individuals in boxcar train     *
*              per timestep (#)                                    O   *
* INFRHTS I4  Infection difference by horizontal transmission of L5    *
*              larvae into sublethally infected L5 larvae in boxcar    *
*              train per timestep (#)                              O   *
* INFRHTD I4  Infection difference by horizontal transmission of       *
*              L5 larvae into lethally infected L5 larvae per          *
*              boxcar train per timestep (#)                       O   *
* INFRSS   I4  Infection difference by ingestion of sprayed            *
*              polyhedra to sublethally infected L5 larvae in          *
*              boxcar train per timestep (#)                       O   *
* INFRSD   I4  Infection difference by ingestion of sprayed            *
*              polyhedra to lethally infected L5 larvae in boxcar      *
*              train per timestep (#)                              O   *
* INFRREDHTS                                                           *
*          I4  Infection difference by horizontal transmission of      *
*              L5 larvae into sublethally infected L5 larvae per       *
*              boxcar per timestep (#)                             O   *
* INFRREDSS                                                          *
*         I4  Infection difference by ingestion of sprayed            *
*             polyhedra of sublethally infected L5 larvae per         *
*             boxcar per timestep (#)                              O  *
*                                                                     *
* Fatal error checks: DVR*DELT > F/IN                                 *
* warnings:           IA < 0                                          *
* Subroutines or functions called:                                    *
* own:                IROUND,DRWMORT                                  *
* File usage:         none                                            *
*---------------------------------------------------------------------*
      SUBROUTINE IBOXPATL5(ICOUNT,ITASK,DELT,IAIN,IN,DVR,CV,CDS,
     $                    IP,RATR,RINFRHT,RINFRS,FINFL5S,
     $                        IA,IATOT,IOUTFL,IATR,INFRHTS,INFRHTD,
     $                        INFRSS,INFRSD,INFRREDHTS,INFRREDSS)

      PARAMETER (IPMAX = 200)
      INTEGER   IN,IK,IA,IOUTFL,IAIN,IRA,ITASK,IP,IATOT,IATR,
     $          IR,INTINFHT,INTATR,IATRRED,INFRREDHT,INFRREDS,
     $          INFRREDHTD,INFRREDSD,INFRREDHTS,INFRREDSS,INFRHTS,
     $          INFRHTD,INFRSD,INFRSS
      REAL      F,OUTFL,RA,CDS,RINFRHT,REALINFHT,REALATR,REALINFS,
     $          RINFRS,CV,FINFL5S
      DIMENSION IRA(IPMAX,0:4),IA(IPMAX,0:4),RA(IPMAX,0:4),
     $          IATRRED(2,IPMAX,0:4),INFRREDHT(2,IPMAX,0:4),
     $          INFRREDHTD(IPMAX,0:4),INFRREDHTS(IPMAX,0:4),
     $          INFRREDS(2,IPMAX,0:4),INFRREDSD(IPMAX,0:4),
     $          INFRREDSS(IPMAX,0:4),REALINFHT(0:4),
     $          REALINFS(0:4),REALATR(0:4)
      SAVE

*     Calculation of IBOXPATL5 rate variables
      IF (ITASK.EQ.3) THEN
*        Calculation of infection and attrition differences of larvae
         DO 10 IK = 0,IN
            REALINFHT(IK) = RINFRHT*DELT*IA(IP,IK)
            INTINFHT = IROUND(REALINFHT(IK))
            INFRREDHT(ICOUNT,IP,IK) = INTINFHT

            REALINFS(IK) = RINFRS*DELT*IA(IP,IK)
            INTINFS = IROUND(REALINFS(IK))
            INFRREDS(ICOUNT,IP,IK) = INTINFS

            REALATR(IK) = RATR*DELT*IA(IP,IK)
            INTATR = IROUND(REALATR(IK))
            IATRRED(ICOUNT,IP,IK) = INTATR
10       CONTINUE

*        Mortality in boxcar cannot be higher as boxcar contents
         DO 20 IK = 0,4
```

```
            IF ((IATRRED(ICOUNT,IP,IK) + INFRREDHT(ICOUNT,IP,IK) +
     $        INFRREDS(ICOUNT,IP,IK)).GT.IA(IP,IK)) THEN

*             Drawing of process responsible for death of larvae
              CALL DRWMORT(IA(IP,IK),REALATR(IK),REALINFS(IK),
     $                     REALINFHT(IK),
     $                     IATRRED(ICOUNT,IP,IK),INFRREDS(ICOUNT,IP,IK),
     $                     INFRREDHT(ICOUNT,IP,IK))
            ENDIF
20        CONTINUE

*         Calculation of number of infected L5 larvae that will get
*         infected, sublethally infected and die by attrition
          IATR = 0
          INFRHTD = 0
          INFRHTS = 0
          INFRSD = 0
          INFRSS = 0

          DO 30 IK = 0,IN
            INFRREDHTS(IP,IK) = IROUND(INFRREDHT(ICOUNT,IP,IK)*
     $                                 FINFL5S)
            INFRREDHTD(IP,IK) = INFRREDHT(ICOUNT,IP,IK) -
     $                                 INFRREDHTS(IP,IK)
            INFRREDSS(IP,IK) = IROUND(INFRREDS(ICOUNT,IP,IK)*
     $                                FINFL5S)
            INFRREDSD(IP,IK) = INFRREDS(ICOUNT,IP,IK) -
     $                                INFRREDSS(IP,IK)

            IATR = IATR + IATRRED(ICOUNT,IP,IK)
            INFRHTD = INFRHTD + INFRREDHTD(IP,IK)
            INFRHTS = INFRHTS + INFRREDHTS(IP,IK)
            INFRSD = INFRSD + INFRREDSD(IP,IK)
            INFRSS = INFRSS + INFRREDSS(IP,IK)
30        CONTINUE

*         Calculation of outflow in final boxcar
          OUTFL = MIN( (DVR*IA(IP,IN)/(1./IN-CDS)),
     $          (IA(IP,IN)/DELT)) * DELT - IATRRED(ICOUNT,IP,IN) -
     $          INFRREDHT(ICOUNT,IP,IN) - INFRREDS(ICOUNT,IP,IN)
          IOUTFL = IROUND(OUTFL)

*         When all individuals in final boxcar become infected or die
*         by attrition, outflow is set to 0.
          IF (IOUTFL.LT.0) THEN
             IOUTFL = 0
          ENDIF
      ENDIF

*     Output of IBOXPATL5 variables
```

```
      IF (ITASK.EQ.4) THEN
*         Total number in boxcar
          IATOT = IA(IP,0)
          DO 70 IR = 1,IN
              IATOT = IATOT + IA(IP,IR)
70        CONTINUE
      ENDIF

*     Calculation of IBOXPATL5 state variables
      IF (ITASK.EQ.5) THEN
*         Cyclic development stage integral
          CDS = CDS + DELT*DVR

*         Calculation of new states in boxcars
          IA(IP,0) = IA(IP,0) + IAIN - INFRREDHT(ICOUNT,IP,0) -
     $              IATRRED(ICOUNT,IP,0) - INFRREDS(ICOUNT,IP,0)
          DO 40 IK = 1,IN-1
          IA(IP,IK) = IA(IP,IK) - INFRREDHT(ICOUNT,IP,IK) -
     $                 IATRRED(ICOUNT,IP,IK) - INFRREDS(ICOUNT,IP,IK)
40        CONTINUE

*         Calculation of new state in final boxcar. The contents of this boxcar
*         is reduced with the outflow that takes place in the timestep.
          IA(IP,IN) = IA(IP,IN) - IOUTFL -INFRREDHT(ICOUNT,IP,IN) -
     $                IATRRED(ICOUNT,IP,IN) - INFRREDS(ICOUNT,IP,IN)

*         Check for negative boxcar contents
          IF (IA(IP,0).LT.0.OR.IA(IP,1).LT.0.OR.IA(IP,2).LT.0.OR.
     $        IA(IP,3).LT.0.OR.IA(IP,4).LT.0.) THEN
              WRITE (*,50)
50            FORMAT('IBOXPATL5 content negative!')
          ENDIF
      ENDIF

*     Calculation of shift
      IF (ITASK.EQ.6) THEN
*         Calculation of F
          F = 1. - IN*CV*CV

*         Check on parameter values
          IF (DVR*DELT.GE.F/IN) THEN
              STOP'DVR*DELT.GE.F/IN!'
          ENDIF

*         The shift takes place immediatly after the new states have been
*         calculated. The contents of the different boxcars are shifted.
          IF (CDS.GE.F/IN) THEN
              RA(IP,IN) = F*IA(IP,IN-1)
              IRA(IP,IN) = IROUND(RA(IP,IN))
              IA(IP,IN) = IA(IP,IN) + IRA(IP,IN)
              DO 60 IR = IN-1,2,-1
                  RA(IP,IR) = F*IA(IP,IR-1)
                  IRA(IP,IR) = IROUND(RA(IP,IR))
                  IA(IP,IR) = IA(IP,IR) - IRA(IP,IR+1) + IRA(IP,IR)
60            CONTINUE

              IRA(IP,1) = IA(IP,0)
              IA(IP,1) = IA(IP,1) - IRA(IP,2) + IRA(IP,1)

*             The contents of the inflow boxcar is set to zero
              IA(IP,0) = 0

*             Cyclic development stage is updated after shift
              CDS = CDS - F/IN
          ENDIF
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE IBOXPTIS                                                  *
*                                                                      *
* Purpose: Simulation of development and attrition of sublethally      *
*          infected L5 larvea in patches that develop in sublethally   *
*          infected adults.                                            *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                     class *
* ----   ---- ---------------                                    ----- *
* ITASK   I4  Task that subroutine should perform (-)               I  *
* DELT    R4  Time step (d)                                         I  *
* IRL5SHTSAR                                                           *
*         I4  Array of infection differences by horizontal             *
*             transmission of susceptible L5 larvae into               *
*             sublethally infected L5 larvae per timestep (#)       I  *
* IRL5RHTSAR                                                           *
*         I4  Array of infection differences by horizontal             *
*             transmission of resistant L5 larvae into sublethally     *
*             infected L5 larvae per timestep (#)                   I  *
* IRL5SSSAR                                                            *
*         I4  Array of Infection differences by ingestion of           *
*             sprayed polyhedra of susceptible L5 larvae into          *
*             sublethally infected L5 larvae per timestep (#)       I  *
* IRL5RSSAR                                                            *
*         I4  Array of infection differences by ingestion of           *
*             sprayed polyhedra of resistant L5 larvae into            *
*             sublethally infected L5 larvae per timestep (#)       I  *
* IN      I4  Number of boxcars per boxcar train (#)                I  *
* DVR     R4  Development rate (d-1)                                I  *
* CV      R4  Coefficient of variation (-)                          I  *
* CDS     R4  Cyclic development stage (-)                          I  *
* IP      I4  Pointer for patch number (-)                          I  *
* RATR    R4  Relative attrition rate (d-1)                         I  *
* IA      I4  Number of individuals in boxcar (#)                   O  *
* IATOT   I4  Number of individuals in boxcar train (#)             O  *
* IOUTFL  I4  Outflow difference of individuals from final boxcar      *
*             per timestep (#)                                      O  *
* IATR    I4  Attrition difference of individuals in boxcar train      *
*             per timestep (#)                                      O  *
*                                                                      *
* Fatal error checks: DVR*DELT > F/IN                                  *
* warnings:           IA < 0                                           *
* Subroutines or functions called:                                     *
* own:                IROUND                                           *
* File usage:         none                                             *
*----------------------------------------------------------------------*
      SUBROUTINE IBOXPTIS(ITASK,DELT,IRL5SHTSAR,IRL5RHTSAR,IRL5SSSAR,
     $                    IRL5RSSAR,IN,DVR,CV,CDS,IP,RATR,
     $                        IA,IATOT,IOUTFL,IATR)
```

```
      PARAMETER (IPMAX = 200)
      INTEGER   IN,IK,IP,IA,IOUTFL,IRA,ITASK,IATOT,IATR,
     $          IR,IATRRED,INTATR,IRL5SHTSAR,IRL5RHTSAR,
     $          IRL5SSSAR,IRL5RSSAR
      REAL      F,OUTFL,RA,CDS,REALATR,CV
      DIMENSION IRA(IPMAX,0:4),IA(IPMAX,0:4),RA(IPMAX,0:4),
     $          IATRRED(IPMAX,0:4),IRL5SHTSAR(IPMAX,0:4),
     $          IRL5RHTSAR(IPMAX,0:4),IRL5SSSAR(IPMAX,0:4),
     $          IRL5RSSAR(IPMAX,0:4)
      SAVE

*     Calculation of IBOXPTIS rate variables
      IF (ITASK.EQ.3) THEN
*         Calculation of infection rate and attrition rate of larvae
          IATR = 0
          DO 10 IK = 0,IN
             REALATR = RATR*DELT*IA(IP,IK)
             INTATR = IROUND(REALATR)
             IATR = INTATR + IATR
             IATRRED(IP,IK) = INTATR
10          CONTINUE

