

Diss. ETH No. 10086

The Role of Aphid Behaviour in the Epidemiology of Potato Virus Y: a Simulation Study

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZÜRICH
for the degree of
Doctor of Technical Sciences

presented by
Thomas Nemecek
Dipl. Ing. Agr. ETH
born 3rd April, 1962
citizen of Obersiggenthal (AG)

accepted on the recommendation of
Prof. Dr. V. Delucchi, examiner
Prof. Dr. W. Nentwig, co-examiner
Dr. A. Fischlin, co-examiner
Dr. J. Derron, co-examiner

1993

7.93 300 V 14199

Table of Contents

ABSTRACT	7
KURZFASSUNG	9
1. INTRODUCTION	11
2. MATERIAL AND METHODS	16
2.1. Modelling and Simulation Tools	16
2.2. The PAV Simulation System	17
2.3. Implementation of the EPOVIR Model	20
2.4. Statistical Analyses of Simulation Results	20
2.4.1. Monitoring Times of Output Variables	20
2.4.2. Spatial Pattern of Diseased Plants	20
2.5. Sensitivity Analysis	25
2.6. Statistical Computations	26
2.7. Field Data Sampling	26
2.7.1. Experimental Fields	26
2.7.2. Epidemiological Experiments	26
2.7.3. Artificial Inoculation Experiments	27
2.7.4. Monitoring of Aphid Abundance	28
2.7.5. Meteorological Data	28
3. THE PVY-PATHOSYSTEM	29
3.1. Real System	29
3.1.1. System Elements	29
3.1.2. System Structure	30
3.1.3. System Behaviour	34
3.2. Model Systems	34
3.2.1. Resolution Levels and System Boundary	34
3.2.2. Epidemic Models	36
4. THE ROLE OF VECTOR DISPERSAL IN SPATIO-TEMPORAL VIRUS SPREAD - A SIMULATION STUDY	40
4.1. Model Description	40
4.2. Vector Dispersal and Virus Epidemics	45
4.3. Discussion	48
5. EXPERIMENTAL AND MARKOV MODEL STUDIES OF WINGED APHID BEHAVIOUR	50
5.1. Material and Methods	51
5.1.1. Observation of Behaviour Sequences	51
5.1.2. Analysis of Markov Chains	54
5.1.3. Observation of Flight Behaviour	56
5.2. Quantification of Behaviour Sequences	59
5.3. Analysis of the Behaviour Model	65

5.4. Flight Distance and Flight Direction in Relation to Weather	69
5.5. Discussion	73
5.5.1. Behaviour Sequences	73
5.5.2. Behaviour Model	74
5.5.3. Flight Behaviour	75
6. EPOVIR: MODEL OF THE EPIDEMIOLOGY OF POTATO VIRUSES	78
6.1. The Concept of Inoculation and Infection Submodels	78
6.2. Model Structure	79
6.3. Model Equations	81
6.3.1. Inoculation Submodel	81
6.3.2. Infection Submodel	85
6.3.3. Plant Growth Submodel	89
6.3.4. Water Balance Submodel	89
6.4. Parameter Estimation	89
6.4.1. Inoculation Submodel	89
6.4.2. Infection Submodel	92
6.4.3. Plant Growth and Water Balance Submodels	94
7. SIMULATION STUDIES OF THE EPOVIR-MODEL	95
7.1. Analysis of Model Behaviour	95
7.1.1. Standard Run	95
7.1.2. Simulation Technical Aspects	96
7.1.2.1. Effect of Torus and Selection of Representative Field Size	96
7.1.2.2. Selection of Representative Size of Stochastic Sample	98
7.1.3. Relationship Between Epidemic Progress and Predictor Variables	100
7.1.4. Structural Analysis	101
7.1.4.1. Nonrandomness of Spatial Pattern	101
7.1.4.2. Vector Abundance	102
7.1.4.3. Behaviour Characteristics	103
7.1.4.4. Vector Propensity	104
7.1.4.5. Plant-Virus Interaction	104
7.2. Validation	105
7.2.1. Validation Experiments	105
7.2.2. Validation of Temporal Behaviour	106
7.2.3. Validation of the Spatial Pattern	110
7.3. Sensitivity Analysis	114
7.3.1. Relative Importance of Vector Behaviour Parameters	114
7.3.2. Effects of Single Parameters	117
7.3.2.1. Behaviour Sequence Parameters	117
7.3.2.2. Dispersal Parameters	122
7.3.2.4. Other Parameters	124
7.4. The Role of Vector Groups and Aphid Morphs	126
7.4.1. Colonizing and Noncolonizing Aphids	126
7.4.2. Winged and Wingless Aphids	128
7.5. Assessment of Control Measures Affecting Winged Aphid Behaviour	129
7.5.1. Measures Reducing Immigration	129
7.5.2. Aphicides	130
7.5.3. Repellents	131
7.5.4. Plant Resistance to Aphids	133
7.5.5. Alarm Pheromones	134

8. DISCUSSION	136
8.1. Model Structure and Behaviour	136
8.2. Validation	137
8.3. Structural and Sensitivity Analysis	139
8.4. The Role of Vector Groups and Aphid Morphs	144
8.5. Assessment of Control Measures Affecting Winged Aphid Behaviour	145
8.6. Modelling and Simulation Tools	147
9. CONCLUSIONS	149
REFERENCES	152
APPENDIX I: GLOSSARY	160
APPENDIX II: TABLE OF SYMBOLS	163
APPENDIX III: LISTING OF THE SIMULATION MODEL 'SPATIAL EPIDEMIC' [0]	169
APPENDIX IV: LISTINGS OF THE PAV SIMULATION SYSTEM	174
PAV Base Modules	174
PAVTypes [1] D *)	174
PAVBase [2] D	175
PAVAux [3] D	177
PAVSensAna [4] D	177
General Input and Validation Modules	179
PAVValid [5] D	179
Weather [6] D	179
PAVSuctTrap [7] D	180
EPOVIR modules	180
SoilWat [8] DI (Soil Water Balance Submodel of EPOVIR)	180
PotatoMod [9] DI (Plant Growth Submodel of EPOVIR)	186
PotModValid [10] D	195
PAVMetHour [11] D	195
PAVSteadyBeh [12] D	195
PAVInocs [13] DI (Inoculation Submodel of EPOVIR)	196
PAVStochInfs [14] DI (Infection Submodel of EPOVIR)	204
PAVSpatDistr [15] D	218
PAVMonit [16] D	218
PAV Management Modules	219
PAVSetGetObj [17] D	219
PAVMoEDMngr [18] D	219
PAVExpmnts [19] D	220
PAVMaster [20] I	220
APPENDIX V: LISTINGS OF AUXILIARY LIBRARY MODULES	221
Experiment Definition Modules	221
DFData D	221

*) D = definition, I = implementation module, the numbers in brackets refer to the module numbers in Tab. 2.1 and Tab. A2.

ReadWriteDF D	222
SetGetDF D	224
Other Auxiliary Modules	225
CAModBase D	225
RandGens D	227
SpatPattern D	228
StochStat D	229
Vectors2D D	231
CURRICULUM VITAE	232
ACKNOWLEDGEMENTS	232

ABSTRACT

THOMAS NEMECEK: *THE ROLE OF APHID BEHAVIOUR IN THE EPIDEMIOLOGY OF POTATO VIRUS Y: A SIMULATION STUDY*, PH. D. THESIS NO. 10086, ETH ZÜRICH, SWITZERLAND, 1993

The behaviour of virus vectors is an element of plant virus pathosystems that is considered to be highly relevant, but its effects on virus epidemics are little known. In this study a combined experimental-simulation approach was applied to quantify the effects of changes in vector behaviour parameters on virus spread and to compare the importance of these effects with those of other epidemiological parameters. The nonpersistently aphid-transmitted potato virus Y (PVY), which causes high economic damage to potatoes and other crops, was chosen for a case study.

The study was part of a project, where interactive modelling and simulation tools for ecological research were developed and applied. Interactive simulation, full access to the programming language Modula-2 and modular organization of models proved to be very useful features of these tools.

As a first step, the effects of the dispersal distances of the virus vectors on spatio-temporal virus spread was studied with a cellular automaton model called 'Spatial Epidemic'. It simulates each plant as a cell, which can have the state healthy, latently infected, primary infection source, secondary infection source and removed. The temporal rate of disease spread was maximal for large dispersal distances in a torus and was markedly reduced for distances below 5 to 10 meters.

To estimate parameters for a Markov chain model, the behaviour sequences of winged *Myzus persicae* SULZER (a potato colonizing species) and *Aphis fabae* KALTENBACH (a noncolonizing species) were observed on potato plants in a climatic chamber and in the field. Five behaviour states were distinguished: flight, walking, resting, probing and 'long penetration' (stylet penetration activities longer than 3 min). *A. fabae* had higher flight, walking and probing frequencies and shorter penetration times than *M. persicae*. It changed the behaviour state twice as often as *M. persicae*. A comparison with behaviour sequences of *Aphis nasturtii* KALTENBACH (colonizing) and *Brachycaudus helichrysi* KALTENBACH (noncolonizing) showed that differences in behaviour between the groups of the colonizing and noncolonizing species were much larger than those within the groups. The parameters estimated for *A. fabae* and *M. persicae* were considered as representative for the respective groups of aphid species.

Flight behaviour parameters were estimated by field observation of alatae with the aid of simulation models. The flight direction was strongly correlated with the wind direction. Short flights were prevailing and the mean flight distance was 13 m. An analysis of the wind directions showed that the main wind direction during periods suitable for aphid flight substantially differed from the overall main wind direction.

Finally the model EPOVIR (= epidemiology of potato viruses) was developed to study the effects of changes in the estimated behaviour parameters on PVY epidemics. It consists of four submodels:

- *Inoculation submodel*: calculates vector intensity (defined as number of inoculations per source plant per time unit) as the product of (1) unsettled winged vector abundance in the field, (2) the rate of interplant vector movement preceded

and followed by probing and (3) vector propensity (species specific transmission probability).

- *Infection submodel*: a modified version of the model 'Spatial Epidemic', simulates virus transmissions from infection sources to healthy plants, taking the nonrandom distribution of the inoculations into account. The disease state of individual plants and the proportion of infected tubers is calculated.
- *Plant growth submodel*: simulates the dry matter of leaves, stems, tubers, roots and assimilates and the physiological age structure of the potato canopy.
- *Soil water balance submodel*: simulates the water stress for the plants.

The model was validated against eleven data sets collected during eight years in Western Switzerland. The model behaviour was judged satisfactory in seven data sets.

The sensitivity analysis showed that behaviour sequence parameters were little sensitive. The most important among them were the scaling factors of walking frequency and 'long penetration' duration, the former was positively, the latter negatively correlated to PVY spread. Flight and probing frequencies showed optima for PVY spread, which were near the values found in the behaviour observations: lower flight or probing frequencies reduced virus incidence, whereas higher ones hardly affected it. The most sensitive parameters were: take-off threshold for temperature, the age resistance parameter, the vector efficiency, the relative immigration rate and the emigration probability, the latent period for PVY in the plants and the initial disease incidence. The nonrandomness of the spatial pattern of infected plants was more determined by the ratio of infections contributed by flying and walking aphids than by the mean flight and walking distances or the distribution of flight directions. This nonrandomness should be considered in epidemic models, since models assuming inoculations to be randomly distributed in space would overestimate disease severity by up to 25 %.

Although behaviour sequence parameters were little sensitive, the much more frequent dispersal allowed the noncolonizing species to transmit PVY twice as frequently than colonizing ones. This difference should be considered in virus epidemic models. Due to lower dispersal frequency and abundance, colonizing aphids were responsible for only 20 % of the infections on the average of the eight years in the simulations. The potential effect of wingless aphids, which were not included in the model, was estimated in a hypothetical simulation experiment as 2-3 % only.

The potential of influencing vector behaviour by measures reducing immigration, aphicides, repellents, plant resistance and alarm pheromones to control PVY was studied in simulation experiments. Reduced immigration was found to be the most promising method. The other measures can be effective, if the vector abundance is reduced by increasing the emigration or mortality rates. Manipulation of vector behaviour sequences looks less promising. PVY spread can be reduced by lower dispersal frequency due to increased 'long penetration' duration on potato plants, but can also be favoured by increasing the rate of interplant movements. All methods except reduced immigration bear the risk to increase virus spread and can thus not be recommended as a control measure.