*         Calculation of outflow in final boxcar
          OUTFL = MIN( (DVR*IA(IP,IN)/(1./IN-CDS)),
     $            (IA(IP,IN)/DELT)) * DELT - IATRRED(IP,IN)
          IOUTFL = IROUND(OUTFL)

*         When all individuals in final boxcar become infected or die
*         by attrition, outflow is set to 0.
          IF (IOUTFL.LT.0) THEN
              IOUTFL = 0
          ENDIF
      ENDIF

*     Output of IBOXPTIS variables
      IF (ITASK.EQ.4) THEN
*         Total number in boxcar
          IATOT = IA(IP,0)
          DO 50 IR = 1,IN
              IATOT = IATOT + IA(IP,IR)
50        CONTINUE
      ENDIF

*     Calculation of IBOXPTIS state variables
      IF (ITASK.EQ.5) THEN
*         Cyclic development stage integral
          CDS = CDS + DELT*DVR

*         Calculation of new states in boxcars
```

```
          IA(IP,0) = IA(IP,0) + IRL5SHTSAR(IP,0) + IRL5RHTSAR(IP,0) +
     $             IRL5SSSAR(IP,0) + IRL5RSSAR(IP,0)- IATRRED(IP,0)
          DO 20 IK = 1,IN-1
              IA(IP,IK) = IA(IP,IK)+IRL5SHTSAR(IP,IK)+IRL5RHTSAR(IP,IK)+
     $                   IRL5SSSAR(IP,IK)+IRL5RSSAR(IP,IK)-
     $                   IATRRED(IP,IK)
20        CONTINUE

*         Calculation of new state in final boxcar. The contents of this boxcar
*         is reduced with the outflow that takes place in the timestep.
          IA(IP,IN) = IA(IP,IN) + IRL5SHTSAR(IP,IN) + IRL5RHTSAR(IP,IN)+
     $              IRL5SSSAR(IP,IN) + IRL5RSSAR(IP,IN) - IOUTFL -
     $              IATRRED(IP,IN)

*         Check for negative boxcar contents
          IF (IA(IP,0).LT.0.OR.IA(IP,1).LT.0.OR.IA(IP,2).LT.0.OR.
     $        IA(IP,3).LT.0.OR.IA(IP,4).LT.0.) THEN
              WRITE (*,30)
30            FORMAT('IBOXPTIS content negative!')
          ENDIF
      ENDIF

*     Calculation of shift for IBOXPTIS variables
      IF (ITASK.EQ.6) THEN
*         Calculation of F
          F = 1. - IN*CV*CV

*         Check on parameter values
          IF (DVR*DELT.GE.F/IN) THEN
              STOP'DVR*DELT.GE.F/IN!'
          ENDIF

*         The shift takes place immediatly after the new states have been
*         calculated. The contents of the different boxcars are shifted.
          IF (CDS.GE.F/IN) THEN
              RA(IP,IN) = F*IA(IP,IN-1)
              IRA(IP,IN) = IROUND(RA(IP,IN))
              IA(IP,IN) = IA(IP,IN) + IRA(IP,IN)
              DO 40 IR = IN-1,2,-1
                  RA(IP,IR) = F*IA(IP,IR-1)
                  IRA(IP,IR) = IROUND(RA(IP,IR))
                  IA(IP,IR) = IA(IP,IR) - IRA(IP,IR+1) + IRA(IP,IR)
40            CONTINUE

              IRA(IP,1) = IA(IP,0)
              IA(IP,1) = IA(IP,1) - IRA(IP,2) + IRA(IP,1)

*             The contents of the inflow boxcar is set to zero
              IA(IP,0) = 0
```

```
*              Cyclic development stage is updated after shift
               CDS = CDS - F/IN
           ENDIF
       ENDIF

       RETURN
       END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE SPADIS                                                    *
*                                                                      *
* Purpose: Spatial distribution of Spodoptera exigua. The location of  *
*          egg batches and foraging domains of larvae as well as the   *
*          location of cadavers in the crop are stored.                *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* ITASK    I4  Task that subroutine should perform (-)              I  *
* IPATLOW I4  Pointer for lowest patch number containing larvae (-) I  *
* IPATCUM I4  Cumulative number of patches in greenhouse (#)        I  *
* IRL1HT   I4  Infection difference by horizontal transmission of      *
*              L1 larvae per timestep (#)                           I  *
* IRL2HT   I4  Infection difference by horizontal transmission of      *
*              L2 larvae per timestep (#)                           I  *
* IRL3SHT I4  Infection difference by horizontal transmission of       *
*              susceptible L3 larvae per timestep (#)               I  *
* IRL3RHT I4  Infection difference by horizontal transmission of       *
*              resistant L3 larvae per timestep (#)                 I  *
* IRL4SHT I4  Infection difference by horizontal transmission of       *
*              susceptible L4 larvae per timestep (#)               I  *
* IRL4RHT I4  Infection difference by horizontal transmission of       *
*              resistant L4 larvae per timestep (#)                 I  *
* IRL5SHTD                                                             *
*          I4  Infection difference by horizontal transmission of      *
*              susceptible L5 larvae into lethally infected L5         *
*              larvae per timestep (#)                              I  *
* IRL5RHTD                                                             *
*          I4  Infection difference by horizontal transmission of      *
*              resistant L5 larvae into lethally infected L5 larvae    *
*              per timestep (#)                                     I  *
* IRL1S    I4  Infection difference by ingestion of sprayed            *
*              polyhedra of L1 larvae per timestep (#)              I  *
* IRL2S    I4  Infection difference by ingestion of sprayed            *
*              polyhedra of L2 larvae per timestep (#)              I  *
* IRL3SS   I4  Infection difference by ingestion of sprayed            *
*              polyhedra of susceptible L3 larvae per timestep (#)  I  *
* IRL3RS   I4  Infection difference by ingestion of sprayed            *
*              polyhedra of resistant L3 larvae per timestep (#)    I  *
* IRL4SS   I4  Infection difference by ingestion of sprayed            *
*              polyhedra of susceptible L4 larvae per timestep (#)  I  *
* IRL4RS   I4  Infection difference by ingestion of sprayed            *
*              polyhedra of resistant L4 larvae per timestep (#)    I  *
* IRL5SSD I4   Infection difference by ingestion of sprayed            *
*              polyhedra of susceptible L5 larvae into lethally        *
*              infected L5 larvae per timestep (#)                  I  *
* IRL5RSD I4   Infection difference by ingestion of sprayed            *
*              polyhedra of resistant L5 larvae into lethally          *
*             infected L5 larvae per timestep (#)                I  *
* IDRL1I   I4  Death difference of infected L1 larvae per time step    *
*             (#)                                                I  *
* IDRL2I   I4  Death difference of infected L2 larvae per time step    *
*             (#)                                                I  *
* IDRL3I   I4  Death difference of infected L3 larvae per time step    *
*             (#)                                                I  *
* IDRL4I   I4  Death difference of infected L4 larvae per time step    *
*             (#)                                                I  *
* IDRL5ID I4  Death difference of lethally infected L5 larvae per     *
*             time step (#)                                      I  *
* IPATL1I I4  Number of infected L1 larvae in boxcar train per        *
*             patch (#)                                          I  *
* IPATL2I I4  Number of infected L2 larvae in boxcar train per        *
*             patch (#)                                          I  *
* IPATL3I I4  Number of infected L3 larvae in boxcar train per        *
*             patch (#)                                          I  *
* IPATL4I I4  Number of infected L4 larvae in boxcar train per        *
*             patch (#)                                          I  *
* ITIL5D   I4  Number of lethally infected L5 larvae per boxcar (#) I  *
* IBATOUT I4  Number of batches in oviduct flowing out IBATCH         *
*             boxcar train (#)                                   I  *
* IBATOUTI                                                            *
*         I4  Number of batches in infected oviduct flowing out       *
*             IBATCHI boxcar train (#)                           I  *
* IHATCHI I4  Hatch difference of infected L1 larvae per patch per    *
*             time step (#)                                      I  *
* ILPP     I4  Number of leaves per plant in patch (#)           I  *
* LAIBED   R4  Leaf area index of crop in bed (m2.m-2)           I  *
* IHARV    I4  Flag for harvest (-)                              I  *
* FCLVSL1 R4  Fraction contaminated leaves in L1 foraging domain      *
*             (-)                                                O  *
* FCLVSL2 R4  Fraction contaminated leaves in L2 foraging domain      *
*             (-)                                                O  *
* FCLVSL3 R4  Fraction contaminated leaves in L3 foraging domain      *
*             (-)                                                O  *
* FCLVSL4 R4  Fraction contaminated leaves in L4 foraging domain      *
*             (-)                                                O  *
* FCLVSL5 R4  Fraction contaminated leaves in L5 foraging domain      *
*             (-)                                                O  *
* VERTDIST                                                            *
*         R4  Fraction larvae per canopy layer (-)               O  *
* IPATDAT I4  Array containing co-ordinates of patches in             *
*             greenhouse (-)                                     O  *
* IPLUP    I4  Pointer for infected larvae with highest rank number    *
*             (-)                                                O  *
* IPCLVS   I4  Pointer for number of contaminated leaves (-)     O  *
* ICAD     I4  Number of cadavers in greenhouse (#)              O  *
* IPLANTL1                                                            *
*         I4  Number of plants in foraging domain of L1 larvae (#) O  *
* IPLANTL2                                                        *
*         I4  Number of plants in foraging domain of L2 larvae (#)  O  *
* IPLANTL3                                                        *
*         I4  Number of plants in foraging domain of L3 larvae (#)  O  *
* IPLANTL4                                                        *
*         I4  Number of plants in foraging domain of L4 larvae (#)  O  *
* IPLANTL5                                                        *
*         I4  Number of plants in foraging domain of L5 larvae (#)  O  *
* INPLANT I4  Number of plants per bed (#)                          O  *
* PHDLARTOT                                                       *
*         R4  Total number of polyhedra produced in larvae (#)      O  *
*                                                                 *
* Fatal error checks: INPAT > IPMAX                               *
*                     INCLVS > IPOINM                             *
*                     INPAT > IBEDMAX                             *
*                     INPAT > IROWMAX                             *
*                     INPAT > IPPRMAX                             *
*                     INFOCLVS > IBEDMAX                          *
*                     INFOCLVS > IROWMAX                          *
*                     INFOCLVS > IPPRMAX                          *
*                     INFOCLVS > LPPMAX                           *
* warnings:           none                                        *
* Subroutines or functions called:                                *
* from TTUTIL:        RDINIT,RDSINT,RDFINT,RDFREA,RDSREA,         *
*                     OUTDAT                                      *
* own:                NEWINF,NEWDIL,CONTROL,HARV,IDRWLOT          *
* File usage:         INCON.DAT, SPADIS.DAT, CROP.DAT, VIRUS.DAT  *
*-----------------------------------------------------------------*
      SUBROUTINE SPADIS(ITASK,IPATLOW,IPATCUM,IRL1HT,IRL2HT,
     $                  IRL3SHT,IRL3RHT,IRL4SHT,IRL4RHT,IRL5SHTD,
     $                  IRL5RHTD,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $                  IRL4RS,IRL5SSD,IRL5RSD,IDRL1I,IDRL2I,IDRL3I,
     $                  IDRL4I,IDRL5ID,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $                  ITIL5D,IBATOUT,IBATOUTI,IHATCHI,ILPP,LAIBED,
     $                  IHARV,
     $                      FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,
     $                      VERTDIST,IPATDAT,IPLUP,IPCLVS,ICAD,IPLANTL1,
     $                      IPLANTL2,IPLANTL3,IPLANTL4,IPLANTL5,INPLANT,
     $                      PHDLARTOT)

      PARAMETER (IPOINM = 1000)
      PARAMETER (IPMAX = 200)
      PARAMETER (ICLMAX = 3)
      PARAMETER (ICLVSMAX = 400)
      INTEGER   INFOCLVS,INFOLIL,IPATDAT,ICLVS,IPATLOW,ICLVSNRL1,
     $          ICLVSNRL2,ICLVSNRL3,ICLVSNRL4,ICLVSNRL5,IPPRDIF,IROWDIF,
     $          INCLVS,INPAT,IPCLVS,IPLLOW,IPLUP,ICLVSL1CL,ICLVSL2CL,
     $          ICLVSL3CL,ICLVSL4CL,ICLVSL5CL,ICLCLVS,IPLANTL1,IPLANTL2,
     $          IPLANTL3,IPLANTL4,IPLANTL5,IPATCUM,IRL1HT,IRL2HT,
     $          IRL3SHT,IRL3RHT,IRL4SHT,IRL4RHT,IRL5SHTD,IRL5RHTD,
     $          IDRL1I,IDRL2I,IDRL3I,IDRL4I,IDRL5ID,ICLVSL1,ICLVSL2,
     $          ICLVSL3,ICLVSL4,ICLVSL5,IPAT1I,IPAT2I,IPAT3I,IPAT4I,
     $          IPAT5I,ITIL5D,IRL1S,IRL2S,IRL3SS,IRL3RS,IRL4SS,
     $          IRL4RS,IRL5SSD,IRL5RSD,ILPP,IBEDMAX,IROWMAX,IPPRMAX,
     $          IHORDIST,IHATCHI,IACTRADL1,IACTRADL2,IACTRADL3,
     $          IACTRADL4,IACTRADL5,IBEDPREF,IBEDDEP,IHARV,IP,IB,IFA,
     $          ICL,INEWBAT,IL,ILPPBED,ILVSCL,INPLANT
      REAL      FCLVSL1,FCLVSL2,FCLVSL3,FCLVSL4,FCLVSL5,VERTDIST,
     $          LAIBED,LAIPREF,PHDPROD,PHDLARTOT,LPP0,RGRLPP,LPPMAX,
     $          DAPLPP50,DAP,REST
      LOGICAL   FORAR
      DIMENSION ICLVSL1CL(IPMAX,ICLMAX),ICLVSL2CL(IPMAX,ICLMAX),
     $          ICLVSL3CL(IPMAX,ICLMAX),ICLVSL4CL(IPMAX,ICLMAX),
     $          ICLVSL5CL(IPMAX,ICLMAX),FCLVSL1(IPMAX,ICLMAX),
     $          FCLVSL2(IPMAX,ICLMAX),FCLVSL3(IPMAX,ICLMAX),
     $          FCLVSL4(IPMAX,ICLMAX),FCLVSL5(IPMAX,ICLMAX),
     $          IRL1HT(IPMAX),IRL2HT(IPMAX),IRL3SHT(IPMAX),
     $          IRL3RHT(IPMAX),IRL4SHT(IPMAX),IRL4RHT(IPMAX),
     $          IRL5SHTD(IPMAX),IRL5RHTD(IPMAX),IDRL1I(IPMAX),
     $          IDRL2I(IPMAX),IDRL3I(IPMAX),IDRL4I(IPMAX),
     $          IDRL5ID(IPMAX),INFOCLVS(5,IPOINM),INFOLIL(6,IPOINM),
     $          IPATDAT(3,IPMAX),IHORDIST(4,5),ICLVS(IPMAX),
     $          ICLVSNRL1(IPMAX,ICLVSMAX),ICLVSNRL2(IPMAX,ICLVSMAX),
     $          ICLVSNRL3(IPMAX,ICLVSMAX),ICLVSNRL4(IPMAX,ICLVSMAX),
     $          ICLVSNRL5(IPMAX,ICLVSMAX),IPPRDIF(IPOINM),
     $          IROWDIF(IPOINM),ICLVSL1(IPMAX),ICLVSL2(IPMAX),
     $          ICLVSL3(IPMAX),ICLVSL4(IPMAX),ICLVSL5(IPMAX),
     $          IPATL1I(IPMAX),IPATL2I(IPMAX),IPATL3I(IPMAX),
     $          IPATL4I(IPMAX),ITIL5D(IPMAX),IRL1S(IPMAX),IRL2S(IPMAX),
     $          IRL3SS(IPMAX),IRL3RS(IPMAX),IRL4SS(IPMAX),IRL4RS(IPMAX),
     $          IRL5SSD(IPMAX),IRL5RSD(IPMAX),ILPP(IPMAX),VERTDIST(3,5),
     $          IHATCHI(IPMAX),IBEDDEP(IPMAX),LAIBED(IPMAX),FORAR(5),
     $          PHDPROD(5),ILPPBED(IPMAX),DAP(50),ILVSCL(IPMAX,ICLMAX)
      SAVE

*     Initialization of SPADIS variables
      IF (ITASK.EQ.1) THEN
*         Initialization of arrays
          DO 10 IP = 1,IPOINM
              INFOLIL(1,IP) = 0
              INFOLIL(2,IP) = 0
              INFOLIL(3,IP) = 0
              INFOLIL(4,IP) = 0
              INFOLIL(5,IP) = 0
              INFOLIL(6,IP) = 0
10        CONTINUE
          DO 20 IP = 1,IPOINM
              INFOCLVS(1,IP) = 0
              INFOCLVS(2,IP) = 0
              INFOCLVS(3,IP) = 0
              INFOCLVS(4,IP) = 0
              INFOCLVS(5,IP) = 0
20        CONTINUE
          DO 40 IP = 1,IPMAX
              IPATDAT(1,IP) = 0
              IPATDAT(2,IP) = 0
              IPATDAT(3,IP) = 0
              DO 30 IL = 1,ICLVSMAX
                  ICLVSNRL1(IP,IL) = 0
                  ICLVSNRL2(IP,IL) = 0
                  ICLVSNRL3(IP,IL) = 0
                  ICLVSNRL4(IP,IL) = 0
                  ICLVSNRL5(IP,IL) = 0
30            CONTINUE
40        CONTINUE

*         Read initial conditions
          CALL RDINIT (30,40,'INCON.DAT')
          CALL RDFREA ('DAP',DAP(1),50,50)
          CALL RDSINT ('INCLVS',INCLVS)
          IF (INCLVS.GT.0) THEN
              CALL RDFINT ('INFOCLVS',INFOCLVS(1,1),100,5*INCLVS)
          ENDIF
          CALL RDSINT ('INPAT',INPAT)
          IF (INPAT.GT.0) THEN
              CALL RDFINT ('IPATDAT',IPATDAT(1,1),150,3*INPAT)
          ENDIF
          CALL RDSINT ('IBEDMAX',IBEDMAX)
          CALL RDSINT ('IROWMAX',IROWMAX)
          CALL RDSINT ('IPPRMAX',IPPRMAX)
*         Read parameter values
          CALL RDINIT (30,40,'SPADIS.DAT')
          CALL RDFINT ('IHORDIST',IHORDIST(1,1),20,20)
          CALL RDFREA ('VERTDIST',VERTDIST(1,1),15,15)
          CALL RDSINT ('IACTRADL1',IACTRADL1)
          CALL RDSINT ('IACTRADL2',IACTRADL2)
          CALL RDSINT ('IACTRADL3',IACTRADL3)
          CALL RDSINT ('IACTRADL4',IACTRADL4)
          CALL RDSINT ('IACTRADL5',IACTRADL5)
          CALL RDINIT (30,40,'CROP.DAT')
          CALL RDSREA ('LAIPREF',LAIPREF)
          CALL RDSREA ('RGRLPP',RGRLPP)
          CALL RDSREA ('DAPLPP50',DAPLPP50)
          CALL RDSREA ('LPP0',LPP0)
          CALL RDSREA ('LPPMAX',LPPMAX)
          CALL RDINIT (30,40,'VIRUS.DAT')
          CALL RDFREA ('PHDPROD',PHDPROD(1),5,5)
          CLOSE (30,STATUS='DELETE')

*         pointer for living infected larvae with highest rank number
```

```
          IPLUP = 0
*         pointer for living infected larvae with lowest rank number
          IPLLOW = 1
*         number of cadavers
          ICAD = 0
*         pointer for contaminated leaf with highest rank number
          IPCLVS = INCLVS

*         Patch initialization
          IPATLOW = 1
          IPATCUM = INPAT

*         Check for invalid input values
          IF (INPAT.GT.IPMAX) THEN
              STOP'initial number of patches too high!'
          ENDIF
          IF (INCLVS.GT.IPOINM) THEN
              STOP'initial number of contaminated leaves too high!'
          ENDIF

*         Loop over beds
          DO 50 IB = 1,IBEDMAX
*           Initial leaves per plant for each bed
            ILPPBED(IB) = NINT(LPP0 + ((LPPMAX - LPP0)/
     $                   (1. + EXP(-RGRLPP*(DAP(IB) - DAPLPP50)))))
50        CONTINUE

*         Loop over contaminated leaf locations
          DO 60 IP = 1,IPMAX
              IF ((IPATDAT(1,IP).GT.IBEDMAX).OR.
     $            (IPATDAT(2,IP).GT.IROWMAX).OR.
     $            (IPATDAT(3,IP).GT.IPPRMAX)) THEN
                  STOP'initial co-ordinates of patches out of range!'
              ENDIF
60        CONTINUE

*         Loop over contaminated leaf locations
          DO 70 IB = 1,INCLVS
              IF ((INFOCLVS(1,IB).GT.IBEDMAX).OR.
     $            (INFOCLVS(2,IB).GT.IROWMAX).OR.
     $            (INFOCLVS(3,IB).GT.IPPRMAX).OR.
     $            (INFOCLVS(4,IB).GT.ILPPBED(INFOCLVS(1,IB)))) THEN
          STOP'initial co-ordinates of contaminated leaves out of range'
              ENDIF
70        CONTINUE

*         Number of plants per larval foraging domain
          IPLANTL1 = (1 + IHORDIST(1,1) + IHORDIST(2,1))*
     $               (1 + IHORDIST(3,1) + IHORDIST(4,1))
          IPLANTL2 = (1 + IHORDIST(1,2) + IHORDIST(2,2))*
     $               (1 + IHORDIST(3,2) + IHORDIST(4,2))
          IPLANTL3 = (1 + IHORDIST(1,3) + IHORDIST(2,3))*
     $               (1 + IHORDIST(3,3) + IHORDIST(4,3))
          IPLANTL4 = (1 + IHORDIST(1,4) + IHORDIST(2,4))*
     $               (1 + IHORDIST(3,4) + IHORDIST(4,4))
          IPLANTL5 = (1 + IHORDIST(1,5) + IHORDIST(2,5))*
     $               (1 + IHORDIST(3,5) + IHORDIST(4,5))

*         Number of plants in greenhouse
          INPLANT = IROWMAX * IPPRMAX

*         Initialization of total number of polyhedra produced
*         in larvae
          PHDLARTOT = 0
      ENDIF

*     Calculation of SPADIS auxiliary variables
      IF (ITASK.EQ.2) THEN
*         Calculation of number of contaminated leaves in each patch
*         loop over patches
          DO 120 IP = IPATLOW,IPATCUM
*           number of contaminated leaves in current patch set to zero
            ICLVS(IP) = 0
*           number of contaminated leaves per larval foraging domain set to zero
            ICLVSL1(IP) = 0
            ICLVSL2(IP) = 0
            ICLVSL3(IP) = 0
            ICLVSL4(IP) = 0
            ICLVSL5(IP) = 0

*           Initialization of number of contaminated leaves per larval
*           foraging domain per canopy layer
            DO 80 ICL = 1,3
              ICLVSL1CL(IP,ICL) = 0
              ICLVSL2CL(IP,ICL) = 0
              ICLVSL3CL(IP,ICL) = 0
              ICLVSL4CL(IP,ICL) = 0
              ICLVSL5CL(IP,ICL) = 0
80          CONTINUE

*           Calculation of number of leaves per canopy layer
            REST = (ILPP(IP)/3. - INT(ILPP(IP)/3.))
            IF(REST.LT.0.1) THEN
              ILVSCL(IP,1) = INT(ILPP(IP)/3.)
              ILVSCL(IP,2) = INT(ILPP(IP)/3.)
              ILVSCL(IP,3) = INT(ILPP(IP)/3.)
            ENDIF
            IF(REST.LT.0.4.AND.REST.GT.0.3) THEN
              ILVSCL(IP,1) = INT(ILPP(IP)/3.)
              ILVSCL(IP,2) = INT(ILPP(IP)/3.)
              ILVSCL(IP,3) = INT(ILPP(IP)/3.)+1
            ENDIF
            IF(REST.LT.0.7.AND.REST.GT.0.6) THEN
              ILVSCL(IP,1) = INT(ILPP(IP)/3.)
              ILVSCL(IP,2) = INT(ILPP(IP)/3.)+1
              ILVSCL(IP,3) = INT(ILPP(IP)/3.)+1
            ENDIF

*           For each contaminated leaf it is checked whether it is located
*           in a foraging domain of the patch.
            DO 100 IL = 1,IPCLVS
*             Check if contaminated leaf is located in the same bed as the
*             current patch
              IF (INFOCLVS(1,IL).EQ.IPATDAT(1,IP)) THEN

*               Calculation of distance (# plants) between the egg batch and
*               contaminated leaf
                IPPRDIF(IL) = INFOCLVS(3,IL) - IPATDAT(3,IP)
                IROWDIF(IL) = INFOCLVS(2,IL) - IPATDAT(2,IP)

*               When the distance between the contaminated leaf
*               and the egg batch is less than the action radius
*               of L5 larvae, the contaminated leaf is located
*               inside the patch.
                IF (((IPPRDIF(IL).GE.-1*IHORDIST(1,5)).AND.
     $             (IPPRDIF(IL).LE.IHORDIST(2,5))).AND.
     $             ((IROWDIF(IL).GE.-1*IHORDIST(4,5)).AND.
     $             (IROWDIF(IL).LE.IHORDIST(3,5)))) THEN
*                  Number of contaminated leaves in current patch is
*                  raised with 1
                   ICLVS(IP) = ICLVS(IP) + 1