To summarize, the behaviour sequence parameters were found less sensitive than formerly believed. Nevertheless, naturally occurring differences in behaviour between colonizing and noncolonizing species and influences of weather on aphid flight are so large that they are relevant for PVY epidemics.

KURZFASSUNG

THOMAS NEMECEK: *DIE ROLLE DES BLATTLAUSVERHALTENS IN DER EPIDEMIOLOGIE DES KARTOFFELVIRUS Y: EINE SIMULATIONSSTUDIE.* DISS NR. 10086, ETH ZÜRICH,
SCHWEIZ, 1993

Das Verhalten der Vektoren von Viruskrankheiten ist ein Element der Pflanzen-Virus-Pathosysteme, welches als wesentlich erachtet wird, dessen Einflüsse auf die Virus-epidemiologie jedoch kaum untersucht sind. In dieser Arbeit wurden Auswirkungen der Änderungen von experimentell ermittelten Verhaltensparametern der Vektoren auf die Virus-epidemiologie mittels Simulationen untersucht und mit Auswirkungen anderer Parameter verglichen. Der nichtpersistente durch Blattläuse übertragene Kartoffelvirus Y (PVY), welcher grosse wirtschaftliche Verluste in Kartoffeln und anderen Nutzpflanzen verursacht, wurde für eine Fallstudie ausgewählt.

Die Fallstudie war Teil eines Projektes, in welchem interaktive Modellierungs- und Simulationswerkzeuge entwickelt und angewandt wurden. Interaktive Simulation, uneingeschränkter Zugriff auf die Programmiersprache Modula-2 und modulare Organisation der Modelle waren jene Eigenschaften der Werkzeuge, die sich als sehr nützlich für die Fallstudie erwiesen.

Zuerst wurde der Einfluss der Dispersionsdistanzen auf die räumlich-zeitliche Ausbreitung des PVY mittels des Simulationsmodells "Spatial Epidemic" untersucht, welches der Klasse der zellulären Automaten angehört. Jede Pflanze wird als eine Zelle dargestellt, welche die Zustände gesund, latent infiziert, primäre Infektionsquelle, sekundäre Infektionsquelle und eliminiert annehmen kann. Die zeitliche Ausbreitungsgeschwindigkeit war maximal für grosse durchschnittliche Dispersionsdistanzen in einem Torus und war wesentlich reduziert für durchschnittliche Dispersionsdistanzen unter 5 bis 10 m.

Die Verhaltenssequenzen geflügelter *Myzus persicae* SULZER (eine Kartoffeln besiedelnde Art) und *Aphis fabae* KALTENBACH (nicht besiedelnd) wurden in einer Klimakammer und im Feld beobachtet, um Parameter für ein Markov-Modell zu schätzen. Hierbei wurden fünf Verhaltenszustände unterschieden: Flug, Laufen, Ruhe, Probesaugsticke und "lange Saugsticke" (Saugsticke länger als 3 min). *A. fabae* hatte eine grössere Flug-, Lauf- und Probesaugstichhäufigkeit als *M. persicae*. *A. fabae* wechselte den Verhaltenszustand zweimal häufiger als *M. persicae*. Ein Vergleich mit Verhaltenssequenzen von *Aphis nasturtii* KALTENBACH (besiedelnd) und *Brachycaudus helichrysi* KALTENBACH (nicht besiedelnd) zeigte, dass die Unterschiede im Verhalten zwischen besiedelnden und nicht besiedelnden Arten wesentlich grösser als innerhalb dieser Gruppen waren. Die für *A. fabae* und *M. persicae* geschätzten Verhaltensparameter wurden als repräsentativ für die jeweiligen Artengruppen erachtet.

Parameter des Blattlausfluges wurden durch Feldbeobachtungen mit Hilfe von Simulationsmodellen geschätzt. Die Flugrichtung war stark mit der Windrichtung korreliert. Kurze Flüge überwogen; die mittlere Flugdistanz betrug 13 m. Eine Analyse zeigte, dass die Windrichtung während Perioden, in denen Blattlausflug stattfinden kann, wesentlich von der allgemeinen Hauptwindrichtung abweicht.

Schliesslich wurde das Modell EPOVIR (= epidemiology of potato viruses) für die Untersuchung der Auswirkungen von Änderungen in den ermittelten Verhaltensparametern auf die PVY Epidemiologie entwickelt. Es besteht aus vier Untermodellen:

- *Virusübertragungsmodell:* berechnet die Vektorintensität (definiert als Anzahl Inokulationen pro Infektionsquelle und Zeiteinheit) als Produkt der (1) Abundanz geflügelter Vektoren im Feld, (2) Häufigkeit der Pflanzenwechsel, denen Probesaugsticke vorausgehen und nachfolgen und (3) der Übertragungsfähigkeit einzelner Vektorarten.
- *Virusinfektionsmodell:* eine erweiterte Version des Modells "Spatial Epidemic", simuliert Virusübertragungen von Infektionsquellen zu gesunden Pflanzen unter Be-

achtung der räumlich nichtzufälligen Verteilung der Inokulationen. Die Zustände einzelner Pflanzen in Bezug auf die Krankheit, sowie der Anteil infizierter Knollen werden berechnet.

- *Pflanzenwachstumsmodell*: simuliert die Trockenmasse der Blätter, Stengel, Wurzeln und Knollen, sowie die physiologische Altersstruktur der Blätter.
- *Bodenwasserhaushaltsmodell*: simuliert den Wasserstress für die Pflanzen.

Das Modell wurde anhand von elf Datensätzen validiert, die während acht Jahren an der Eidgenössischen landwirtschaftlichen Forschungsanstalt in Changins (Schweiz) erhoben wurden. Die Übereinstimmung der Simulationen mit den Versuchsresultaten war in sieben Datensätzen zufriedenstellend.

Die Verhaltenssequenzparameter erwiesen sich als wenig sensitiv. Die wichtigsten unter diesen Parametern waren Skalierungsfaktoren der Laufhäufigkeit und der Dauer "langer Saugstiche", wobei der erstere positiv, der letztere negativ mit der PVY-Ausbreitung korreliert war. Die Flug- und Probesaugstichhäufigkeit zeigten Optima für die PVY-Übertragung, die nahe den experimentell ermittelten Werten lagen. Weniger häufige Flüge und Probesaugstiche konnten die Virusausbreitung vermindern, höhere Häufigkeiten brachte jedoch kaum eine Änderung. Die sensitivsten Parameter waren: Temperaturschwellenwert für Abflug, der Altersresistenzparameter, die Übertragungsfähigkeit für PVY, die relative Immigrationsrate und die Emigrationswahrscheinlichkeit, die Latenzperiode für PVY in den Pflanzen und der Befall des Ausgangsaatguts. Das räumliche Verteilungsmuster der Infektionen war hauptsächlich durch das Verhältnis der Virustragungen durch laufende und fliegende Alaten beeinflusst und nur wenig durch die mittleren Lauf- und Flugdistanzen und die Flugrichtungsverteilung. Die nichtzufällige räumliche Verteilung der Inokulationen sollte in Virusepidemiomodellen berücksichtigt werden, da Modelle, die eine zufällige räumliche Verteilung der Inokulationen annehmen, den Befallsgrad um bis zu 25 % überschätzen können.

Obwohl Verhaltenssequenzparameter wenig sensitiv waren, können nichtbesiedelnde Arten dank viel häufigerem Pflanzenwechsel PVY zweimal häufiger übertragen als besiedelnde Arten. Diese Unterschiede sollten in Epidemiomodellen berücksichtigt werden. Da die besiedelnden Arten zusätzlich noch eine kleinere Abundanz aufwiesen, waren sie im Mittel von acht Jahren nur für etwa 20 % der Virusübertragungen verantwortlich. Der potentielle Beitrag ungeflügelter Blattläuse, welche nicht im Modell berücksichtigt wurden, wurde in einer hypothetischen Simulation auf lediglich 2-3 % geschätzt.

Durch Simulationen wurde ferner untersucht, inwieweit sich Veränderungen im Vektorverhalten durch Aphizide, Repellentien, Pflanzenresistenz, Alarmpheromone und Massnahmen, welche die Immigration vermindern, zur Bekämpfung des PVY einsetzen lassen. Verminderte Immigration erwies sich als die aussichtsreichste Methode. Die anderen Massnahmen können effektiv sein, falls die Vektorabundanz durch erhöhte Emigration oder Mortalität vermindert wird. Induzierte Änderungen der Verhaltenssequenzen der Vektoren scheinen jedoch weniger geeignet zur Virusbekämpfung. Wohl kann die PVY-Übertragung durch geringere Dispersionshäufigkeit dank längeren Saugstichen auf Kartoffelpflanzen reduziert werden. Sie kann jedoch auch durch häufigeren Pflanzenwechsel gefördert werden. Alle Massnahmen außer verminderter Immigration bergen die Gefahr einer erhöhten Virusausbreitung in sich und sind deshalb nicht zu empfehlen.

Zusammenfassend erwiesen sich die Verhaltenssequenzparameter als weniger sensitiv als bisher angenommen. Anderseits sind die natürlichen Unterschiede im Verhalten der besiedelnden und nicht besiedelnden Arten und die Einflüsse der Wetters auf den Blattlausflug so gross, dass sie für die Virusepidiologie bedeutend sind.

1. INTRODUCTION

Virus diseases can cause important crop losses (MATTHEWS, 1991). Since most viruses infect their hosts systemically, they are especially widespread in vegetatively propagated crops, such as potatoes (KLINKOWSKI, 1980).

The earliest description of virus diseases in potatoes dates from the 18th century. The disease called 'curl' or 'Krauselkrankheit' was believed to be a degeneration due to continued vegetative reproduction or due to 'being too long planted in a country, widely different from its native soil and climate' (NAISMITH in 1792, cited by DAVIDSON, 1928). It was noted that 'curl' occurred less at higher altitudes and in northern areas. Around 1920 it was shown that 'curl' was caused mainly by two aphid-transmitted virus diseases (BAGNALL, 1991): potato virus Y (PVY) and potato leaf roll virus (PLRV). *Myzus persicae* SULZER was found to be a major vector of these diseases. *M. persicae* is a potato colonizing or resident species¹, i.e. it can continuously reproduce on potato.

Aphids play a key role in the epidemics of PVY and PLRV, because aphid transmission is the only relevant way of virus dissemination from plant to plant in the field (BEEMSTER & DE BOKX, 1987). It was found that the low virus disease incidence at higher altitudes and in northern areas coincided with a low abundance of aphids, especially of *M. persicae*. BROADBENT (1950) found good correlations between the abundance of *M. persicae* and virus spread in potatoes. Therefore the abundance of this vector was believed to be a major factor in the epidemiology of potato virus diseases (GABRIEL, 1987). Despite the results of earlier experiments indicating that species other than *M. persicae* can transmit PVY (BRADLEY & RIDEOUT, 1953; EDWARDS, 1963), the role of these other species was disregarded, until VAN HOOF (1977) observed that PVY spread occurred before the spring flight of *M. persicae* and that other, mainly non-colonizing or non-resident species accounted for this spread. The probability of PVY transmission, which differs among aphid species, has been estimated in a series of experiments (reviewed by DE BOKX & PIRON, 1990). IRWIN & RUESINK (1986) defined the species specific ability to transmit a virus disease as vector propensity.

In addition to the presence of vectors (abundance) and the ability to transmit a virus disease (vector propensity), a successful transmission requires that an aphid probes or feeds on an infection source plant, moves to a healthy plant and again probes or feeds. Although dispersal and probing/feeding behaviour of the vectors is indispensable for virus dissemination and is believed to be important for the epidemiology of aphid-borne virus diseases, its role has been investigated little and is poorly understood (DE BOKX, 1989; IRWIN & KAMPMEIJER, 1989).

Moreover, different authors assess the relative importance of vector behaviour versus vector abundance in the epidemic process in a controversial manner. In a review of the role of aphids in virus epidemiology, ZITTER (1977) found good correlations between disease incidence and vector abundance. He concluded that vector abundance is probably the most important factor in determining virus spread. Although ROCHOW (1974) and HARREWIJN (1989) agree that vector abundance is important, they put forward the hy-

¹For the reader's convenience, definitions of terms are also listed in the Glossary (Appendix I).

pothesis that vector behaviour determines virus spread even more than vector abundance does. If this hypothesis is true, vector behaviour parameters¹ should be among the most sensitive ones in determining the rate of virus spread, if included in a virus epidemic model. Therefore arises the need to compare the sensitivity of virus spread to changes in vector behaviour versus changes in vector abundance.

Insect behaviour is determined mainly genetically, but influenced also by the environment (MATTHEWS & MATTHEWS, 1978). Aphid species show genetically determined differences in behaviour (WOODFORD, 1976; WEIDEMANN, 1981; DIXON, 1985) and in host ranges (BLACKMAN, 1990). Aphid behaviour can be decomposed into behaviour units (HUNTINGFORD, 1984), out of which only dispersal and probing/feeding activities are directly involved in virus transmission (IRWIN & RUESINK, 1986). In the following the term 'behaviour' will be restricted to these epidemiologically relevant activities; other activities, such as reproductive behaviour, will not be considered.

Virus diseases are transmitted mainly during host selection (KENNEDY *et al.*, 1959; KRING, 1972). In this phase of aphid life, individuals show a typical behaviour (JOHNSON, 1969; KRING, 1972; KLINGAUF, 1987; ROBERT, 1987). This behaviour, which is time continuous, can be divided into a finite number of distinct states (SLATER, 1978; HUNTINGFORD, 1984). The temporal sequences of these states and the spatial movement can be quantified (FAGEN & YOUNG, 1978; HUNTINGFORD, 1984). During host selection aphids react to stimuli of their natural environment (plants, weather, etc., KENNEDY *et al.*, 1959) and stimuli produced by man (GIBSON & RICE, 1989), which can influence their behaviour quantitatively.

Such differences in behaviour, due to genetic differences or to different environmental stimuli, are not taken into account in most epidemic models and forecasting systems of virus diseases. E.g. the haulm-killing dates for seed potatoes in the Netherlands are today determined by vector abundance and propensity (VAN HARTEN, 1983); vector behaviour is ignored, due to the difficulty in quantifying this complex process (PETERS *et al.*, 1990). It is unknown, how the rate of virus spread can be affected by different behaviour on host and nonhost plants and by environmentally modifiable behaviour parameters. Better understanding of the relationship between vector behaviour and epidemiology could help to improve virus-forecasting systems, which are needed in seed potato production (HARRINGTON *et al.*, 1986; DE BOKX, 1989).

According to the aphid-host plant relationship, aphid species can be grouped into potato colonizing and noncolonizing species. Individuals of both groups can be found on potato plants, since host recognition is possible only after landing and probing on the plant (KENNEDY *et al.*, 1959; KENNEDY & BOOTH, 1963). The different behaviour of the two species groups on potato plants leads to differences in the transmission of the two main potato viruses (PETERS, 1987). PLRV is transmitted by colonizing species only, since its persistent transmission mode requires feeding times of several hours to acquire and to inoculate the virus and a latent period of about one day in the vector (BOS, 1983; PETERS, 1987). PVY on the other hand can be transmitted also by noncolonizing vectors, because its nonpersistent transmission mode requires only probes of some seconds to acquire and inoculate the virus, but no latent period. Hence, the effect of differences in be-

¹The term 'parameter' will be used in the sense of 'parameter of a mathematical, simulation or statistical model'.

haviour between colonizing and noncolonizing species on virus epidemiology can be studied only for PVY.

Due to their different reactions to potato plants, colonizing and noncolonizing species are also affected differently by changes in plant state (e.g. induced by fertilization) and by control measures, such as planting of aphid-resistant varieties (HARREWIJN, 1989). Therefore it is important to quantify the contribution of both species groups to PVY epidemics. Some authors consider colonizing species to be more important than noncolonizing ones (GABRIEL *et al.*, 1975; TURL & MACDONALD, 1987; CUPERUS *et al.*, 1988), others found that colonizing species are responsible for only a small fraction of the virus infections (SIGVALD, 1986; HARRINGTON *et al.*, 1986). A quantification of the contribution of colonizing and noncolonizing species to PVY spread with a simulation model could help to improve our understanding of epidemic mechanisms and to optimize virus control.

Noncolonizing species seem to probe and move more frequently than colonizing ones (KENNEDY *et al.*, 1959; MCLEAN & KINSEY, 1968; PETERS *et al.*, 1990), a behaviour called 'restlessness'. A higher dispersal and probing frequency is believed to result in a higher rate of virus spread (KENNEDY, 1976; ROMANOW, 1985; CARTER & HARRINGTON, 1991). According to this hypothesis, the behaviour of noncolonizing species should enable them to transmit PVY more frequently than colonizing species, provided all other factors be kept equal. Such restlessness can apparently also be induced by man. Accelerated dissemination of aphid-borne viruses, e.g. after insecticide treatment (GABRIEL *et al.*, 1981) or on plants resistant to aphids (KENNEDY, 1976) was attributed to increased 'restlessness' of the vectors. However, no experimental evidence of a positive correlation between probing frequency and virus transmission rates have been detected under laboratory conditions (WEIDEMANN, 1981; BOITEAU & SINGH, 1991). The hypothesis that virus transmission rates are positively correlated to the dispersal and probing frequencies in the field should be tested, since this would allow to improve virus control.

Insecticides and antagonists, which can reduce the incidence of the persistently transmitted PLRV, are hardly effective against the nonpersistently transmitted PVY, because they do not kill the vectors fast enough to prevent transmission (DE BOKX, 1989). GIBSON & RICE (1989) suggested that influencing vector behaviour could be a promising alternative for nonpersistent virus control, e.g. for PVY, which is the major virus problem in most potato production regions (BEMSTER & DE BOKX, 1987). Yet, the use of such methods requires knowledge of their effects on vector behaviour and of the relationship between vector behaviour and virus spread. At present, both relationships are poorly understood (GIBSON & RICE, 1989; IRWIN & KAMPMEIJER, 1989), which limits the application of behaviour influencing methods. This study shall contribute to the understanding of the second step, the relationship between vector behaviour and virus spread.

For this purpose the system could be manipulated experimentally (GIBSON & RICE, 1989). The disadvantage of this approach is that other parts of the system are also affected. E.g. alarm pheromones affect aphid behaviour (PHELAN & MILLER, 1982), but interfere also with the virus acquisition process (GIBSON *et al.*, 1984), so that these two phenomena can hardly be distinguished, if alarm pheromones are applied in virus epidemic studies. Moreover, the ineffectiveness of alarm pheromones against PVY in the

field (HILLE RIS LAMBERS & SCHEPERS, 1978) could also be caused by inadequate application techniques (GIBSON & RICE, 1989). Due to such difficulties, simulation models calibrated and validated on field experiments, are preferable to a purely experimental approach, because only simulation models allow to control all these conditions easily. Epidemic models could further be used to assess the effects of ignoring behaviour parameters in these models, by comparing models including behaviour parameters and processes with models ignoring them.

ZADOKS & SCHEIN (1979) pointed out that the choice of an adequate level of resolution in time, space and biological organization (individual, population, community) might be crucial for modelling studies of plant-pathogen interactions. However, most of the currently available models of viral and fungal plant diseases describe epidemic processes during one season and for a single field (e.g. VANDERPLANK, 1963; MARCUS & RACCAH, 1986; RUESINK & IRWIN, 1986; SIGVALD, 1986; MADDEN *et al.*, 1990; MARCUS, 1990). The adequate level for the present study had yet to be defined.

To study the role of vector behaviour in an epidemic model, virus spread should not only be modelled in time but also in space, since vector behaviour can influence both (GRAY *et al.*, 1986a; IRWIN & KAMPMEIJER, 1989). Analytical models can describe epidemics either in time or in space, but only simulation models can appropriately include both (ZADOKS & SCHEIN, 1979). A few such models have been used in plant disease epidemiology (e.g. KAMPMEIJER & ZADOKS, 1977; BERGER & FERRISS, 1989; VAN DER WERF *et al.*, 1989), partly using nonstandard cellular automaton models (WOLFRAM, 1986). Cellular automata are a class of time discrete models, representing a one-, two- or three-dimensional space by discrete cells with discrete states (e.g. healthy or infected). To quantify temporal epidemics, disease severity, i.e. the fraction of plant tissue that is infected, was used in the model of KAMPMEIJER & ZADOKS (1977), whereas the models of BERGER & FERRISS (1989) and VAN DER WERF *et al.* (1989) used disease incidence, i.e. the fraction of infected plants in the plant population. In a PVY epidemic management model, the fraction of tubers infected (disease severity) is preferable to disease incidence as a measure of temporal epidemics, since tuber infection is measured in all seed potato fields resp. epidemiological experiments and is further of high economic interest. The spatial pattern, i.e. the spatial arrangement of infected plants (after CAMPBELL & MADDEN, 1990), was not quantified but represented graphically in these models. Quantitative analysis methods for spatial patterns of plant diseases were divided into three classes by CAMPBELL & MADDEN (1990): doublets and runs analysis, quadrat-based analysis and distance based analysis. The first class of methods compares the occurrence of sequences of healthy resp. diseased plants with a random model. Quadrat-based analyses include various indices of dispersion and spatial autocorrelation analysis. Distance based analysis methods compare the frequency distribution of the distances between diseased plants with a random model. The two-dimensional distance class analysis of GRAY *et al.* (1986b), which allows to analyze disease incidence data in a lattice structure, was designed for and successfully applied to the analysis of spatial patterns of plant virus diseases.

In the epidemic models mentioned above, vector behaviour is included implicitly in the infection rate (e.g. Vanderplank's model) or similar parameters. Since the rate of disease spread is sensitive to the infection rate (ZADOKS & SCHEIN, 1979), the former could

also potentially be sensitive to vector behaviour parameters. To allow a study of the role of vector behaviour in the epidemic process, the rate of PVY spread should be formulated as a function of vector behaviour parameters. Only the model of beet yellowing viruses of VAN DER WERF *et al.* (1989) includes some of these processes, but is not applicable to PVY, because the semi-persistently transmitted beet yellowing viruses involve behaviour characteristics different from those likely to be relevant for nonpersistent virus transmission. Moreover, the model takes into account only colonizing species and wingless aphids (apterae), whereas PVY is apparently transmitted mainly by winged aphids (alatae, BROADBENT & TINSLEY, 1951) of colonizing and noncolonizing species. Thus, in order to account also for vector behaviour, a new epidemic model ought to be developed. Hereby Markov chains could be used, since they have in many cases served as simple descriptive models of animal behaviour sequences (CANE, 1978; METZ *et al.*, 1983; HACCOU, 1986).

In short the following basic question shall be studied in this thesis by means of a PVY epidemic model: how does the fraction of tubers infected by PVY and the spatial pattern of infected potato plants respond to changes in dispersal and probing parameters of the aphid vectors? The results should allow an assessment of the relative importance of vector behaviour by comparing the sensitivity of behaviour parameters to that of other parameters, the investigation of possible consequences of ignoring vector behaviour in virus epidemic models as well as the comparative study of the effects of different behaviour of colonizing and noncolonizing aphids on PVY epidemics. The model should further allow to quantify the contribution of colonizing and noncolonizing species to PVY transmission and to evaluate the potential usefulness of measures influencing vector behaviour for PVY control.

The study served also as a case study for a project ('Computer Aided Modelling and Simulation in Quantitative Ecology', CAMS; FISCHLIN, 1991), where interactive modelling and simulation software based on modelling theory (ZEIGLER, 1976; WYMORE, 1984) were developed. These tools were applied in the case study.