*                  Calculation in which foraging domain the
*                  contaminated leaf is situated using the IHORDIST-array
                   DO 90 IFA = 5,1,-1
                     IF (((IPPRDIF(IL).GE.-1*IHORDIST(1,IFA)).AND.
     $               (IPPRDIF(IL).LE.IHORDIST(2,IFA))).AND.
     $               ((IROWDIF(IL).GE.-1*IHORDIST(4,IFA)).AND.
     $               (IROWDIF(IL).LE.IHORDIST(3,IFA)))) THEN
                      FORAR(IFA) = .TRUE.
                     ELSE
                      FORAR(IFA) = .FALSE.
                     ENDIF
90                 CONTINUE

*                  Calculation in which canopy layer the contaminated
*                  leaf is situated
                   IF (INFOCLVS(4,IL).LE.(ILPP(IP)/3)) THEN
                       ICLCLVS = 1
                   ELSE IF (INFOCLVS(4,IL).LE.(2*ILPP(IP)/3).AND.
     $            INFOCLVS(4,IL).GT.(ILPP(IP)/3)) THEN
                  ICLCLVS = 2
                ELSE IF (INFOCLVS(4,IL).LE.(ILPP(IP)).AND.
     $            INFOCLVS(4,IL).GT.(2*ILPP(IP)/3)) THEN
                  ICLCLVS = 3
                ENDIF

*               Calculation of the number of contaminated leaves in the
*               larval foraging domains and canopy layers. ICLVSL1CL is the
*               number of contaminated leaves per larval foraging domain
*               and canopy layer; ICLVSL1 is number the of contaminated
*               leaves per horizontal larval foraging area; ICLVSNRL1 is
*               an array with the INFOCLVS rank number of the contaminated
*               leaf.
                IF (FORAR(1)) THEN
                 ICLVSL1CL(IP,ICLCLVS) = ICLVSL1CL(IP,ICLCLVS) + 1
                 ICLVSL1(IP) = ICLVSL1(IP) + 1
                 IF (ICLVSL1(IP).GE.ICLVSMAX) THEN
                  WRITE(*,*)'maximal number of contaminated'
                  WRITE(*,*)'leaves in L1 foraging domain reached!'
                  STOP
                 ENDIF
                 ICLVSNRL1(IP,ICLVSL1(IP)) = IL
                ENDIF
                IF (FORAR(2)) THEN
                 ICLVSL2CL(IP,ICLCLVS) = ICLVSL2CL(IP,ICLCLVS) + 1
                 ICLVSL2(IP) = ICLVSL2(IP) + 1
                 IF (ICLVSL2(IP).GE.ICLVSMAX) THEN
                  WRITE(*,*)'maximal number of contaminated'
                  WRITE(*,*)'leaves in L2 foraging domain reached!'
                  STOP
                 ENDIF
                 ICLVSNRL2(IP,ICLVSL2(IP)) = IL
                ENDIF
                IF (FORAR(3)) THEN
                 ICLVSL3CL(IP,ICLCLVS) = ICLVSL3CL(IP,ICLCLVS) + 1
                 ICLVSL3(IP) = ICLVSL3(IP) + 1
                 IF (ICLVSL3(IP).GE.ICLVSMAX) THEN
                  WRITE(*,*)'maximal number of contaminated'
                  WRITE(*,*)'leaves in L3 foraging domain reached!'
                 STOP
                 ENDIF
                 ICLVSNRL3(IP,ICLVSL3(IP)) = IL
                ENDIF
                IF (FORAR(4)) THEN
                 ICLVSL4CL(IP,ICLCLVS) = ICLVSL4CL(IP,ICLCLVS) + 1
                 ICLVSL4(IP) = ICLVSL4(IP) + 1
                 IF (ICLVSL4(IP).GE.ICLVSMAX) THEN
                  WRITE(*,*)'maximal number of contaminated'
                  WRITE(*,*)'leaves in L4 foraging domain reached!'
```

```
                        STOP
                      ENDIF
                      ICLVSNRL4(IP,ICLVSL4(IP)) = IL
                    ENDIF
                    IF (FORAR(5)) THEN
                      ICLVSL5CL(IP,ICLCLVS) = ICLVSL5CL(IP,ICLCLVS) + 1
                      ICLVSL5(IP) = ICLVSL5(IP) + 1
                      IF (ICLVSL5(IP).GE.ICLVSMAX) THEN
                        WRITE(*,*)'maximal number of contaminated'
                        WRITE(*,*)'leaves in L5 foraging domain reached!'
                        STOP
                      ENDIF
                      ICLVSNRL5(IP,ICLVSL5(IP)) = IL
                    ENDIF
                ENDIF
              ENDIF
100         CONTINUE

*           Calculation of the fraction of contaminated leaves per
*           larval foraging domain and canopy layer. Leaves with more
*           than one cadaver have only one location in the contaminated
*           leaf array.
*           Loop over canopy layers
            DO 110 ICL = 1,3
                FCLVSL1(IP,ICL) = REAL(ICLVSL1CL(IP,ICL))/
     $                            REAL(ILVSCL(IP,ICL)*IPLANTL1)
                FCLVSL2(IP,ICL) = REAL(ICLVSL2CL(IP,ICL))/
     $                            REAL(ILVSCL(IP,ICL)*IPLANTL2)
                FCLVSL3(IP,ICL) = REAL(ICLVSL3CL(IP,ICL))/
     $                            REAL(ILVSCL(IP,ICL)*IPLANTL3)
                FCLVSL4(IP,ICL) = REAL(ICLVSL4CL(IP,ICL))/
     $                            REAL(ILVSCL(IP,ICL)*IPLANTL4)
                FCLVSL5(IP,ICL) = REAL(ICLVSL5CL(IP,ICL))/
     $                            REAL(ILVSCL(IP,ICL)*IPLANTL5)
110         CONTINUE
120       CONTINUE

*         Determination of dying locations for newly infected larvae in each
*         patch by vertical, horizontal transmission and by virus application
          DO 130 IP = IPATLOW,IPATCUM
*             L1 larvae that are contaminated by vertical transmission are
*             considered infected as soon as they hatch. Check for array
*             overflow takes place in NEWINF subroutine.
              IF (IHATCHI(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IHATCHI(IP),-99,
     $                      ICLVSNRL1,1,IP,IPATDAT,ILPP,IACTRADL1,
     $                      IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                      INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL1HT(IP).GT.0) THEN
```

```
                CALL NEWINF(IPLUP,IRL1HT(IP),ICLVSL1(IP),
     $                       ICLVSNRL1,1,IP,IPATDAT,ILPP,IACTRADL1,
     $                       IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                       INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL2HT(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IRL2HT(IP),ICLVSL2(IP),
     $                       ICLVSNRL2,2,IP,IPATDAT,ILPP,IACTRADL2,
     $                       IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                       INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL3SHT(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IRL3SHT(IP),ICLVSL3(IP),
     $                       ICLVSNRL3,3,IP,IPATDAT,ILPP,IACTRADL3,
     $                       IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                       INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL3RHT(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IRL3RHT(IP),ICLVSL3(IP),
     $                       ICLVSNRL3,3,IP,IPATDAT,ILPP,IACTRADL3,
     $                       IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                       INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL4SHT(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IRL4SHT(IP),ICLVSL4(IP),
     $                       ICLVSNRL4,4,IP,IPATDAT,ILPP,IACTRADL4,
     $                       IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                       INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL4RHT(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IRL4RHT(IP),ICLVSL4(IP),
     $                       ICLVSNRL4,4,IP,IPATDAT,ILPP,IACTRADL4,
     $                       IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                       INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL5SHTD(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IRL5SHTD(IP),ICLVSL5(IP),
     $                       ICLVSNRL5,5,IP,IPATDAT,ILPP,IACTRADL5,
     $                       IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                       INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL5RHTD(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IRL5RHTD(IP),ICLVSL5(IP),
     $                       ICLVSNRL5,5,IP,IPATDAT,ILPP,IACTRADL5,
     $                       IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                       INFOLIL,INFOCLVS)
              ENDIF
              IF (IRL1S(IP).GT.0) THEN
                CALL NEWINF(IPLUP,IRL1S(IP),0,
     $                       ICLVSNRL1,1,IP,IPATDAT,ILPP,IACTRADL1,
      $                     IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
      $                     INFOLIL,INFOCLVS)
               ENDIF
               IF (IRL2S(IP).GT.0) THEN
                 CALL NEWINF(IPLUP,IRL2S(IP),0,
      $                     ICLVSNRL2,2,IP,IPATDAT,ILPP,IACTRADL2,
      $                     IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
      $                     INFOLIL,INFOCLVS)
               ENDIF
               IF (IRL3SS(IP).GT.0) THEN
                 CALL NEWINF(IPLUP,IRL3SS(IP),0,
      $                     ICLVSNRL3,3,IP,IPATDAT,ILPP,IACTRADL3,
      $                     IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
      $                     INFOLIL,INFOCLVS)
               ENDIF
               IF (IRL3RS(IP).GT.0) THEN
                 CALL NEWINF(IPLUP,IRL3RS(IP),0,
      $                     ICLVSNRL3,3,IP,IPATDAT,ILPP,IACTRADL3,
      $                     IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
      $                     INFOLIL,INFOCLVS)
               ENDIF
               IF (IRL4SS(IP).GT.0) THEN
                 CALL NEWINF(IPLUP,IRL4SS(IP),0,
      $                     ICLVSNRL4,4,IP,IPATDAT,ILPP,IACTRADL4,
      $                     IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
      $                     INFOLIL,INFOCLVS)
               ENDIF
               IF (IRL4RS(IP).GT.0) THEN
                 CALL NEWINF(IPLUP,IRL4RS(IP),0,
      $                     ICLVSNRL4,4,IP,IPATDAT,ILPP,IACTRADL4,
      $                     IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
      $                     INFOLIL,INFOCLVS)
               ENDIF
               IF (IRL5SSD(IP).GT.0) THEN
                 CALL NEWINF(IPLUP,IRL5SSD(IP),0,
      $                     ICLVSNRL5, 5,IP,IPATDAT,ILPP,IACTRADL5,
      $                     IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
      $                     INFOLIL,INFOCLVS)
               ENDIF
               IF (IRL5RSD(IP).GT.0) THEN
                 CALL NEWINF(IPLUP,IRL5RSD(IP),0,
      $                     ICLVSNRL5, 5,IP,IPATDAT,ILPP,IACTRADL5,
      $                     IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
      $                     INFOLIL,INFOCLVS)
               ENDIF
130        CONTINUE

*          Infected larvea that died are shifted from the living infected
*          array INFOLIL to contaminated leaves array INFOCLVS. The pointer
*          of INFOCLVS (=IPCLVS) is raised with 1. The INFOLIL array is filled
*        with nills, hence the living infected larvae does no longer exist.
         DO 140 IP = IPATLOW,IPATCUM
             IF (IDRL1I(IP).GT.0) CALL NEWDIL(IDRL1I(IP),IPLUP,1,IP,
     $                                        PHDPROD,
     $                                        IPCLVS,ICAD,INFOLIL,
     $                                        INFOCLVS,PHDLARTOT)
             IF (IDRL2I(IP).GT.0) CALL NEWDIL(IDRL2I(IP),IPLUP,2,IP,
     $                                        PHDPROD,
     $                                        IPCLVS,ICAD,INFOLIL,
     $                                        INFOCLVS,PHDLARTOT)
             IF (IDRL3I(IP).GT.0) CALL NEWDIL(IDRL3I(IP),IPLUP,3,IP,
     $                                        PHDPROD,
     $                                        IPCLVS,ICAD,INFOLIL,
     $                                        INFOCLVS,PHDLARTOT)
             IF (IDRL4I(IP).GT.0) CALL NEWDIL(IDRL4I(IP),IPLUP,4,IP,
     $                                        PHDPROD,
     $                                        IPCLVS,ICAD,INFOLIL,
     $                                        INFOCLVS,PHDLARTOT)
             IF (IDRL5ID(IP).GT.0) CALL NEWDIL(IDRL5ID(IP),IPLUP,5,IP,
     $                                         PHDPROD,
     $                                         IPCLVS,ICAD,INFOLIL,
     $                                         INFOCLVS,PHDLARTOT)
140        CONTINUE
      ENDIF

*     Output of SPADIS variables
      IF (ITASK.EQ.4) THEN
          CALL OUTDAT(2,20,'CONTLEAF',REAL(IPCLVS))
          CALL OUTDAT(2,20,'ICUMCADAVER',REAL(ICAD))
          CALL OUTDAT(2,20,'PHDLARTOT',PHDLARTOT)

*         Calculaton of the total number of infected larvae per instar
          IPAT1I = 0
          IPAT2I = 0
          IPAT3I = 0
          IPAT4I = 0
          IPAT5I = 0
          DO 150 IP = IPATLOW,IPATCUM
              IPAT1I = IPATL1I(IP) + IPAT1I
              IPAT2I = IPATL2I(IP) + IPAT2I
              IPAT3I = IPATL3I(IP) + IPAT3I
              IPAT4I = IPATL4I(IP) + IPAT4I
              IPAT5I = ITIL5D(IP) + IPAT5I
150       CONTINUE

*         Check if number of living infected larvae in boxcar
*         and in array are equal
          CALL CONTROL(IPAT1I,IPLUP,INFOLIL,1)
          CALL CONTROL(IPAT2I,IPLUP,INFOLIL,2)
          CALL CONTROL(IPAT3I,IPLUP,INFOLIL,3)
          CALL CONTROL(IPAT4I,IPLUP,INFOLIL,4)
          CALL CONTROL(IPAT5I,IPLUP,INFOLIL,5)
      ENDIF

*     Calculation of shift for SPADIS variables
      IF (ITASK.EQ.6) THEN
*         All larvae are removed from the bed after a harvest
          IF (IHARV.GT.0) THEN
              CALL HARV(IHARV,INFOLIL,IPLUP,IPCLVS,INFOCLVS,IPLLOW)
          ENDIF
*         Determination of patch co-ordinates of new patch
          IF (IBATOUT.GT.0.OR.IBATOUTI.GT.0) THEN
              IBEDPREF = 0
              DO 160 IB = 1,IBEDMAX
                  IBEDDEP(IB) = 0
160           CONTINUE
*             Selection of beds with young plants that are preferred for
*             egg deposition
              DO 170 IB = 1,IBEDMAX
                IF (LAIBED(IB).LE.LAIPREF) THEN
                   IBEDPREF = IBEDPREF + 1
                   IBEDDEP(IBEDPREF) = IB
                ENDIF
170           CONTINUE

              DO 180 INEWBAT = 1,(IBATOUT+IBATOUTI)
*                 If there are no young plants in the greenhouse, females
*                 have no preference for oviposition in specific beds
                  IF (IBEDPREF.EQ.0) THEN
                      IPATDAT(1,IPATCUM-INEWBAT+1) = IDRWLOT(1,IBEDMAX)
*                 If there are young plants in the greenhouse, females
*                 prefer plants with a LAI lower or equal than LAIPREF.
                  ELSE
                      IPATDAT(1,IPATCUM-INEWBAT+1) =
     $                IBEDDEP(IDRWLOT(1,IBEDPREF))
                  ENDIF
*                 Location of deposited egg batch in bed is random
                  IPATDAT(2,IPATCUM-INEWBAT+1) = IDRWLOT(1,IROWMAX)
                  IPATDAT(3,IPATCUM-INEWBAT+1) = IDRWLOT(1,IPPRMAX)
180           CONTINUE
          ENDIF
      ENDIF

      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* SUBROUTINE RESET                                                      *
*                                                                       *
* Purpose: Reset locations of infected larvae in INFOLIL array.         *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
* name    type meaning (unit)                                   class *
* ----    ---- ---------------                                  ----- *
* IPLUP    I4  Pointer for infected larvae with highest rank number     *
*              (-)                                                 I  *
* IPLLOW   I4  Pointer for infected larvae with lowest rank number      *
*              (-)                                                 I  *
* INFOLIL  I4  Array containing co-ordinates of infected larvae (-) I  *
*                                                                       *
* Fatal error checks: none                                              *
* warnings:           none                                              *
* Subroutines called: none                                              *
* File usage:         none                                              *
*-----------------------------------------------------------------------*
      SUBROUTINE RESET(IPLUP,IPLLOW,INFOLIL)

      PARAMETER (IPOINM = 1000)
      INTEGER   IPLUP,IPLLOW,INFOLIL,IEMP,IL
      DIMENSION INFOLIL(6,IPOINM)
      SAVE

*     Determining infected larvae in living infected larvae array
*     with lowest IPL number
      DO 10 IL = IPLUP,1,-1
          IF(INFOLIL(5,IL).NE.0) IPLLOW = IL
10    CONTINUE

*     Reset rank number of living infected larvae
      DO 20 IL = 1,IPLUP-IPLLOW+1
          INFOLIL(1,IL) = INFOLIL(1,IPLLOW+IL-1)
          INFOLIL(2,IL) = INFOLIL(2,IPLLOW+IL-1)
          INFOLIL(3,IL) = INFOLIL(3,IPLLOW+IL-1)
          INFOLIL(4,IL) = INFOLIL(4,IPLLOW+IL-1)
          INFOLIL(5,IL) = INFOLIL(5,IPLLOW+IL-1)
          INFOLIL(6,IL) = INFOLIL(6,IPLLOW+IL-1)
20    CONTINUE

*     Reset pointer in order to be able to overwrite old locations
      IEMP = IPLUP - IPLLOW + 2

*     Reset data of empty array numbers
      DO 30 IL = IEMP,IPLUP
          INFOLIL(1,IL) = 0
          INFOLIL(2,IL) = 0
          INFOLIL(3,IL) = 0
          INFOLIL(4,IL) = 0
          INFOLIL(5,IL) = 0
          INFOLIL(6,IL) = 0
30    CONTINUE

*     Reset pointers of infected larvae array
      IPLUP = IPLUP - IPLLOW + 1
      IPLLOW = 1

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* FUNCTION IDRWLOT                                                     *
*                                                                      *
* Purpose: Random drawing of integer value between minimum and         *
*          maximum value.                                              *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* IDRWLOT I4   Stochastically chosen number between minimum and        *
*              maximum value (-)                                   O   *
* IMIN    I4   Help variable for drawing procedure (-)             I   *
* IMAX    I4   Help variable for drawing procedure (-)             I   *
*                                                                      *
* Fatal error checks: IMAX-IMIN < 0                                    *
*                     IMAX < 0                                         *
* warnings:           none                                             *
* Subroutines called: none                                             *
* File usage:         none                                             *
*----------------------------------------------------------------------*
      FUNCTION IDRWLOT(IMIN,IMAX)

      INTEGER IMIN,IMAX,IDIFF
      SAVE

      IF (((IMAX - IMIN).LT.0).OR.(IMAX.LT.0)) THEN
          STOP'IDRWLOT ERROR'
      ENDIF

      IF (IMIN.EQ.0.AND.IMAX.EQ.0) THEN
         IDRWLOT = 0
      ELSE
         IF (IMIN.GT.0) IDIFF = IMAX - IMIN + 1
         IF (IMIN.LT.0.AND.IMAX.GT.0) IDIFF = IMAX - IMIN + 1
         IF (IMIN.LE.0.AND.IMAX.EQ.0) IDIFF = -IMIN
         IF (IMIN.EQ.0.AND.IMAX.GT.0) IDIFF = IMAX
         IDRWLOT = INT(UNIFL()*IDIFF) + IMIN
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE NEWINF                                                    *
*                                                                      *
* Purpose: Determination of dying locations of newly infected         *
*          larvae                                                      *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
* name    type meaning (unit)                                  class *
* ----    ---- ---------------                                 ----- *
* IPLUP    I4  Pointer for infected larvae with highest rank number   *
*              (-)                                               I   *
* ILIL     I4  Help variable for determination of dying location of   *
*              newly infected larvae (-)                         I   *
* IPSOURCE                                                            *
*          I4  Number of contaminated leaves in larval foraging       *
*              domain (#)                                        I   *
* ICLVSNR I4  Array containing rank numbers of contaminated leaves    *
*              per larval foraging domain (-)                    I   *
* INSTAR   I4  Instar of larvae (-)                              I   *
* IP       I4  Pointer for patch number (-)                      I   *
* IPATDAT I4  Array containing co-ordinates of patches in             *
*              greenhouse (-)                                    I   *
* ILPP     I4  Number of leaves per plant in patch (#)           I   *
* IACTRAD I4  Action radius of infected larvae (# plants)        I   *
* IROWMAX I4  Maximum number plant rows in bed (#)               I   *
* IPPRMAX I4  Maximum number of plants per row in bed (#)        I   *
* IHORDIST                                                            *
*          I4  Array containing the shape and location of larval      *
*              foraging domains (# plants)                       I   *
* VERTDIST                                                            *
*          R4  Fraction larvae per canopy layer (-)              I   *
* INFOLIL I4  Array containing co-ordinates of infected larvae (-) O  *
* INFOCLVS                                                            *
*          I4  Array containing co-ordinates of contaminated leaves   *
*              (-)                                               O   *
*                                                                     *
* Fatal error checks: IPLUP > IPOINM                                  *
* warnings:           none                                            *
* Subroutines or functions called:                                    *
*                     IDRWLOT,IDRWLEAF,RESET                          *
* File usage:         none                                            *
*---------------------------------------------------------------------*
      SUBROUTINE NEWINF(IPLUP,ILIL,IPSOURCE,ICLVSNR,
     $                  INSTAR,IP,IPATDAT,ILPP,IACTRAD,
     $                  IROWMAX,IPPRMAX,IHORDIST,VERTDIST,
     $                     INFOLIL,INFOCLVS)

      PARAMETER (IPOINM = 1000)
      PARAMETER (IPMAX = 200)
      PARAMETER (ICLVSMAX = 400)
      INTEGER   IPLUP,INFOLIL,IPSOURCE,INFOCLVS,ILIL,ICLVSNR,
     $          ISOURCE,INSTAR,IP,IL,IPATDAT,ILPP,IACTRAD,
     $          IROWMAX,IPPRMAX,IHORDIST,IROWLOW,IROWUP,IPPRLOW,
     $          IPPRUP
      REAL      VERTDIST
      DIMENSION INFOLIL(6,IPOINM),INFOCLVS(5,IPOINM),
     $          ICLVSNR(IPMAX,ICLVSMAX),IPATDAT(3,IPMAX),ILPP(IPMAX),
     $          IHORDIST(4,5),VERTDIST(3,5)
      SAVE

*     Loop over newly infected larvae
      DO 10 IL = 1,ILIL
*         Pointer for living infected larvae array is raised with 1
          IPLUP = IPLUP + 1

          IF (IPLUP.GT.IPOINM) THEN
              STOP'maximal number of living infected larvae reached!'
          ENDIF

*         Determination of dying location of infected larvae by horizontal
*         transmission
          IF (IPSOURCE.GT.0) THEN
*             Contaminated leaf that is source of infection of newly infected
*             larvae by horizontal transmission is determined randomly
              ISOURCE = IDRWLOT(1,IPSOURCE)

*             Row and plant per row co-ordinate of source
              INFOLIL(2,IPLUP) = INFOCLVS(2,ICLVSNR(IP,ISOURCE))
              INFOLIL(3,IPLUP) = INFOCLVS(3,ICLVSNR(IP,ISOURCE))

*             Deviation range of newly infected larvae from source
              IROWLOW = MAX(1,INFOLIL(2,IPLUP) - IACTRAD,
     $                  IPATDAT(2,IP) - IHORDIST(4,INSTAR))
              IROWUP  = MIN(IROWMAX,INFOLIL(2,IPLUP) + IACTRAD,
     $                  IPATDAT(2,IP) + IHORDIST(3,INSTAR))
              IPPRLOW = MAX(1,INFOLIL(3,IPLUP) - IACTRAD,
     $                  IPATDAT(3,IP) - IHORDIST(1,INSTAR))
              IPPRUP  = MIN(IPPRMAX,INFOLIL(3,IPLUP) + IACTRAD,
     $                  IPATDAT(3,IP) + IHORDIST(2,INSTAR))

*             Dying location of newly infected larvae
              INFOLIL(1,IPLUP) = INFOCLVS(1,ICLVSNR(IP,ISOURCE))
              INFOLIL(2,IPLUP) = IDRWLOT(IROWLOW,IROWUP)
              INFOLIL(3,IPLUP) = IDRWLOT(IPPRLOW,IPPRUP)
              INFOLIL(4,IPLUP) = IDRWLEAF(ILPP(IP),INSTAR,VERTDIST)
              INFOLIL(5,IPLUP) = INSTAR
              INFOLIL(6,IPLUP) = IP
          ENDIF

*         Determination of dying location of larvae infected by vertical
*         transmission
          IF (IPSOURCE.EQ.-99) THEN
*             Row and plant per row co-ordinate of source
              INFOLIL(2,IPLUP) = IPATDAT(2,IP)
              INFOLIL(3,IPLUP) = IPATDAT(3,IP)

*             Deviation range of newly infected larvae from source
              IROWLOW = MAX(1,INFOLIL(2,IPLUP) - IACTRAD)
              IROWUP = MIN(IROWMAX,INFOLIL(2,IPLUP) + IACTRAD)
              IPPRLOW = MAX(1,INFOLIL(3,IPLUP) - IACTRAD)
              IPPRUP = MIN(IPPRMAX,INFOLIL(3,IPLUP) + IACTRAD)