The analysis begins with a system description and a review of epidemic models applied to similar problems (Chapter 3). The spatio-temporal epidemic models, required for the study, tend to be more complex than models of temporal epidemics only (ZADOKS & SCHEIN, 1979). Therefore a simple simulation model was developed first to study the relationship between the dispersal distance and spatio-temporal virus epidemics (Chapter 4). Epidemiologically relevant parameters of winged aphid behaviour, which could not be derived from the literature, were estimated in field and laboratory experiments (Chapter 5). The behaviour parameters derived were subsequently used to construct a model of PVY epidemics (EPOVIR, Chapter 6 & 7) and the model was validated using experimental field data sampled during eight years. Thereafter, the response of the model outputs to changes in behaviour parameters was studied by varying single parameters in a first step. In a second step, several behaviour parameters were varied simultaneously to compare the different behaviour of colonizing and noncolonizing aphids and to assess the effects of different measures influencing aphid behaviour on PVY epidemiology.

2. MATERIAL AND METHODS

2.1. Modelling and Simulation Tools

Modelling and analysis of the complex interactions in ecological and agricultural systems, such as the potato-aphid-virus system in this case study, seems difficult with the currently commercially available modelling and simulation software (FISCHLIN, 1991). Ecological and agricultural systems are often ill-defined, i.e. neither the model equations nor the parameter values are well known or fully understood at the beginning of the study (FISCHLIN, 1982 & 1991). The analysis of such a system can be supported by interactive modelling and simulation software. Moreover, models for subsystems of the potato-aphid-virus system are available, which belong to different classes of model formalisms, e.g. differential equation models, difference equation models, discrete-event models, cellular automata, etc.. Most of the currently available software does not well support the coupling of models belonging to different model classes. Hence, to allow the development and study of epidemic models and other ecosystem models, software tools were developed during the course of the CAMS-project in the Systems Ecology Group of the Swiss Federal Institute of Technology and subsequently also applied in the case study. Based on concepts of the modelling theory (ZEIGLER, 1976; WYMORE, 1984), the software was designed to support interactive modelling and simulation, the coupling of models with different model formalisms and modular modelling, allowing to split complex models into a number of smaller submodels.

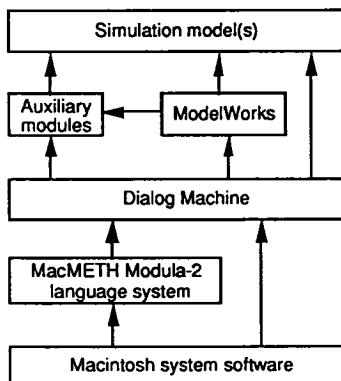


Fig. 2.1: Software layers involved in the simulation studies. The arrows represent imports.

All software used in the simulation studies (Fig. 2.1), except the system software, was implemented in Modula-2 (WIRTH, 1985), using the *MacMETH Modula-2 language*

system (WIRTH *et al.*, 1992). The programming language Modula-2 emphasizes the module concept, which proved particularly useful in supporting modular modelling.

The modelling and simulation environment is based on the *Dialog Machine* (DM) (FISCHLIN, 1986; FISCHLIN & SCHAUFELBERGER, 1987; KELLER, 1989). The Dialog Machine is a software layer between the computer's system software and application programs, supporting and simplifying implementation of interactive programs. DM programs contain no hardware dependencies and are therefore source code compatible with several personal computers.

The interactive simulation environment *ModelWorks* (FISCHLIN & ULRICH, 1987; FISCHLIN *et al.*, 1990) is built on top of the DM and supports modular modelling, the coupling of discrete and continuous time models and interactive simulation as well as batch oriented simulation. It allows full access to Modula-2 and the DM.

The simulation models were run from the MacMETH-shell (WIRTH *et al.*, 1992) or from the programming session of the RAMSES-Shell (FISCHLIN, 1991). Like the former, the programming session of the RAMSES-Shell supports editing, compiling and executing Modula-2 programs. It offers an enhanced functionality of the MacMETH-shell (WIRTH *et al.*, 1992).

The simulation studies were carried out on AppleTM MacintoshTM computers model II (Motorola 68020 CPU) or model IIfx (Motorola 68030 CPU), partly used as simulation servers (FISCHLIN, 1991). For efficiency reasons, most computations were made by code, which uses the mathematical coprocessor directly (Motorola 68881 [model II] or 68882 [model IIfx]), hereby bypassing SANE (APPLE COMPUTER, 1988). The latter code was produced by a special version of the MacMETH Modula-2 compiler ('Compile20', WIRTH *et al.*, 1992).

2.2. The PAV¹ Simulation System

The PAV simulation system (PAVSS, Fig. 2.2), which was used to simulate the EPOVIR-model and five other models, is composed of 25 modules. The module groups (Fig. 2.2) have the following functions:

- PAV base modules: declaration resp. implementation of globally used types, variables and procedures.
- General input and validation modules: management of input and validation data.
- Model1 to Model6: models installed in the PAVSS. The models of VANDERPLANK (1963), SIGVALD (1986), RUESINK & IRWIN (1986), the stochastic spatial version of EPOVIR (Chapter 6), the deterministic version of EPOVIR and the coupled plant growth-soil water balance model (ROTH *et al.*, in press) may be installed according to needs. The system allows to run any subset of the models simultaneously.
- PAV management modules: experiment definition (installation and deinstallation of models, assignment of a data frame [see below] to the models), management of simulation experiments (structured runs for sensitivity analysis, etc.).

¹PAV is an acronym for Potato-Aphid-Virus.

Tab. 2.1: Imported modules (at the left) and importing modules (upper row) in the PAV simulation system. • = import in definition and implementation module, o = import in implementation module only. Module numbers (#): 1 = PAVTypes, 2 = PAVBase, 3 = PAVAux, 4 = PAVSensAna, 5 = PAVValid, 6 = Weather, 7 = PAVSuctTrap, 8 = SoilWat, 9 = PotatoMod, 10 = PotModValid, 11 = PAVMetHour, 12 = PAVSteadyBeh, 13 = PAVInocs, 14 = PAVStochInfs, 15 = PAVSpatDistr, 16 = PAVMonit, 17 = PAVSetGetObj, 18 = PAVMoEDMngr, 19 = PAVExpmnts, 20 = PAVMaster. For simplicity's sake, modules used in the Vanderplank, Sigvald and Ruesink model and in the deterministic version of EPOVIR only are not listed. The circular imports between the modules 14 and 15 were imposed by the close interrelationship of the two modules and have been chosen by purpose.

	PAV base modules	General input and validation modules	EPOVIR modules								PAV management modules				
#	1 2 3 4	5 6 7	8	9	10	11	12	13	14	15	16	17	18	19	20
PAV management modules	19												•	o	
	18												•	o	o
	17												•	o	
EPOVIR modules	16											•	o	o	
	15											o	•	o	o
	14											•	o	o	o
	13											o	o	o	o
	12								•	o			o	o	
	11								o	o			o	o	
	10								•	o			o	o	
	9								o	o			o	o	
	8								o	o			o	o	
General input and validation modules	7			•							o		o	o	
	6			•							o		o	o	
	5		•								o		o	o	
PAV base modules	4		•								o	o		o	
	3	•	o								o	o		o	
	2	•	o		o	o	o				o	o	o	o	o
	1	•	o	o		o					•	•	o	o	o
experiment definition modules			•	•	•			•			•		•	o	
auxiliary modules		o	o	o	o	o	o	o	o	o	o	o	o	o	o
ModelWorks modules		•	•	o	•	o	•	o	•	•	•	•	•	o	o
DialogMachine modules		•	o	o	o	o	o	o	o	o	o	o	o	o	o

The import structure of the PAV simulation system is shown in Tab. 2.1.

The concept of *data frames* was introduced in order to organize model parameters, input data and validation data sets. A data frame is a set of initial values, parameters, input and validation data that differ in their values from one situation (i.e. one experiment) to the other. The experiment definition modules read the data frames from files, store the data in the memory and supply them to the models, whenever necessary. This procedure allowed easy switching from one situation to another. Twelve data frames were used, corresponding to eleven experimental situations (Tab. 2.3) and one data frame for the standard run. The data frame currently in use is called 'active'. During data frame switches (Fig. 2.3) the objects of the active data frame are assigned to the models. If a new model is activated, the active data frame is assigned to it. The switch of data frames as well as the activation and deactivation of models can be performed interactively by

pull-down menu commands or under program control. It is possible to inactivate all models (e.g. if one wishes to observe input data only), but it is not allowed to be in a state without an active data frame. Fig. 2.3 shows an example of possible states and state transitions.

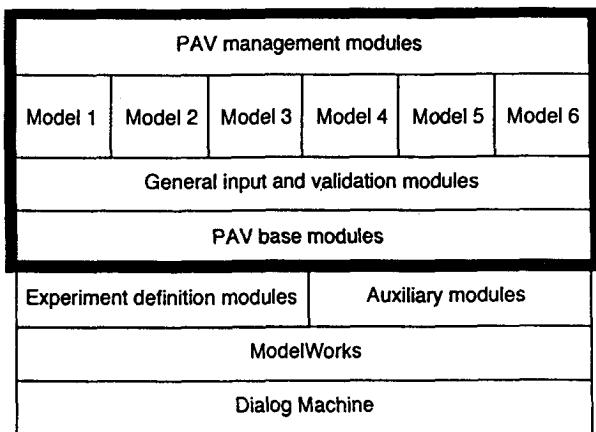


Fig. 2.2: Organization of the PAV simulation system (shadowed box). Objects are imported only in upward or horizontal direction (see Tab. 2.1).

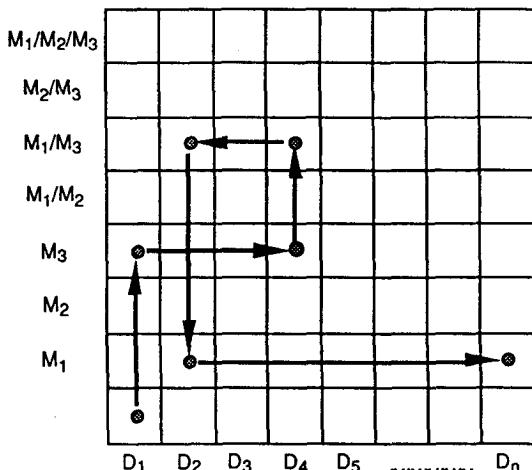


Fig. 2.3: States and one possible path of state transitions of the PAV simulation system (simplified representation of an exemplary case with three models: M_1 = Vanderplank, M_2 = Sigvald and M_3 = stochastic spatial EPOVIR model). D_1 to D_n are n data frames. Each square represents a possible state, those containing circles show states that are taken on during the represented simulation session.

The submodels were coupled according to the model coupling mechanism provided by ModelWorks (FISCHLIN *et al.*, 1990).

2.3. Implementation of the EPOVIR Model

The EPOVIR simulation model is a component of the PAV simulation system (Fig. 2.2). The inoculation submodel was integrated with the forth order Runge-Kutta method, using an integration step of 0.1 day. The other three submodels are discrete time with a time step of one day.

The pseudo-random number generator of WICHMANN & HILL (1982 & 1987) was used to generate uniformly distributed variates in the interval $[0, 1]$ ¹. Random variates of the von Mises distribution (VON MISES, 1918) were generated with the algorithm of BEST & FISHER (1979) using the acceptance-rejection method.

2.4. Statistical Analyses of Simulation Results

2.4.1. MONITORING TIMES OF OUTPUT VARIABLES

The output variables of the EPOVIR-model (p_t = proportion of infected tubers and E = evenness of the distance class distribution, see Chapter 2.4.2) were monitored at each time step (i.e. each day) or only at three times (t_1 to t_3) during a simulation run starting at t_0 and ending at t_{end} :

$$t_i = t_0 + \frac{i}{3} (t_{end} - t_0) \quad i = 1, 2, 3 \quad (2.1)$$

2.4.2. SPATIAL PATTERN OF DISEASED PLANTS

A statistical method, derived from the two-dimensional distance class analysis of GRAY *et al.* (1986b), was used to analyze simulated and observed spatial patterns of diseased plants. The authors developed a method to compare the frequency distribution of distances between any combination of two infected plants in a field with those of several hundred 'simulated fields', which were generated by stochastic sampling under the assumption of randomness in space. The analysis allows significance testing of the deviation from randomness for each distance between infected plants. Due to high computation time requirements, the method of GRAY *et al.* (1986b) had to be simplified.