*             Dying location of newly infected larvae.
              INFOLIL(1,IPLUP) = IPATDAT(1,IP)
              INFOLIL(2,IPLUP) = IDRWLOT(IROWLOW,IROWUP)
              INFOLIL(3,IPLUP) = IDRWLOT(IPPRLOW,IPPRUP)
              INFOLIL(4,IPLUP) = IDRWLEAF(ILPP(IP),INSTAR,VERTDIST)
              INFOLIL(5,IPLUP) = INSTAR
              INFOLIL(6,IPLUP) = IP
          ENDIF

*         larvae infected by virus application
          IF (IPSOURCE.EQ.0) THEN
*             Row and plant per row co-ordinate of source
              INFOLIL(2,IPLUP) = IPATDAT(2,IP)
              INFOLIL(3,IPLUP) = IPATDAT(3,IP)

*             Deviation range of newly infected larvae from source
              IROWLOW = MAX(1,INFOLIL(2,IPLUP) - IHORDIST(4,INSTAR))
              IROWUP =  MIN(IROWMAX,INFOLIL(2,IPLUP) +
     $                  IHORDIST(3,INSTAR))
              IPPRLOW = MAX(1,INFOLIL(3,IPLUP) - IHORDIST(1,INSTAR))
              IPPRUP =  MIN(IPPRMAX,INFOLIL(3,IPLUP) +
     $                  IHORDIST(2,INSTAR))

*             Dying location of newly infected larvae.
              INFOLIL(1,IPLUP) = IPATDAT(1,IP)
              INFOLIL(2,IPLUP) = IDRWLOT(IROWLOW,IROWUP)
              INFOLIL(3,IPLUP) = IDRWLOT(IPPRLOW,IPPRUP)
              INFOLIL(4,IPLUP) = IDRWLEAF(ILPP(IP),INSTAR,VERTDIST)
              INFOLIL(5,IPLUP) = INSTAR
              INFOLIL(6,IPLUP) = IP
          ENDIF

*         Check for necessity of reset
          IF (IPLUP.GE.(IPOINM)) THEN
              CALL RESET(IPLUP,IPLLOW,INFOLIL)
          ENDIF
10    CONTINUE
```

```
RETURN
END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE NEWDIL                                                    *
*                                                                      *
* Purpose: Transfer of larvae from living infected larvae array        *
*          to contaminated leaves array.                               *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* ICLVSNR I4  Array containing rank numbers of contaminated leaves     *
*             per larval foraging domain (-)                       I   *
* IPLUP   I4  Pointer for infected larvae with highest rank number     *
*             (-)                                                  I   *
* INSTAR  I4  Instar of larvae (-)                                 I   *
* IP      I4  Pointer for patch number (-)                         I   *
* PHDPROD R4  Array containing polyhedra production per larval         *
*             instar (#)                                           I   *
* IPCLVS  I4  Pointer for number of contaminated leaves (-)        O   *
* ICAD    I4  Number of cadavers in greenhouse (#)                 O   *
* INFOLIL I4  Array containing co-ordinates of infected larvae (-) O   *
* INFOCLVS                                                             *
*         I4  Array containing co-ordinates of contaminated leaves     *
*             (-)                                                  O   *
* PHDLARTOT                                                            *
*         R4  Total number of polyhedra produced in larvae (#)     O   *
*                                                                      *
* Fatal error checks: IPCLVS > IPOINM                                  *
* warnings:           none                                             *
* Subroutines called: none                                             *
* File usage:         none                                             *
*----------------------------------------------------------------------*
      SUBROUTINE NEWDIL(ICLVSNR,IPLUP,INSTAR,IP,PHDPROD,
     $                      IPCLVS,ICAD,INFOLIL,INFOCLVS,PHDLARTOT)

      PARAMETER (IPOINM = 1000)
      INTEGER   ICLVSNR,IPLUP,INFOLIL,INFOCLVS,IPCLVS,IL,INSTAR,IP,
     $          IDOUBLE,IOLDLAR,ICAD
      REAL      PHDLARTOT,PHDPROD
      DIMENSION INFOLIL(6,IPOINM),INFOCLVS(5,IPOINM),PHDPROD(5)
      SAVE

*     Loop over newly contaminated leaf
      DO 30 ID = 1,ICLVSNR
*         Determining oldest infected larvae of actual patch in
*         living infected larvae array (larvae with lowest IPL number).
          DO 10 IL = IPLUP,1,-1
              IF((INFOLIL(5,IL).EQ.INSTAR).AND.
     $          (INFOLIL(6,IL).EQ.IP)) THEN
                 IOLDLAR = IL
              ENDIF
```

```
10          CONTINUE

*           Check for more than one cadaver per leaf
            IDOUBLE = 0
            DO 20 IL = 1,IPCLVS
              IF ((INFOCLVS(1,IL).EQ.INFOLIL(1,IOLDLAR)).AND.
     $            (INFOCLVS(2,IL).EQ.INFOLIL(2,IOLDLAR)).AND.
     $            (INFOCLVS(3,IL).EQ.INFOLIL(3,IOLDLAR)).AND.
     $            (INFOCLVS(4,IL).EQ.INFOLIL(4,IOLDLAR))) THEN
                  IDOUBLE = 1
               ENDIF
20          CONTINUE

*           Cadaver pointer is raised with 1
            ICAD = ICAD + 1

            IF (IDOUBLE.NE.1) THEN
*               Contaminated leaves pointer is raised with 1
                IPCLVS = IPCLVS + 1

                IF (IPCLVS.GT.IPOINM) THEN
                    STOP'maximal number of contaminated leaves reached!'
                ENDIF

*               Larval position is written to cadaver array
                INFOCLVS(1,IPCLVS) = INFOLIL(1,IOLDLAR)
                INFOCLVS(2,IPCLVS) = INFOLIL(2,IOLDLAR)
                INFOCLVS(3,IPCLVS) = INFOLIL(3,IOLDLAR)
                INFOCLVS(4,IPCLVS) = INFOLIL(4,IOLDLAR)
                INFOCLVS(5,IPCLVS) = INFOLIL(5,IOLDLAR)
            ENDIF

*           Larval position is cleared from living infected larvae array
            INFOLIL(1,IOLDLAR) = 0
            INFOLIL(2,IOLDLAR) = 0
            INFOLIL(3,IOLDLAR) = 0
            INFOLIL(4,IOLDLAR) = 0
            INFOLIL(5,IOLDLAR) = 0
            INFOLIL(6,IOLDLAR) = 0

*           The number of polyhedra produced in larvae is raised with the
*           newly produced polyhedra
            PHDLARTOT = PHDLARTOT + PHDPROD(INSTAR)
30      CONTINUE

        RETURN
        END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE CONTROL                                                   *
*                                                                      *
* Purpose: Every timestep the number of infected larvae in the boxcars *
*          and in the living infected larvae array are compared for    *
*          every instar. If these numbers are not equal the program is *
*          terminated.                                                 *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                     class *
* ----   ---- ---------------                                    ----- *
* ITOTAL  I4  Help variable for population totals (#)              I   *
* IPLUP   I4  Pointer for infected larvae with highest rank number     *
*             (-)                                                  I   *
* INFOLIL I4  Array containing co-ordinates of infected larvae (-) I   *
* INSTAR  I4  Instar of larvae (-)                                 I   *
*                                                                      *
* Fatal error checks: none                                             *
* warnings:           none                                             *
* Subroutines called: none                                             *
* File usage:         none                                             *
*----------------------------------------------------------------------*
      SUBROUTINE CONTROL(ITOTAL,IPLUP,INFOLIL,INSTAR)

      PARAMETER (IPOINM = 1000)
      INTEGER   ITOTAL,IPLUP,INFOLIL,IL,INSTAR,ITOTLIL
      DIMENSION INFOLIL(6,IPOINM)
      SAVE

*     Determination of number of infected larvae in living infected larvae
*     array
      ITOTLIL = 0
      DO 10 IL = IPLUP,1,-1
          IF(INFOLIL(5,IL).EQ.INSTAR) THEN
             ITOTLIL = ITOTLIL + 1
          ENDIF
10    CONTINUE

*     No error message because infected larvae can die by attrition
*     in current model version.
      IF (ITOTLIL.NE.ITOTAL) THEN
*       Possibility to send error message to sceen
      ENDIF

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* FUNCTION IDRWLEAF                                                    *
*                                                                      *
* Purpose: Stochastic determination of dying location of infected     *
*          larvae on leaf weighed by vertical larvae distribution in  *
*          crop.                                                       *
*                                                                      *
* Fatal error checks: ILPP < 0                                         *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
* name   type meaning (unit)                                   class *
* ----   ---- ---------------                                  ----- *
* IDRWLEAF                                                             *
*         I4  Stochastically chosen leaf number (-)               O  *
* ILPP    I4  Number of leaves per plant in patch (#)             I  *
* INSTAR  I4  Instar of larvae (-)                                I  *
* VERTDIST                                                             *
*         R4  Fraction larvae per canopy layer (-)                I  *
*                                                                      *
* Fatal error checks: ILPP < 0                                         *
* warnings:           none                                             *
* Subroutines called: none                                             *
* File usage:         none                                             *
*----------------------------------------------------------------------*
      FUNCTION IDRWLEAF(ILPP,INSTAR,VERTDIST)

      INTEGER   ILPP,INSTAR,ILAYER1,ILAYER2,ILAYER3,IL,IHIT
      REAL      VERTDIST,SUMW,CUM,LOT
      DIMENSION VERTDIST(3,5)
      SAVE

      IF ((ILPP).LE.0) STOP'IDRWLEAF error!'

*     Calculation of number of leaves per canopy layer
      ILAYER1 = INT(ILPP/3)
      ILAYER2 = INT(2*ILPP/3) - ILAYER1
      ILAYER3 = ILPP - ILAYER2 - ILAYER1

*     Weighed sum of number of leaves per leaf layer times fraction of
*     larvae per canopy layer
      SUMW = ILAYER1*VERTDIST(1,INSTAR) + ILAYER2*VERTDIST(2,INSTAR) +
     $       ILAYER3*VERTDIST(3,INSTAR)

*     drawing lot
      LOT = UNIFL()*SUMW

      CUM = 0.
      IHIT = 0
*     Check for a hit in lowest canopy layer
      DO 10 IL = 1,ILAYER1
         CUM = CUM + VERTDIST(1,INSTAR)
```

```
          IF (CUM.GE.LOT.AND.IHIT.EQ.0) THEN
              IDRWLEAF = IL
              IHIT = 1
          ENDIF
10    CONTINUE

*     Check for a hit in middle canopy layer
      DO 20 IL = ILAYER1+1,ILAYER1+ILAYER2
          CUM = CUM + VERTDIST(2,INSTAR)
          IF (CUM.GE.LOT.AND.IHIT.EQ.0) THEN
              IDRWLEAF = IL
              IHIT = 1
          ENDIF
20    CONTINUE

*     Check for a hit in upper canopy layer
      DO 30 IL = ILAYER1+ILAYER2+1,ILAYER1+ILAYER2+ILAYER3
          CUM = CUM + VERTDIST(3,INSTAR)
          IF (CUM.GE.LOT.AND.IHIT.EQ.0) THEN
              IDRWLEAF = IL
              IHIT = 1
          ENDIF
30    CONTINUE

      RETURN
      END
```

```
*---------------------------------------------------------------------*
* SUBROUTINE HARV                                                      *
*                                                                      *
* Purpose: Simulation of harvest of crop. The crop as well as the      *
*          uninfected and infected larvae and cadavers are removed.    *
*          New cuttings are planted.                                   *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* IHARV   I4   Flag for harvest (-)                                I   *
* INFOLIL I4   Array containing co-ordinates of infected larvae (-) O  *
* IPLUP   I4   Pointer for infected larvae with highest rank number    *
*              (-)                                                 O   *
* IPCLVS  I4   Pointer for number of contaminated leaves (-)       O   *
* INFOCLVS                                                             *
*         I4   Array containing co-ordinates of contaminated leaves    *
*              (-)                                                 O   *
* IPLLOW  I4   Pointer for infected larvae with lowest rank number     *
*              (-)                                                 O   *
*                                                                      *
* Fatal error checks: none                                             *
* warnings:           none                                             *
* Subroutines called: none                                             *
* File usage:         none                                             *
*---------------------------------------------------------------------*
      SUBROUTINE HARV(IHARV,
     $                    INFOLIL,IPLUP,IPCLVS,INFOCLVS,IPLLOW)

      PARAMETER (IPOINM = 1000)
      INTEGER   IPLUP,INFOLIL,INEWNR,IL,IPCLVS,INFOCLVS,IPLLOW
      DIMENSION INFOLIL(6,IPOINM), INFOCLVS(5,IPOINM)
      SAVE

*     Set locations of infected larvae in harvested bed to zero
      DO 10 IL = 1,IPLUP
          IF (INFOLIL(1,IL).EQ.IHARV) THEN
             INFOLIL(1,IL) = 0
             INFOLIL(2,IL) = 0
             INFOLIL(3,IL) = 0
             INFOLIL(4,IL) = 0
             INFOLIL(5,IL) = 0
             INFOLIL(6,IL) = 0
          ENDIF
10    CONTINUE

*     Reset locations of infected larvae in other beds
      INEWNR = 1
      DO 20 IL = 1,IPLUP
          IF (INFOLIL(1,IL).GT.0) THEN
```

```
             INFOLIL(1,INEWNR) = INFOLIL(1,IL)
             INFOLIL(2,INEWNR) = INFOLIL(2,IL)
             INFOLIL(3,INEWNR) = INFOLIL(3,IL)
             INFOLIL(4,INEWNR) = INFOLIL(4,IL)
             INFOLIL(5,INEWNR) = INFOLIL(5,IL)
             INFOLIL(6,INEWNR) = INFOLIL(6,IL)
             INEWNR = INEWNR + 1
          ENDIF
20    CONTINUE

      DO 30 IL = INEWNR,IPOINM
             INFOLIL(1,IL) = 0
             INFOLIL(2,IL) = 0
             INFOLIL(3,IL) = 0
             INFOLIL(4,IL) = 0
             INFOLIL(5,IL) = 0
             INFOLIL(6,IL) = 0
30    CONTINUE

*     Reset pointers of infected larvae array
      IPLLOW = 1
      IPLUP = INEWNR - 1

*     Set locations of cadavers in harvested bed to zero
      DO 40 IL = 1,IPCLVS
          IF (INFOCLVS(1,IL).EQ.IHARV) THEN
              INFOCLVS(1,IL) = 0
              INFOCLVS(2,IL) = 0
              INFOCLVS(3,IL) = 0
              INFOCLVS(4,IL) = 0
              INFOCLVS(5,IL) = 0
          ENDIF
40    CONTINUE

*     Reset locations of cadavers in other beds
      INEWNR = 1
      DO 50 IL = 1,IPCLVS
          IF (INFOCLVS(1,IL).GT.0) THEN
             INFOCLVS(1,INEWNR) = INFOCLVS(1,IL)
             INFOCLVS(2,INEWNR) = INFOCLVS(2,IL)
             INFOCLVS(3,INEWNR) = INFOCLVS(3,IL)
             INFOCLVS(4,INEWNR) = INFOCLVS(4,IL)
             INFOCLVS(5,INEWNR) = INFOCLVS(5,IL)

             INEWNR = INEWNR + 1
          ENDIF
50    CONTINUE

*     Reset pointers of contaminated leaves array
      IPCLVS = INEWNR - 1
      DO 60 IL = INEWNR,IPOINM
             INFOCLVS(1,IL) = 0
             INFOCLVS(2,IL) = 0
             INFOCLVS(3,IL) = 0
             INFOCLVS(4,IL) = 0
             INFOCLVS(5,IL) = 0
60    CONTINUE

      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* SUBROUTINE DRWMORT                                                    *
*                                                                       *
* Purpose: Stochastic determination of infection process responsible  *
*          for the death of larvae, weighed by the relative infection  *
*          rates of these processes (attrition, horizontal            *
*          transmission and infection by ingestion of sprayed         *
*          polyhedra).                                                 *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
* name    type meaning (unit)                                  class *
* ----    ---- ---------------                                 ----- *
* IN       I4  Number of boxcars per boxcar train (#)            I  *
* RLATR    R4  Attrition difference per timestep (#)             I  *
* RLINFS   R4  Infection difference by ingestion of sprayed          *
*              polyhedra of larvae in boxcar per timestep (#)    I  *
* RLINFHT R4   Infection difference by horizontal transmission of    *
*              larvae in boxcar per timestep (#)                 I  *
* IREDATR I4   Attrition difference of individuals in boxcar per     *
*              timestep (#)                                      O  *
* IREDINFRS                                                          *
*          I4  Infection difference by ingestion of sprayed          *
*              polyhedra of individuals in boxcar per timestep (#)  O  *
* IREDINFRHT                                                         *
*          I4  Infection difference by horizontal transmission of    *
*              individuals in boxcar per timestep (#)            O  *
*                                                                     *
* Fatal error checks: none                                            *
* warnings:           none                                            *
* Subroutines called: none                                            *
* File usage:         none                                            *
*---------------------------------------------------------------------*
      SUBROUTINE DRWMORT(IN,RLATR,RLINFS,RLINFHT,
     $                         IREDATR,IREDINFRS,IREDINFRHT)

      INTEGER IN,IREDATR,IREDINFRHT,IREDINFRS,I,IATR,
     $        INFRS,INFRHT,IHIT
      REAL    RLATR,RLINFS,RLINFHT,CUM,LOT,SUMW
      SAVE

*     IREDATR,IREDINFRS and IREDINFRHT are both input as output
*     variabels and have to be renamed and initialized.
      IATR = IREDATR
      INFRS = IREDINFRS
      INFRHT = IREDINFRHT
      IREDATR = 0
      IREDINFRS = 0
      IREDINFRHT = 0

*     Loop over larvae in boxcar
```

```
      DO 10 I = 1,IN
*         Calculation of weighed sum of real value of mortality rate
*         per mortality process multiplied by original reduction
          SUMW = RLATR*IATR+RLINFS*INFRS+RLINFHT*INFRHT
*         drawing lot
          LOT = SUMW*UNIFL()

          CUM = 0.
          IHIT = 0
*         Check for a hit for attrition
          CUM = CUM + RLATR*IATR
          IF (LOT.LE.CUM.AND.IATR.NE.0) THEN
             IREDATR = IREDATR + 1
             IATR = IATR - 1
             IHIT = 1
          ENDIF

*         Check for a hit for infection by ingestion of sprayed polyhedra
          CUM = CUM + RLINFS*INFRS
          IF (LOT.LE.CUM.AND.LOT.GT.IATR*RLATR.AND.
     $         IHIT.NE.1) THEN
              IREDINFRS = IREDINFRS + 1
              INFRS = INFRS - 1
              IHIT = 1
          ENDIF

*         Check for a hit for infection by horizontal transmission
          CUM = CUM + RLINFHT*INFRHT
          IF (LOT.LE.CUM.AND.LOT.GT.(RLATR*IATR + RLINFS*INFRS)
     $         .AND.IHIT.NE.1) THEN
              IREDINFRHT = IREDINFRHT + 1
              INFRHT = INFRHT - 1
              IHIT = 1
           ENDIF
10    CONTINUE

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE ACCOUNT                                                   *
*                                                                      *
* Purpose: Calculation of population totals in and outside             *
*          patches.                                                    *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* IPATCUM I4   Cumulative number of patches in greenhouse (#)      I   *
* IPATEGG I4   Number of egg batches in boxcar train per patch (#) I   *
* IPATL1  I4   Number of L1 larvae in boxcar train per patch (#)   I   *
* IPATL2  I4   Number of L2 larvae in boxcar train per patch (#)   I   *
* IPATL3S I4   Number of susceptible L3 larvae in boxcar train per     *
*              patch (#)                                           I   *
* IPATL3R I4   Number of resistant L3 larvae in boxcar train per       *
*              patch (#)                                           I   *
* IPATL4S I4   Number of susceptible L4 larvae in boxcar train per     *
*              patch (#)                                           I   *
* IPATL4R I4   Number of resistant L4 larvae in boxcar train per       *
*              patch (#)                                           I   *
* IPATL5S I4   Number of susceptible L5 larvae in boxcar train per     *
*              patch (#)                                           I   *
* IPATL5R I4   Number of resistant L5 larvae in boxcar train per       *
*              patch (#)                                           I   *
* IPATEGGI                                                             *
*         I4   Number of contaminated egg batches in boxcar train      *
*              per patch (#)                                       I   *
* IPATL1I I4   Number of infected L1 larvae in boxcar train per        *
*              patch (#)                                           I   *
* IPATL2I I4   Number of infected L2 larvae in boxcar train per        *
*              patch (#)                                           I   *
* IPATL3I I4   Number of infected L3 larvae in boxcar train per        *
*              patch (#)                                           I   *
* IPATL4I I4   Number of infected L4 larvae in boxcar train per        *
*              patch (#)                                           I   *
* ITIL5S  I4   Number of sublethally infected L5 larvae per boxcar     *
*              (#)                                                 I   *
* ITIL5D  I4   Number of lethally infected L5 larvae per boxcar (#) I  *
* ITOTPUP I4   Number of pupea in boxcar train (#)                 I   *
* ITOTMAL I4   Number of male moths in boxcar train (#)            I   *
* ITOTFEM I4   Number of female moths in boxcar train (#)          I   *
* ITOTPUPI                                                             *
*         I4   Number of sublethally infected pupea in boxcar train    *
*              (#)                                                 I   *
* ITOTMALI                                                             *
*         I4   Number of sublethally infected male moths in boxcar     *
*              train (#)                                           I   *
* ITOTFEMI                                                             *
*         I4   Number of sublethally infected female moths in          *
*           boxcar train (#)                                   I  *
* IPATLOW I4  Pointer for lowest patch number containing larvae (-) O  *
*                                                                   *
* Fatal error checks: none                                          *
* warnings:           none                                          *
* Subroutines called:                                               *
* from TTUTIL:        OUTDAT                                        *
* File usage:         none                                          *
*-------------------------------------------------------------------*
      SUBROUTINE ACCOUNT(IPATCUM,IPATEGG,IPATL1,IPATL2,IPATL3S,
     $                   IPATL3R,IPATL4S,IPATL4R,IPATL5S,IPATL5R,
     $                   IPATEGGI,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $                   ITIL5S,ITIL5D,ITOTPUP,ITOTMAL,ITOTFEM,
     $                   ITOTPUPI,ITOTMALI,ITOTFEMI,
     $                       IPATLOW)

      PARAMETER (IPMAX = 200)
      INTEGER   IP,ITOT,ITOTIP,ITOTOP,ILIP,IPATEGG,IPATL1,
     $          IPATL2,IPATL3S,IPATL3R,IPATL4S,IPATL4R,IPATL5S,
     $          IPATL5R,IPATEGGI,IPATL1I,IPATL2I,IPATL3I,IPATL4I,
     $          ITIL5S,ITIL5D,ITOTPUP,ITOTMAL,ITOTFEM,ITOTPUPI,
     $          ITOTMALI,ITOTFEMI,ITOTEGG,ITOTL1,ITOTL2,ITOTL3,
     $          ITOTL4,ITOTL5,ITOTEGGI,ITOTL1I,ITOTL2I,ITOTL3I,
     $          ITOTL4I,ITOTL5S,ITOTL5D
      DIMENSION IPATEGG(IPMAX),IPATL1(IPMAX),IPATL2(IPMAX),
     $          IPATL3S(IPMAX),IPATL3R(IPMAX),IPATL4S(IPMAX),
     $          IPATL4R(IPMAX),IPATL5S(IPMAX),IPATL5R(IPMAX),
     $          IPATEGGI(IPMAX),IPATL1I(IPMAX),IPATL2I(IPMAX),
     $          IPATL3I(IPMAX),IPATL4I(IPMAX),ITIL5S(IPMAX),
     $          ITIL5D(IPMAX)
      SAVE

*     Initialization cumulative number eggs and larvae in patches and number
*     of damaged plants
      ITOTIP = 0
      ITOTEGG = 0
      ITOTL1 = 0
      ITOTL2 = 0
      ITOTL3 = 0
      ITOTL4 = 0
      ITOTL5 = 0
      ITOTEGGI = 0
      ITOTL1I = 0
      ITOTL2I = 0
      ITOTL3I = 0
      ITOTL4I = 0
      ITOTL5S = 0
      ITOTL5D = 0

*     Calculation of total living uninfected and infected individuals
*     in patches and the number of damaged plants per patch
      DO 10 IP = IPATLOW,IPATCUM
          ILIP = IPATEGG(IP)+IPATL1(IP)+IPATL2(IP)+IPATL3S(IP)+
     $           IPATL3R(IP)+IPATL4S(IP)+IPATL4R(IP)+IPATL5S(IP)+
     $           IPATL5R(IP)+IPATEGGI(IP)+IPATL1I(IP)+IPATL2I(IP)+
     $           IPATL3I(IP)+IPATL4I(IP)+ITIL5S(IP)+ITIL5D(IP)
          ITOTIP = ITOTIP + ILIP