The analysis method applied in this study is based on the following assumptions: the main infection sources are inside the field. Vectors landing on these sources can acquire the virus and subsequently move to healthy plants, which might become infected. If the dispersal mechanism of the vectors is completely random, any healthy plant in the field would have the same probability to become infected. In most plant-virus systems however, plants near infection sources become infected more frequently than plants lo-

¹Ordinary meaning of opened or closed intervals, e.g. the interval $[a,b)$ means that only a but not b is included in the interval.

cated farther away, thus creating more or less steep disease gradients (THRESH, 1976). Experiments on the spread of PVY show that the infections occurring during a growing season (primary infections¹) tend to be clumped around the secondary infection sources (GREGORY & READ, 1949; THRESH, 1976), i.e. the plants, emerging from infected tubers, which can serve as infection sources from the emergence of the crop. Thus, contrary to the method of GRAY *et al.* (1986b), not all infected plants, but the secondary sources were chosen as origins for the analysis. The disease incidence observed at a certain distance from a secondary infection source was compared with the expected disease incidence under the assumption of complete randomness.

At the times t_1 to t_3 the plants were classified into four categories: secondary infection sources (δ_s), plants with at least one infected tuber in the sample (subsequently called 'plants with infected tubers', τ_i), plants with no infected tuber in the sample (subsequently called 'plants without infected tubers', τ_h) and removed plants (δ_r). Plants of states τ_i and τ_h were healthy at crop emergence (resp. at simulation start) and could potentially become infected. They will be called 'infectible plants'. Removed plants were not considered in the analysis, since they could neither serve as infection sources nor become infected. Field experiment results were analyzed by classifying plants with at least one infected tuber out of n_t sampled as τ_i and plants without infected tubers as τ_h . In the simulation experiments this classification was performed by a stochastic sampling procedure: given the number of tubers sampled per plant n_t and the proportion of infected tubers of this plant t_{XY} , the probability that no infected tuber will be sampled is $(1-t_{XY})^{n_t}$ and the probability of the complementary event that one or more tubers in the sample are infected is thus $1-(1-t_{XY})^{n_t}$. Uniformly distributed random numbers u were generated in the interval $U(0, 1)$ and the infectible plant was classified into category τ_i if $u \leq 1-(1-t_{XY})^{n_t}$, otherwise into category τ_h .

The distance d_{st} between any secondary infection source (category δ_s) in the field and any infectible plant (τ_i or τ_h) was calculated using the distance between rows d_r and the distance between plants in the row d_p (Fig. 2.4):

$$d_{st} = \sqrt{((tx-sx)d_r)^2 + ((ty-sy)d_p)^2} \quad (2.2)$$

where $[sx, sy]$ are the cell coordinates of the secondary infection source and $[tx, ty]$ those of the infectible plant (sx resp. tx is the number of the row in the field and sy resp. ty the number of the plant in the row). The distances d_{st} were classified into 17 classes; the class limits were calculated as multiples of the distance between plants in a row d_p (Tab. 2.2).

¹This definition of primary and secondary infections, which is used for potatoes, is unusual for virus diseases. Virus spread in potatoes starts at the secondary infection sources present at crop emergence, causing primary infections to initially healthy plants. Primary infections become secondary infections in the following year.

Tab. 2.2: Upper limits of the distance classes.

Class number c	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Upper limits l_c of the distance classes	d_p	$2d_p$	$3d_p$	$4d_p$	$5d_p$	$7d_p$	$10d_p$	$15d_p$	$20d_p$	$25d_p$	$30d_p$	$40d_p$	$50d_p$	$60d_p$	$80d_p$	$100d_p$	> $100d_p$

The distance d_{st} was classified into a distance class c so that $l_{c-1} < d_{st} \leq l_c$. Fig. 2.4 shows an example.

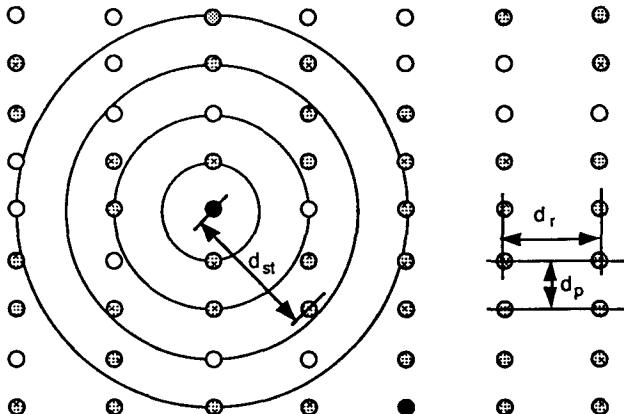


Fig. 2.4: Example of the classification of a distance d_{st} between a secondary infection source and an infectible plant. d_{st} is calculated using the row distance d_r and the plant distance within the row d_p . Black circles = secondary infection sources, grey circles = plants with infected tubers, white circles = plants without infected tubers (in the sample). The distance shown would be classified into class 3. Note that the rows in the field are represented vertically.

Let i_{cso} resp. h_{cso} be the number of plants with resp. without infected tubers in a distance class c from a secondary infection source s in a simulation run o . Given the mechanism of virus spread would be completely random in space, the number of plants with infected tubers i_{cso} in distance class c would be a sample from a binomial distribution with the expected value:

$$e_{cso} = \frac{i_{cso}}{i_{cso} + h_{cso}} (i_{cso} + h_{cso}) \quad (2.3)$$

where $i_{cso} = \sum_{c=1}^{17} i_{cso}$ and $h_{cso} = \sum_{c=1}^{17} h_{cso}$ are the number of plants with resp. without infected tubers in the whole field.

The quotient in Eq. 2.3 is the fraction of the infectible plants in the whole field having infected tubers (disease incidence), which is multiplied by the number of infectible plants in class c (sum in parentheses).

Taking the other secondary infection sources into account, the expected number of plants with infected tubers in class c would be:

$$e_{c,o} = \frac{i_{c,o}}{i_{..o} + h_{c,o}} (i_{c,o} + h_{c,o}) \quad (2.4)$$

where $i_{..o} = \sum_{s=1}^{n_s} i_{s,o}$, $h_{..o} = \sum_{s=1}^{n_s} h_{s,o}$, $i_{c,o} = \sum_{s=1}^{n_s} i_{c,s,o}$ and $h_{c,o} = \sum_{s=1}^{n_s} h_{c,s,o}$ are the sums over all n_s sources.

Since analyses of single simulation runs gave very variable results, the results of n_o simulation runs were summarized. The expected number of plants with infected tubers in class c is then:

$$e_{c..} = \frac{i_{..}}{i_{..} + h_{..}} (i_{c..} + h_{c..}) \quad (2.5)$$

where $i_{..} = \sum_{o=1}^{n_o} i_{..o}$, $h_{..} = \sum_{o=1}^{n_o} h_{..o}$, $i_{c..} = \sum_{o=1}^{n_o} i_{c,o}$ and $h_{c..} = \sum_{o=1}^{n_o} h_{c,o}$ are the sums over all n_o simulation runs.

The expected value $e_{c..}$ was compared with the realized number of plants $i_{c..}$ in a distance class c in the analysis of a field experiment (with $n_o = 1$) or simulation experiment ($n_o \geq 1$):

$$f_c = \frac{i_{c..}}{e_{c..}} \quad (2.6)$$

The ratio of observed and expected number of plants with infected tubers in class c (f_c) gives a measure of the deviation from a random distribution in distance class c (see Fig. 2.5). $f_c > 1$ indicates that the infections occurred more frequently than expected in the random model, $f_c < 1$ that they occurred less frequently and $f_c = 1$ that the occurrence corresponded to what would be expected in the random model. The 17 f_c values can be plotted in a histogram (see Fig. 2.5). The set of these values will be called '*distance class distribution*'.

To summarize the information of the distance class distribution, the evenness E was calculated, an index used in ecology to assess diversity of species communities (MAGURRAN, 1988). This value can serve as an indicator of the deviation of the distance class distribution from uniformity and thus as an indicator of the deviation from randomness in space. It is based on the Shannon-Wiener index H , originating from information theory.

$$H = - \sum_{c=1}^{17} (p_c \ln(p_c)) \quad (2.7)$$

$$p_c = f_c / \sum_{c=1}^{17} f_c$$

H is a measure of 'diversity' or 'uniformity' of a distance class distribution. The smaller H the more the spatial pattern deviates from randomness. The evenness E is the ratio of the calculated H to H_{max} , the maximal H .

$$E = \frac{H}{H_{max}} = \frac{H}{\ln(17)} \quad (2.8)$$

The evenness E resp. H is maximal, when all p_c -values are equal, which implies that also all f_c -values are equal. This can only be the case if $f_c = 1$ for all c , which indicates that the spatial distribution is random.

E is constrained between 0 and 1. $E = 0$ corresponds to maximal unevenness (only one class occupied), $E = 1$ to maximal evenness, i.e. a uniform distance class distribution. Two examples (Fig. 2.5 and 2.6) shall illustrate the analysis:

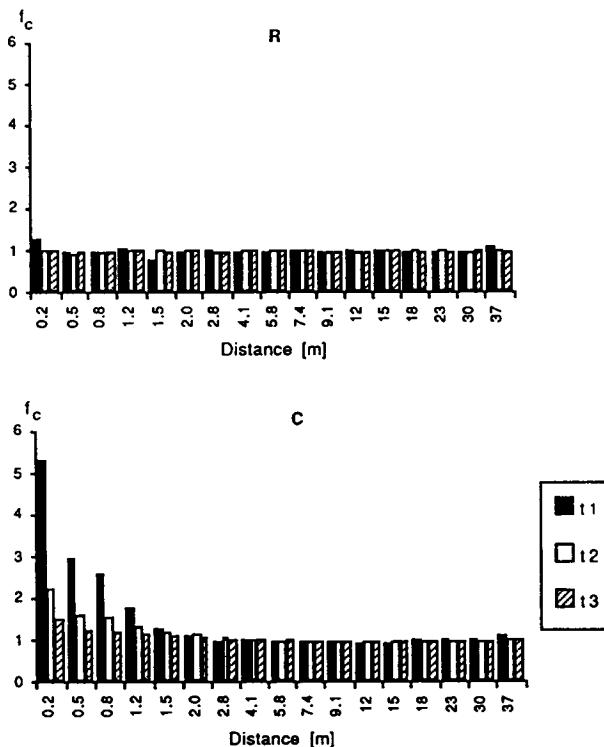


Fig. 2.5: Distance class distribution at t_1 to t_3 of a random pattern (R) and a clumped pattern (C) of plants with infected tubers (10 runs). The labels of the x-axis denote the class middle.

A random spatial pattern was produced by simulations of the EPOVIR-model, where inoculations occurred at any site in the field with equal probability. The f_c -values deviated little from the theoretically expected value of 1 (Fig. 2.5, R). The larger deviations at t_1 were due to the small number of plants with infected tubers present at this time. The evenness values were close to 1 (Fig. 2.6, R).

The 'clumped' pattern was produced by simulations, where the probability of infection decreased with increasing distance from the infections sources (standard run of the EPOVIR-model). The classes in short distance of the sources had f_c -values about 5 (Fig. 2.5, C). This means that infected plants were encountered five times more frequently in the first class at t_1 than expected in the random model, because inoculations at short distances were more likely to occur than at longer ones. The corresponding evenness values were lower than 1 (Fig. 2.6, C).

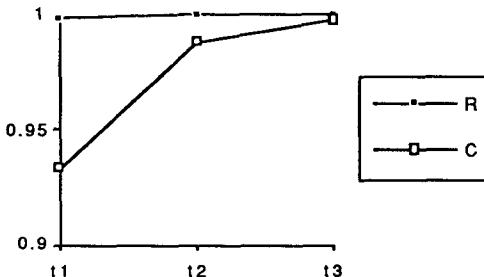


Fig. 2.6: Evenness values E calculated for the patterns in Fig. 2.5. R = random pattern, C = clumped pattern.