          ITOTEGG = ITOTEGG + IPATEGG(IP)
          ITOTL1 = ITOTL1 + IPATL1(IP)
          ITOTL2 = ITOTL2 + IPATL2(IP)
          ITOTL3 = ITOTL3 + IPATL3S(IP) + IPATL3R(IP)
          ITOTL4 = ITOTL4 + IPATL4S(IP) + IPATL4R(IP)
          ITOTL5 = ITOTL5 + IPATL5S(IP) + IPATL5R(IP)
          ITOTEGGI = ITOTEGG + IPATEGGI(IP)
          ITOTL1I = ITOTL1I + IPATL1I(IP)
          ITOTL2I = ITOTL2I + IPATL2I(IP)
          ITOTL3I = ITOTL3I + IPATL3I(IP)
          ITOTL4I = ITOTL4I + IPATL4I(IP)
          ITOTL5S = ITOTL5S + ITIL5S(IP)
          ITOTL5D = ITOTL5D + ITIL5D(IP)
10    CONTINUE


*     Calculation of total living uninfected and infected individuals
*     outside patches
      ITOTOP = 0
      ITOTOP = ITOTPUP + ITOTMAL + ITOTFEM + ITOTPUPI + ITOTMALI +
     $         ITOTFEMI

*     Calculation of total living individuals in and outside patches
      ITOT = ITOTIP + ITOTOP

*     Patches without larvae don't have to be simulated anymore. If
*     a patch is empty, the IPATLOW flag is set to 1.
      ILIP = IPATEGG(IPATLOW)+IPATL1(IPATLOW)+IPATL2(IPATLOW)+
     $       IPATL3S(IPATLOW)+IPATL3R(IPATLOW)+IPATL4S(IPATLOW)+
     $       IPATL4R(IPATLOW)+IPATL5S(IPATLOW)+IPATL5R(IPATLOW)+
     $       IPATEGGI(IPATLOW)+IPATL1I(IPATLOW)+IPATL2I(IPATLOW)+
     $       IPATL3I(IPATLOW)+IPATL4I(IPATLOW)+ITIL5S(IPATLOW)+
     $       ITIL5D(IPATLOW)

*     Patch is empty and will no longer be simulated
      IF (ILIP.EQ.0.AND.IPATLOW.LE.IPATCUM) THEN
          IPATLOW = IPATLOW + 1
      ENDIF

*     Output of population totals
      CALL OUTDAT(2,20,'ITOTEGG',REAL(ITOTEGG))
      CALL OUTDAT(2,20,'ITOTL1',REAL(ITOTL1))
CALL OUTDAT(2,20,'ITOTL2',REAL(ITOTL2))
CALL OUTDAT(2,20,'ITOTL3',REAL(ITOTL3))
CALL OUTDAT(2,20,'ITOTL4',REAL(ITOTL4))
CALL OUTDAT(2,20,'ITOTL5',REAL(ITOTL5))
CALL OUTDAT(2,20,'ITOTPUP',REAL(ITOTPUP))
CALL OUTDAT(2,20,'ITOTMAL',REAL(ITOTMAL))
CALL OUTDAT(2,20,'ITOTFEM',REAL(ITOTFEM))
CALL OUTDAT(2,20,'ITOTEGGI',REAL(ITOTEGGI))
CALL OUTDAT(2,20,'ITOTL1I',REAL(ITOTL1I))
CALL OUTDAT(2,20,'ITOTL2I',REAL(ITOTL2I))
CALL OUTDAT(2,20,'ITOTL3I',REAL(ITOTL3I))
CALL OUTDAT(2,20,'ITOTL4I',REAL(ITOTL4I))
CALL OUTDAT(2,20,'ITOTL5S',REAL(ITOTL5S))
CALL OUTDAT(2,20,'ITOTL5D',REAL(ITOTL5D))
CALL OUTDAT(2,20,'ITOTPUPI',REAL(ITOTPUPI))
CALL OUTDAT(2,20,'ITOTMALI',REAL(ITOTMALI))
CALL OUTDAT(2,20,'ITOTFEMI',REAL(ITOTFEMI))
CALL OUTDAT(2,20,'ITOTIP',REAL(ITOTIP))
CALL OUTDAT(2,20,'ITOTOP',REAL(ITOTOP))
CALL OUTDAT(2,20,'ITOT',REAL(ITOT))

RETURN
END
```

```
*-----------------------------------------------------------------------*
* SUBROUTINE DEPOSIT                                                    *
*                                                                       *
* Purpose: Simulation of egg batch size and fraction of infected        *
*          eggs in contaminated egg batch.                              *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
* name    type meaning (unit)                                    class *
* ----    ---- ---------------                                   ----- *
* IBATOUT I4   Number of batches in oviduct flowing out IBATCH          *
*              boxcar train (#)                                    I   *
* IBATOUTI                                                              *
*         I4   Number of batches in infected oviduct flowing out        *
*              IBATCHI boxcar train (#)                            I   *
* FX      R4   Cumulative frequency distribution of egg batch size      *
*              (-)                                                 I   *
* X       R4   X-axis of cumulative frequency distribution of egg       *
*              batch size (-)                                      I   *
* ICBATSIZE                                                             *
*         I4   Critical size of egg batch to be a large egg batch       *
*              (-)                                                 I   *
* IPATCUM I4   Cumulative number of patches in greenhouse (#)      I   *
* FCBATS  R4   Fraction contaminated small eggbatches deposited by      *
*              sublethally infected moths (-)                      I   *
* FCBATL  R4   Fraction contaminated large eggbatches deposited by      *
*              sublethally infected moths (-)                      I   *
* FCEGGS  R4   Maximum fraction contaminated eggs in small egg          *
*              batches (-)                                         I   *
* FCEGGL  R4   Maximum fraction contaminated eggs in large egg          *
*              batches (-)                                         I   *
* ISTOCH  I4   Switch for stochasticy in egg batch size (-)        I   *
* IEGG    I4   Number of eggs per boxcar per patch (#)             O   *
* IEGGI   I4   Number of contaminated eggs per boxcar per patch (#) O  *
*                                                                       *
* Fatal error checks: none                                              *
* warnings:           none                                              *
* Subroutines called: none                                              *
* File usage:         none                                              *
*-----------------------------------------------------------------------*
      SUBROUTINE DEPOSIT(IBATOUT,IBATOUTI,FX,X,ICBATSIZE,
     $                   IPATCUM,FCBATS,FCBATL,FCEGGS,FCEGGL,ISTOCH,
     $                       IEGG,IEGGI)

      PARAMETER (IPMAX = 200)
      INTEGER   IBATOUT,IBATOUTI,IB,IHIT,I,IEGG,IEGGI,INFEGG,
     $          IPATCUM,ICBATSIZE,ISTOCH
      REAL      LOT,FX,X,FCBATS,FCBATL,FCEGGS,FCEGGL
      DIMENSION IEGG(IPMAX,0:4),IEGGI(IPMAX,0:4),FX(110),X(110)
      SAVE
```

```
*     Loop over uncontaminated egg batches
      DO 20 IB = 1,IBATOUT
*         Number of eggs in egg batch is determined stochastically from
*         a negative binomial distribution
          IF (ISTOCH.EQ.0) THEN
              IEGG(IPATCUM-IB+1,0) = 35
          ENDIF
          IF (ISTOCH.EQ.1) THEN
              IHIT = 0
              LOT = UNIFL()
              DO 10 I = 1,110
                  IF ((LOT.LE.FX(I)).AND.(IHIT.EQ.0)) THEN
                      IEGG(IPATCUM-IB+1,0) = INT(X(I))
                      IHIT = 1
                  ENDIF
10            CONTINUE
          ENDIF
20    CONTINUE

*     Calculation of contaminated egg batches by vertical transmission.
*     Small egg batches generally contain only uninfected larvae. If there
*     are infected larvae in these batches, their number is less than
*     20% of the egg batch size. Large egg batches often contain one or more
*     infected larvae. The distribution of number of contaminated eggs over
*     large egg batches is more ore less uniform (Bianchi et al, 1999).
*     ICBATSIZE is the critical batch size that determines whether an egg batch
is small
*     or large. FINFEGG is the maximal fraction of contaminated eggs per batch.
*     Loop over contaminated egg batches
      DO 40 IB = IBATOUT+1,IBATOUT+IBATOUTI
*         Number of eggs in contaminated egg batch is determined
*         stochastically from a negative binomial distribution
          IF (ISTOCH.EQ.0) THEN
              IEGG(IPATCUM-IB+1,0) = 35
          ENDIF
          IF (ISTOCH.EQ.1) THEN
              IHIT = 0
              LOT = UNIFL()
              DO 30 I = 1,110
                  IF ((LOT.LE.FX(I)).AND.(IHIT.EQ.0)) THEN
                      IEGG(IPATCUM-IB+1,0) = INT(X(I))
                      IHIT = 1
                  ENDIF
30            CONTINUE
          ENDIF
*         Distribution of contaminated eggs in small egg batches.
          IF (IEGG(IPATCUM-IB+1,0).LE.ICBATSIZE) THEN
              LOT = UNIFL()
*             Egg batch contains one or more contaminated eggs
              IF (LOT.LE.FCBATS) THEN
```

```
*                 Number of uninfected and contaminated eggs
                  INFEGG = NINT(UNIFL()*FCEGGS*IEGG(IPATCUM-IB+1,0))
                  IF (INFEGG.EQ.0) INFEGG = 1
                  IEGGI(IPATCUM-IB+1,0) = INFEGG
                  IEGG(IPATCUM-IB+1,0) = IEGG(IPATCUM-IB+1,0)-INFEGG
              ENDIF
*         Distribution of contaminated eggs in large egg batches.
          ELSE
              LOT = UNIFL()
*             Egg batch contains one or more contaminated eggs
              IF (LOT.LE.FCBATL) THEN
*                 Number of uninfected and contaminated eggs
                  INFEGG = NINT(UNIFL()*FCEGGL*IEGG(IPATCUM-IB+1,0))
                  IF (INFEGG.EQ.0) INFEGG = 1
                  IEGGI(IPATCUM-IB+1,0) = INFEGG
                  IEGG(IPATCUM-IB+1,0) = IEGG(IPATCUM-IB+1,0)-INFEGG
              ENDIF
          ENDIF
40    CONTINUE

      RETURN
      END
```

Other volumes in this series:

No. 1	The FSE system for crop simulation, version 2.1	D.W.G. van Kraalingen
No. 2	Description and user guide of GOAL-QUASI: an IMGLP model for the exploration of future land use	M.K. van Ittersum, R.J. Hijmans & D. Scheele
No. 3	Remote sensing subroutines in crop growth simulation models	B.A.M. Bouman
No. 4	LINTUL-POTATO. A simulation model for potential production of different potato cultivars over a range of climates	P.L. Kooman
No. 5	The Fortran Simulation Translator: FST version 2.0. Introduction and reference manual	C. Rappoldt & D.W.G. van Kraalingen
No. 6	Models in action. Proceeding of a seminar series	A. Stein, F.W.T. Penning de Vries & P.J. Schotman
No. 7	Description of the growth model LINGRA as implemented in CGMS	B.A.M. Bouman, A.H.C.M. Schapendonk, W. Stol & D.W.G. van Kraalingen
No. 8	The Conversion of Land Use and its Effects (CLUE-CR): a regression based model applied to Costa Rica (Pascal version 1.2)	J.M. Schoorl, A. Veldkamp & L.O. Fresco
No. 9	Description, parameterization and user guide of LINTUL-BRASNAP 1.1. A crop growth model of winter oilseed rape *(Brassica napus L.)*	B. Habekotté
No. 10	Rotation models for ecological farming	M.C. Plentinger & F.W.T. Penning de Vries (Eds.)
No. 11	Evapotranspiration modules for crop growth simulation	D.W.G. van Kraalingen & W. Stol
No. 12	Data in action. Proceedings of a seminar series 1996/1997	A. Stein, F.W.T. Penning de Vries & J.W. Schut
No. 13	Harvesting the suns's energy using agro-ecosystems	S. Nonhebel
No. 14	SUCROS97: Simulation of crop growth for potential and water-limited production situations. As applied to spring wheat.	H.H. van Laar, J. Goudriaan & H. van Keulen (Eds.)
No. 15	Model-based decision support in agriculture	H.F.M. ten Berge & A. Stein (Eds.)
No. 16	Information technology as a tool to assess land use options in space and time	J.J. Stoorvogel, J. Bouma & W.T. Bowen
No. 17	LUCTOR: Land Use Crop Technical Coefficient Generator; version 2.0	H. Hengsdijk, A. Nieuwenhuyse & B.A.M. Bouman
No. 18	PASTOR: A Technical Coefficient Generator for Pasture and Livestock Systems in the Humid Tropics; version 2.0	B.A.M. Bouman, A. Nieuwenhuyse & H. Hengsdijk
No. 19	Active Methodology: Proceedings of a seminar series 1997/1998	A. Stein, M.K. van Ittersum & G.H.J. Koning (Eds.)
No. 20	FUSSIM2: A twodimensional simulation model for water flow, solute transport, and root uptake of water and nutrients in partly unsaturated porous media	M. Heinen & P. de Willigen
No. 21	Food security at different scales: demographic, biophysical and socio-economic considerations	P.S. Bindraban, H. van Keulen, A. Kuyvenhoven, R. Rabbinge & P.W.J. Uithol (Eds)