2.5. Sensitivity Analysis

Only one parameter was changed at a time in the sensitivity analysis. Six values (p_{n1} to p_{n6}) in the plausibility interval $[p_{nmin}, p_{nmax}]$ were assigned to the parameter p_n , having the standard value \bar{p}_n :

$$p_{ni} = \begin{cases} \bar{p}_n + \Delta_i (p_{nmin} - \bar{p}_n) & i = 1, 2, 3 \quad \Delta_i = 1, 0.5, 0.2 \\ \bar{p}_n + \Delta_i (p_{nmax} - \bar{p}_n) & i = 4, 5, 6 \quad \Delta_i = 0.2, 0.5, 1 \end{cases} \quad (2.9)$$

For many parameters the plausibility interval was asymmetrical. Six simulation experiments were performed for each parameter p_n . The output variables p_t (proportion of infected tubers) and E (evenness of the distance class distribution) were monitored at t_1 to t_3 . y_n shall denote any output variable. Because the output variables are proportions, their values were transformed by the logit transformation, which made their distributions approximately normal and the logit of the output value \bar{y} in the standard run was subtracted:

$$y_n' = \ln \frac{y_n}{1-y_n} - \ln \frac{\bar{y}}{1-\bar{y}} \quad (2.10)$$

The transformed model outputs y_n' were fitted by a linear regression model without y-axis intercept, i.e. the regression was forced through the standard value:

$$y_n' = b_n (p_n' - 1) \quad (2.11)$$

using the standardized parameter value $p_n' = p_n / \bar{p}_n$ as independent. The standardization of the parameter values allowed comparisons of the magnitudes of the regression coefficients b_n .

Fig. 2.7 shows an example of the analysis of the results (see Eq. 2.11). The regression coefficients b_n allowed comparison and ranking of parameter sensitivities. A large b_n indicates high sensitivity of the output variables to parameter p_n , the sign of the coefficient indicates, whether the output variable is positively or negatively correlated to p_n .

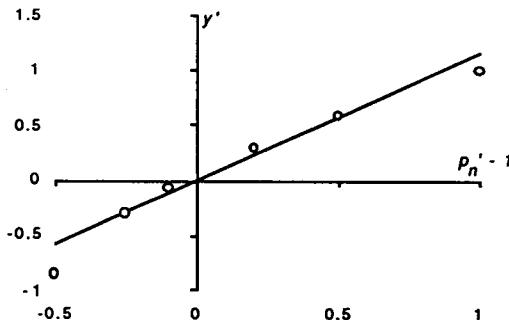


Fig. 2.7: Representation of a regression through sensitivity analysis results. y' = logit-transformed and standardized values of the output variables, p_n' = standardized parameter value.

2.6. Statistical Computations

Parameter estimation and statistical analysis procedures were performed on Apple™ Macintosh™ personal computers, models II or IIfx. Unless stated otherwise, the statistical package SYSTAT (Systat Inc., versions 3.1 or 5.0, modules Corr, MGLH, Nonlin, Npar and Tables) and StatView II (Abacus Concepts, Inc.) were used.

2.7. Field Data Sampling

The data of the field experiments described in this subchapter were used to estimate model parameters, as input data or as validation data for the EPOVIR-model (Chapter 6 & 7).

2.7.1. EXPERIMENTAL FIELDS

The field experiments were carried out at the Swiss Federal Agricultural Research Station of Changins in Western Switzerland. The research station is located at an altitude of 430 m above the sea level between the border of the Lake of Geneva (south-east of the research station) and the Jura mountain chain (north-west of the research station) at a latitude of 46.4 ° North and a longitude of 6.5 ° West.

2.7.2. EPIDEMIOLOGICAL EXPERIMENTS

Eleven epidemiological experiments were carried out between 1983 and 1990 with the variety Bintje. The results were used for spatial and temporal validation of the EPOVIR-model by comparing the following measurements with model outputs: proportion of tubers showing a positive reaction in the *enzyme-linked-immunosorbent-assay*

(ELISA) (GUGERLI, 1979), subsequently called *infected tubers*, proportion of plants with infected tubers and proportion of leaf resp. stem samples showing a positive reaction in ELISA. Prior to testing by ELISA, the tubers were treated with Rindite and presprouted to break the dormancy. This procedure, which was applied in all tuber tests, allowed the virus to multiply and increased the detection probability (REK, 1987). The ELISA on leaf and stem samples was performed immediately after sampling. The 'basic infection' p_{ib} , i.e. the initial disease incidence in the field surrounding the experimental plot, usually differed from the initial disease incidence p_i within the experimental plot. p_{ib} was determined by an ELISA-test of a sample of 100 tubers. Tab. 2.3 summarizes the experiments.

The distance between rows d_r was 0.75 m, the distance between plants in a row d_p was 0.33 (except in 1986, where $d_p = 0.35$ m). The previous crop was wheat or triticale. The herbicide Topogard™ (Terbutryn & Terbuthylazin) was applied before emergence. No plants were rogued and no aphicides were applied in the experimental plots. Treatments against colorado beetles (*Leptinotarsa decemlineata* SAY) were performed with Teflubenzuron, which does not influence aphid populations (Dr. J. Derron, Federal Agricultural Research Station, CH-1260 Nyon, Switzerland, pers. comm.). The plots were regularly treated against *Phytophthora infestans* DE BARY with Mancozeb, Chlorothalonil or Metalaxyl. Haulm-killing was performed by a DNOC or Dinoseb-application.

Tab. 2.3: Epidemiological experiments used for validation. t_{em} = date of 50% emergence, t_{hk} = haulm-killing date, p_i = initial disease incidence, p_{ib} = proportion of infected plants in the area surrounding the experimental plot ('basic infection'), n_t = tuber sample size, P = number of the plot at the research station in Changins.

Data frame	Year	Planting date	t_{em}	t_{hk}	p_i	p_{ib}	Virus strain	n_t	P	Fertilization [kg/ha]			
										N	P ₂ O ₅	K ₂ O	Manure
1983n1	1983	25.4	16.5	20.7	0.1	0.0	PVYn 605	3	41	120	100	300	
1984n1	1984	12.4	3.5	23.7	0.1	0.1	PVYn 605	3	4	120	55	210	40000
1985n1	1985	3.4	1.5	22.7	0.1	0.02	PVYn 605	3	19	120	80	240	40000
1986n1	1986	5.5	16.5	16.7	0.1	0.01	PVYn 605	3	14	120	100	300	
1986n1	1986	5.5	16.5	16.7	0.0159	0.01	PVYo 803	5	14	120	100	300	
1987n1	1987	22.4	10.5	14.7	0.1	0.01	PVYn 605	3	26	140	90	270	
1987n1	1987	22.4	10.5	24.7	0.0204	0.01	PVYo 803	5	26	140	90	270	
1988n1	1988	21.4	11.5	6.7	0.0583	0.0	PVYo 803	3	44	140	90	270	
1989n1	1989	3.5	19.5	5.7	0.0222	0.0	PVYo 803	5	42	150	90	270	
1990n1	1990	28.3	3.5	4.7	0.0909	0.011	PVYo 803	3	41	140	102	230	
1990n2	1990	28.3	3.5	20.6	0.025	0.011	PVYo 803	3	41	140	102	230	

2.7.3. ARTIFICIAL INOCULATION EXPERIMENTS

Three inoculation experiments were performed from 1985 to 1987 with the variety Bintje. In 1985 the strain PVYn 605 was used. In 1986 and 1987 two parallel experiments with PVYn 605 and PVYo 803 were conducted. Prior to inoculation, leaf samples were tested by ELISA (see 2.7.2) and plants showing a positive reaction were eliminated. Groups of ten plants were inoculated at different dates (intervals of one week in 1985, one to two weeks in 1986 and two weeks in 1987). One of the fully developed top leaves

was dusted with carborundum powder to increase the infection success. The experimenter dipped then one finger in sap from infected tobacco leaves and rubbed the sap on the potato leaf. The inoculated stem was labelled, to ensure that leaf samples were taken from the same stem. Simultaneously with the inoculations, three leaf samples were taken per inoculated stem from the plants that were inoculated at the earlier dates. Three tubers per plant were sampled at harvest. All samples were tested by ELISA. The planting, emergence and haulm-killing dates as well as the fertilization are given in Tab. 2.3 (1985n1, 1986n1 and 1987n1).

Since no control without artificial inoculation has been realized, the proportion of infected tubers provoked by natural infections was estimated by simulation with the EPOVIR-model (Chapter 6), using p_{ib} (Tab. 2.3) as initial disease incidence. The proportion of infected tubers in the experiment p_{it} was corrected by the simulated control p_{is} :

$$p_{it}^* = \frac{p_{it} \cdot p_{is}}{1-p_{is}} \quad (2.12)$$

p_{it}^* is the proportion of infected tubers corrected for natural infections.

2.7.4. MONITORING OF APHID ABUNDANCE

Aerial aphid abundance was estimated by a Taylor suction trap (TAYLOR & PALMER, 1972), located at the Swiss Federal Research Station in Changins (DERRON & GOY, 1987). The trap collects 42.8 m³ air per min at the height of 12 m and filters out the aerial plankton. The aphids are sampled continuously in intervals from midnight to midnight, counted and identified.

2.7.5. METEOROLOGICAL DATA

Data recorded by a meteorological station, located at the Swiss Federal Agricultural Research Station Changins, were used as model inputs. The meteorological station is part of the automatic meteorological recording network (ANETZ; SMA, 1980 & 1985). Data are recorded in intervals of 10 min and then averaged resp. integrated over time to obtain hourly and daily means. The experimental fields had a distance up to 2 km from this station.

The following daily (D) resp. hourly (H) values were used as model inputs: total precipitation ([mm], D & H), average rel. humidity ([%], D), average wind speed ([m/s], D & H), average wind direction ([rad], H), daily minimal and maximal temperature ([°C], D), average temperature ([°C], H), total global radiation ([Wh/m²], D).

3. THE PVY-PATHOSYSTEM

3.1. Real System

Only certain components of the agroecosystem are of interest for the study of plant disease epidemics; the latter is defined as the 'change of disease intensity in time and space' (CAMPBELL & MADDEN, 1990). ROBINSON (1976) defined the subsystem of an agroecosystem involved in plant disease epidemics as a 'pathosystem'. The pathosystem of an aphid-borne virus disease consists of the elements virus, the virus host plant, the aphid vectors, the natural environment and man (Fig. 3.1).

In the following the three elements virus, potato plant and aphid vectors will be briefly described. In a second step the relations between these elements, and impacts of the elements natural environment and man on these relations will be discussed. The analysis shall concentrate on the PVY-pathosystem in the seed potato production regions of Western Switzerland, although most considerations are also valid for many other production schemes in other regions.

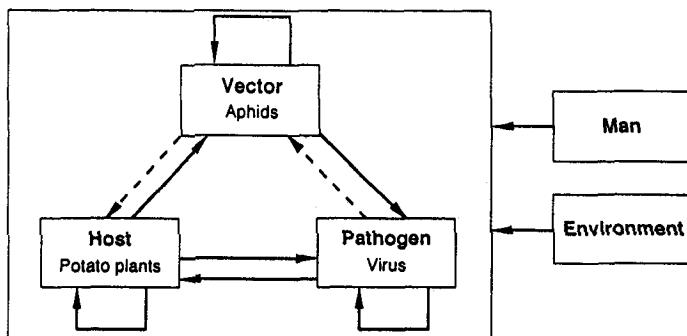


Fig. 3.1: Relational graph (for the predicate $P(x, y) ::= x \text{ influences } y$) of the pathosystem of an aphid-borne virus disease. Dashed arrows indicate relations, which are irrelevant in most systems and are ignored in this study.

3.1.1. SYSTEM ELEMENTS

Potato Crop

Since PVY sources other than potato plants are considered to be unimportant for PVY epidemics in potatoes (CARTER & HARRINGTON, 1991), they were dropped from subsequent analysis.

In Switzerland the potato is usually propagated vegetatively by tubers, which allows most viruses to overwinter in the tubers and to serve as initial infection sources

(secondary infection sources) in the following season. Tubers are usually pre-sprouted to favour early emergence and planted in the field in April. Depending on the winter storage and the variety the mother tuber forms one or several stems, developing 10-15 leaves each. In many varieties flowers are lacking or scarce. After tuber initiation a rapid increase in tuber mass occurs. This absorbs most of the produced assimilates.

Potato varieties differ among other in their tuber yield, their disease resistance, e.g. to PVY and PLRV, and in their maturity (REUST, 1991).

Aphids

Under Swiss conditions aphids can be holocyclic or anholocyclic. In the first case they overwinter in the egg stage on their primary hosts, usually a woody plant. In spring they migrate to secondary hosts, which are mostly herbaceous plants, e.g. potato. In autumn they migrate back to their primary hosts. Anholocyclic species remain on secondary hosts during the whole year or migrate between them. Anholocyclic aphid populations can arrive earlier in potato crops, but are more subject to cold winter temperatures than holocyclic ones (CARTER & HARRINGTON, 1991).

Potato Virus Y (PVY)

PVY belongs to the group of the potyviruses, to which it gives its name, i.e. potato virus Y group. Viruses of this group can be transmitted nonpersistently by aphids or mechanically by artificial inoculation (SYLVESTER, 1989). The latter is believed to be unimportant under field conditions (BEEMSTER & DE BOKX, 1987). According to the symptoms they cause, two main virus strain groups are distinguished: the ordinary or common strains (PVY_O) and the tobacco veinal necrosis strains (PVY_N). Other strain groups are of little economic importance.

3.1.2. SYSTEM STRUCTURE

Aphid-Potato Plant Interaction

After the last moult, an alata needs a period of about 12 h, before it is able to fly (MÜLLER & UNGER, 1955). If the conditions are favourable for flight (temperature not too low, no heavy rainfall, not too high wind speed; DIXON & MERCER, 1983; WALTERS & DIXON, 1984; ROHITHA & PENMAN, 1986), the alatae take-off and fly upwards, attracted by short wave light of the sky. This initial flight is called 'distance flight' (MÖRICKE, 1955) or 'migratory flight'. Thereafter, aphids react to long waved light emitted by the earth's surface, and are especially attracted by yellow objects. Landing on plants is often followed by frequent changes between walking and probing (KLINGAUF, 1987). Most of the alatae continue to walk and probe and finally leave the plant, even if the plant belongs to a host species (KENNEDY *et al.*, 1959). Only a small proportion of the aphids settles. After several visits to host plants, the aphid is likely to settle and to reproduce. Thereafter many species autolyse their flight muscles and lose the ability to fly (DIXON, 1985). Most alatae need a distance flight before reproduction (ROBERT, 1987).

Aphids that cannot find a suitable host are likely to continue flying and searching for a host until death (ROBERT, 1987).

Under Swiss conditions, *M. persicae*, *Aphis nasturtii* KALTENBACH, *Macrosiphum euphorbiae* THOMAS and *Aulacorthrum solani* KALTENBACH colonize the potato crop and are also able to transmit PVY. Alatae of these species immigrate during May and June into potato fields and start reproducing on the canopy (PETERS, 1987). Several generations of wingless aphids follow. The populations on host plants typically grow exponentially, doubling each week on the average, usually followed by a breakdown (DERRON, unpubl. data; STORCH, 1981). The latter is due to several factors: aphid abundance, changes in plant physiology, predators, parasitoids, pathogens and the production of winged aphids that emigrate. Emigrating alatae can produce new colonies elsewhere.

Aphid-Virus Interaction

DE BOKX & PIRON (1990) found that 26 aphid species transmitted PVY in their experiment. Including results from other authors, they present a list of 33 PVY vector species; most of them are noncolonizing. Genetically determined differences in virus transmission exists not only between species, but also between different clones within a species (ROCHOW, 1974; SINGH *et al.*, 1983).

The process of PVY transmission has been shown to depend on the presence of a plant produced 'helper'-component (GOVIER & KASSANIS, 1974; GOVIER *et al.*, 1977). Aphids cannot transmit PVY from purified preparations of virus particles, where this helper-component has been removed.

After PVY acquisition, infectivity can be retained for less than 1 h on average (VÖLK, 1959; KOSTIW, 1975; PRÖSELER & WEIDLING, 1975; VAN HOOF, 1980; KATIS & GIBSON, 1985). The data indicate that infection probability of a fasting aphid declines exponentially. The probability decreases faster in probing than in fasting aphids (BRADLEY & RIDEOUT, 1953). PVY transmission probability is reduced by longer acquisition or inoculation periods (BRADLEY & RIDEOUT, 1953; VÖLK, 1959; PRÖSELER & WEIDLING, 1975). It has a temperature optimum between 15 and 25 °C (PRÖSELER & WEIDLING, 1975; SINGH & KHURANA, 1987).

Plant-Virus Interaction

PVY in potato tubers does not endanger human health, but PVY infection depresses plant growth, leading to reduced tuber yield (VAN DER ZAAG, 1987). The yield reduction depends on whether the infection is secondary or primary. 100 % secondary infection by PVYn in a field can cause a tuber yield loss of 15-30 %, 100 % secondary infection by PVYo a yield loss of 40-70% (VAN DER ZAAG, 1987). Primary infections cause much smaller yield losses than secondary infections and are only relevant if the infections occur early. Because tuber yield reduction of virus infected plants is of great importance for ware crops, tuber infection thresholds for viruses were defined in seed potato certification (see below). In seed potatoes however, the yield reduction due to virus infection can be ignored, because the fraction of secondarily infected tubers should be below 1 %.

After a successful infection of the plant, virus multiplication and virus translocation occur. The presence of virus particles in the plant tissues can be detected by the enzyme-linked immunosorbent assay (ELISA) (GUGERLI, 1979). Infection early in the season

leads to more tubers becoming infected than late infection (BEEMSTER, 1969, 1976 & 1987; BRABER *et al.*, 1982; SIGVALD, 1985; GIBSON, 1991). This phenomenon is called 'mature plant resistance' or 'age resistance'. Its causes are not known, but it is probably related to the physiological activity of the tissues (VENEKAMP *et al.*, 1980).

Impact of Man: Seed Potato Production

To prevent high crop losses in ware potatoes, seed with a low proportion of infected tubers is used. The production of such seed is difficult, especially in regions with high aphid abundance like Switzerland. To avoid too high contamination with virus diseases, much effort is needed. Currently the following measures are in use (HÄNI & WINIGER, 1987; SCHWÄRZEL & GEHRIGER, 1989): choice of seed production fields at higher altitudes and sites exposed to wind, isolation of seed potato fields from ware potato fields to avoid virus transmissions from the latter (which can be highly contaminated with viruses), usage of seed with less than 1 % infected tubers, roguing, i.e. removal of plants with virus disease symptoms, two field inspections (where the crop is examined for general appearance as well as for the presence of diseases), early harvesting by applying a high dose of herbicides. The haulms of seed potatoes are destroyed in July, those of ware potatoes in September. Early harvesting can reduce tuber infection (KELLER, 1958). Fixing of early harvesting dates can be very critical, since a rapidly growing tuber yield can be accompanied by an increase of tuber infections by viruses. Depending on the field surface, a sample of 45 to 450 tubers is tested by ELISA. A seed stock is certified, if the fraction of virus infected tubers does not exceed 10 % (class A) or 16 % (class B) and the field had not been rejected during the field inspections. Aphicides are not used to control virus vectors in Switzerland (HÄNI & WINIGER, 1987).

In 1991 potatoes were grown on a surface of 18100 ha in Switzerland. The fraction reserved for seed potatoes was 2300 ha. In the last years PVY accounted for about 2/3 of the virus diseases, PLRV for most of the rest.

Impact of Man: Control Measures Affecting Aphid Behaviour

Aphids respond to various stimuli of their environment. These stimuli can be modified by man in order to influence aphid behaviour (GIBSON & RICE, 1989). The following techniques were selected from many others, because of their potential to control or otherwise affect virus epidemics. The first method acts on immigration into a field, all others on the behaviour of aphids in the field.

Measures affecting immigration: Immigration into a field can be reduced most effectively by reflective surfaces, which repel aphids by emitting short-waved light (KRING, 1972; LOEBENSTEIN & RACCAH, 1980; GIBSON & RICE, 1989). LOEBENSTEIN & RACCAH (1980) reported that the number of *M. persicae* captured in fields with aluminium foil or plastic sheets could be reduced by over 99 % in the first three weeks after crop emergence. The effectiveness of the method declines during the growing season, due to successive covering by the crop canopy. Due to high costs, reflective surfaces are hardly applied in seed potatoes (WOODFORD, 1979).

Aphicides: Apart from the desired increase in mortality of the target organisms, aphicide treatments can also have secondary effects on other parts of the ecosystem, such as reduction of predator populations, thus making repeated aphicide applications neces-

sary (SCHEPERS, 1989). As another secondary effect, aphicides can induce changes in aphid behaviour. Certain aphicides can influence behaviour of aphids before killing them, or behaviour of individuals, which are not killed due to aphicide resistance or to sublethal doses (GIBSON & RICE, 1989). GABRIEL *et al.* (1981) observed a significantly increased PVY spread in potatoes treated with dimethoate and pirimicarb and attributed it to an increased restlessness of the aphid vectors after the insecticide treatment. Depending on the product, probing frequency and duration can be increased or decreased (SHANKS & CHAPMAN, 1965; LEHMANN *et al.*, 1975; SASSEN, 1983; LOWERY & BOITEAU, 1988). Likewise, walking and flight frequency can be increased or decreased (SHANKS & CHAPMAN, 1965; RICE *et al.*, 1983; LOWERY & BOITEAU, 1988). Increased restlessness might be due to the fact that treated surfaces attract or repel aphids (SHANKS & CHAPMAN, 1965; LOWERY & BOITEAU, 1988). Due to these complex relationships, the effects of aphicides on virus dissemination are difficult to assess.

Repellents: GREENWAY *et al.* (1978) have tested various carboxylic acids, extracted from aphid bodies and found that the undecanoic and dodecanoic acids (fatty acids with 11 resp. 12 C-atoms) had the strongest repellent effects. These acids can reduce settling, increase probing and walking frequencies and reduce the total probing time and the time spent on the treated surfaces (SHERWOOD *et al.*, 1981; GIBSON *et al.*, 1982; PHELAN & MILLER, 1982).

Alarm pheromones: An attack by predators induces aphids to release an alarm pheromone via the cornicles. Aphids receiving the alarm pheromone are likely to disperse rapidly. (*E*)- β -farnesene was identified as the main component of aphid alarm pheromones (HERRBACH, 1985). Alarm pheromone components did not reduce the number of landings on treated surfaces (PHELAN & MILLER, 1982), probably because the distance at which aphids react to the alarm pheromones is short (NAULT & MONTGOMERY, 1977), but caused aphids to walk or fly off the plant. Probing was not prevented, the probing frequency was even increased. Further, the mean duration of walking increased, whereas the mean probing duration decreased. PVY spread in the field could not be reduced (HILLE RIS LAMBERS & SCHEPERS, 1978). Results of laboratory experiments are contradicting, showing reduction of PVY transmission (DAWSON, *et al.*, 1982; GIBSON *et al.*, 1984) or no effect (YANG & ZETTLER, 1975).

Plant resistance: HARREWIJN (1986) distinguished two types of plant resistance against aphids in respect to virus epidemiology. An aphid resistant plant prevents the aphids from multiplying. This can be due to *antixenosis*, which means non-acceptance or non-preference of the host plants by the aphids, or *antibiosis*, e.g. higher mortality or reduced fecundity. A vector resistant plant inhibits virus transmission. Vector resistance can be due e.g. to reduced or inhibited probing. Reduced probing frequency was observed on some aphid resistant potato varieties (HENNIG, 1969; TARN & ADAMS, 1982).

In many reports, however, it is not possible to determine what kind of resistance has been involved. Plant resistance to aphids can lead to decreases and increases of non-persistent virus incidence (KENNEDY, 1976). Probing frequency and dispersal can be increased on resistant plants (GRAY *et al.*, 1986a; HARREWIJN, 1986). Mainly the behaviour of colonizing species is likely to be affected by plant resistance (ROMANOW, 1985).

3.1.3. SYSTEM BEHAVIOUR

PVY is introduced into a potato field by infected tubers (secondary infections) or by winged aphids, which are virus carriers. The initial PVY incidence is related to the disease incidence and severity in the preceding year. After years with a high virus incidence, seed with few infected tubers is rare and expensive. Once PVY sources are present in the field, the disease is transmitted by winged and wingless vectors. Virus infections often occur in the proximity of the infection sources (GREGORY & READ, 1949; GABRIEL *et al.*, 1981), causing a spatially nonrandom pattern. After having passed the latent period¹, plants infected during the growing season can serve as sources for further virus spread.

The epidemics of potato viruses depend on the coincidence of the age resistance and aphid flight. In suitable seed potato production regions, age resistance prevents most of the virus infections in the later part of the growing season, when the aerial aphid abundance is high. However in some years a strong flight occurs early in the season, leading to a high disease incidence.

3.2. Model Systems

3.2.1. RESOLUTION LEVELS AND SYSTEM BOUNDARY

The PVY-pathosystem can be analyzed at different planes of abstraction in time, space and biological organization. ZADOKS & SCHEIN (1979) have proposed three 'levels of integration' or resolution levels on the different scales, which are summarized in Tab. 3.1.

Tab. 3.1: Levels of integration proposed by ZADOKS & SCHEIN (1979).

Processes	Temporal scale	Organization	Spatial scale	Focus of interest
Monocyclic	hour	organism	millimeter	phylosphere/rhizosphere
Polycyclic	day	population	meter	crop
Polyetic	year	community	kilometer	agro-ecosystem

The authors distinguish three different types of epidemic processes: monocyclic, polycyclic and polyetic. A monocyclic process, also called 'infection cycle', is the process from infection until the infected plant material can serve as an infection source. A process consisting of several infection cycles is called polycyclic. Epidemic processes over several years are called polyetic (ZADOKS & SCHEIN, 1979). Since infected plants can serve as infection sources during the same growing season (BEEMSTER, 1979), PVY epidemics are polycyclic. The temporal, spatial and organizational levels are interrelated (see Tab. 3.1). To keep the models simple, the resolution level should not be increased beyond what is needed by the research question. The available knowledge and data about the real system also limit the resolution level. In addition to the resolution level the system boundary should be defined on each scale.

¹Throughout this thesis, the phytopathological definition of the 'latent period' of the virus in the plant will be used. The latent period is the time between the infection and the instant when the plant becomes an infection source (ZADOKS & SCHEIN, 1979).

Time Scale

PVY epidemics take place during the growing season. Epidemic processes during the winter can be ignored in cold temperate regions, since the winter is a natural break in the epidemic cycle. Moreover, the epidemic starts each year with a known initial disease incidence, which can also be influenced by selecting high quality seed. One growing season seems to be a natural time domain in cold temperate regions, i.e. the time interval during which models are simulated. All epidemic field data were sampled for one growing season and no experiments are available studying epidemics over more than one growing season (Chapter 2.7.2). Since the significance of the epidemic processes can be judged only after one growing season from a practical point of view, a smaller time domain than one growing season seems inadequate. A growing season was hence chosen as the time domain.

The time step, i.e. the temporal resolution, is given mainly by the modelled processes and can vary between seconds and several weeks. Individual aphid behaviour has a time coefficient of few seconds, whereas most other relevant processes, such as plant growth, are in the order of days. A time resolution one day seems more adequate for the study of an epidemic process during several weeks than a resolution of one second. This aspect is to be considered in the choice of the organizational resolution.

Organizational Scale

Since PVY particles do not occur outside of plants or aphids, it is not necessary to consider virus particles explicitly in models. The virus can be included implicitly as attributes of plants or aphids (e.g. infected or infectious), which is also done in all virus epidemic models. Plants and aphids can be represented at the individual and the population level. Behaviour sequences of animals are usually quantified at the level of individuals (SLATER, 1978). However, a model of individual aphid behaviour would require a high degree of resolution and a time step of a few seconds. To solve this problem, behaviour of whole aphid populations is modelled. The organizational level for the plants is coupled with the spatial resolution of the field.

Spatial Scale

A single field was preferred to a whole region as spatial system boundary for two reasons: first, experiments where vector behaviour was modified were performed for single fields and little is known about possible effects on vector behaviour and on virus dissemination over larger areas. Although considered to be important (HARREWIJN *et al.*, 1981), virus spread between fields has been little investigated, and only occasionally it has been possible to study aphid migration over longer distances (TATCHELL & WORWOOD, 1989). Second, data sets for model calibration and validation were collected for single fields only (Chapter 2.7.2).

Since vector behaviour affects temporal and spatial virus epidemics (GRAY *et al.*, 1986a; IRWIN & KAMPMEIJER, 1989), a study of the role of vector behaviour in PVY epidemiology should include both aspects. The spatial pattern of infected plants can be modelled by assuming a spatial distribution of infected plants, e.g. Poisson or negative binomial (WAGGONER & RICH, 1981; RUESINK & IRWIN, 1986), or alternatively by modelling plants as points in a two-dimensional grid (BERGER & FERRISS, 1989; VAN

DER WERF *et al.*, 1989). In the former approach the spatial resolution level would be a field, whereas the latter approach requires that the spatial resolution is increased to single plants. The merits of the various approaches are discussed in Chapter 4.

3.2.2. EPIDEMIC MODELS

Most epidemic models of plant diseases known from the literature operate on the temporal and spatial system boundary levels chosen for the study, namely one growing season and one field. Some epidemic models of interest are reviewed. The meaning of the symbols is also listed in Appendix II.

Models with a Constant Infection Rate

These models are applicable to epidemics of diseases that are not vector-borne or to diseases, where the vector intensity is constant in time. KERMACK & MCKENDRICK (1927) have described the 'general epidemic model', which was analyzed by BAILEY (1975). The model is based on the following assumptions:

- I An isolated population of constant size exists.
- II The populations of healthy and infectious individuals are randomly distributed, resp. the contact rates between them are random.
- III Infected individuals are removed with a constant rate from the epidemic process (death, isolation or immunization).
- IV No latent period exists.

The model is of third order:

$$\frac{d D_h}{dt} = -i' D_i D_h \quad (3.1)$$

$$\frac{d D_i}{dt} = i' D_i D_h - \gamma D_i = i' (D_h - \rho) D_i \quad (3.2)$$

$$\frac{d D_r}{dt} = \gamma D_i \quad (3.3)$$

D_h , D_i and D_r are the proportions of healthy, infectious and removed individuals respectively ($D_h + D_i + D_r = 1$), i' is the infection rate and γ is the removal rate. Infection and removal are irreversible processes. $\rho = \frac{\gamma}{i'}$ is the so-called relative removal rate. The model can be used to determine thresholds for epidemic outbreaks. An epidemic can occur, if $\frac{d D_i}{dt} > 0$, which requires that $D_h(0) > \rho$. This model and similar ones have been widely applied in medicine, but never to plant pathology problems. By setting $D_r = 0$ ($\gamma = 0$) the 'general epidemic model' becomes the logistic model, sometimes also called 'simple epidemic model'. It can be formulated as a system of first order:

$$\frac{d D}{dt} = i' D (1-D) \quad (3.4)$$

where $D (= \frac{D_i}{D_i+D_h})$ is the disease incidence. By including immigration, birth and death processes (other than induced by the disease), ANDERSON (1981) presented a generalization of the model of KERMACK & MCKENDRICK.

Most models of plant disease epidemics stem from the work of VANDERPLANK (1963 & 1982). The author developed models for monocyclic and polycyclic epidemics. The latter is of main interest for PVY epidemics. The logistic model is extended by introducing a constant latent period m_l and a constant infectious period m_s :

$$\frac{d D}{dt} = i' (D(t-m_l) - D(t-m_l-m_s)) (1-D(t)) \quad (3.5)$$

ZADOKS & SCHEIN (1979) called i' the *corrected basic infection rate*. It corresponds to vector intensity as defined by IRWIN & RUESINK (1986) for aphid-borne virus disease epidemics. Vector intensity can be considered as the contribution of the aphids to the epidemic. In addition to assumptions I and II (see above), the model assumes that:

V Constant latent and infectious periods exist.

VI All individuals can become potentially infected.

Since in the context of a growing season virus infected potato plants neither die nor become immune, the duration of the infectious period m_s can be assumed as infinite. Eq. 3.5 becomes then:

$$\frac{d D}{dt} = i' D(t-m_l) (1-D(t)) \quad (3.6)$$

MARCUS (1990) presented a generalization of the simple epidemic model (Eq. 3.4), which accounts for inoculations coming from outside the field i_e in addition to those from inside the field i_i .

$$\frac{d D}{dt} = (i_e + i_i D) (1-D) \quad (3.7)$$

Numerous other 'growth curve' models (similar to Eq. 3.4) have been applied to epidemiological problems, among them the Gompertz and the Weibull model (ROUSE, 1985; CAMPBELL & MADDEN, 1990).

The application of the models described in this section to vector-borne diseases is limited, since vector intensity is seldom constant in time. This fact is taken into account by the next group of models.

Models with a Time-Varying Infection Rate

MADDEN *et al.* (1990) presented a class of simple models applicable to nonpersistent virus epidemics. The rate of spread i' is proportional to aphid catches in some kind of trap. The authors propose to apply the models to systems, where the vector propensity of different species is unknown.

The only simulation model specific to PVY in potatoes was developed by SIGVALD (1986). It was used to forecast the percentage of infected tubers at harvest time in Southern Sweden. The model includes a latent period, a period between infection of a plant and infection of its tubers, vector abundance in yellow water traps, vector propensity parameters and age resistance. It can be considered as a modification of Eq. 3.6, where the vector intensity i' is a function of yellow water trap catches and vector propensity. Factors for varietal and age resistance were introduced and the latent period m_l was formulated as a function of crop age.

The model of RUESINK & IRWIN (1986) was developed to describe the epidemiology of soybean-mosaic-virus (SMV), its impact on yield and the proportion of virus infected seed. The system soya-SMV is quite similar to the potato-PVY system. The model

was later modified and adapted to bean common mosaic virus in beans and to maize rayado fino virus in corn. The SMV epidemic model uses daily aphid catches in green water traps as input. The model allows to simulate a nonrandom (negative binomial) distribution of diseased plants in space and to include infective immigrating aphids. Apart from the vector propensity, a parameter for subsequent virus transmissions, the number of movements of an aphid in the field and the proportion of nonhost plants in the field were included.

The model of MARCUS & RACCAH (1986) for cucumber mosaic virus and PVY epidemics in peppers is similar to the model of RUESINK & IRWIN (1986).

The drawback of these models is that they are based on the assumption that the spatial pattern of infected plants in space is random or follows a known statistical distribution, which is seldom the case. The last group of models take the spatial pattern of the infected plants into account.

Models of Spatio-Temporal Epidemics

The model EPIMUL (= epidemics in multilines, KAMPMUIJER & ZADOKS, 1977) was developed to simulate fungal epidemics in a field with different spatial arrangements of multilines, i.e. lines of the same variety containing different resistance genes. A field is divided into squares. Fungal spores are distributed over the field in function of the distance from the source.

The model of the epidemiology of beet yellowing viruses (VAN DER WERF *et al.*, 1989; VAN DER WERF & RIESEBOS, 1990) simulates spatial virus spread by apterous aphids, which either walk or feed. The distance and direction of aphid walks is simulated. Virus acquisition and inoculation is a function of the feeding time.

The model published by BERGER & FERRIS (1989) was designed to investigate virus epidemics in a hypothetical situation, allowing for different virus transmission modes. Spatial and temporal virus dynamics can be studied. The model includes virus transmission parameters, such as latent period of virus in the vector, inoculative period of the vector, acquisition and inoculation probability and certain behaviour parameters, e.g. the distance and kind of vector movement.

Vector Behaviour in Epidemic Models

In an application of the paralogistic model of VANDERPLANK (1963) to aphid-borne virus epidemics, vector abundance, propensity and behaviour must be summarized in a constant vector intensity i' (Eq. 3.6). In the model class proposed by MADDEN *et al.* (1990) vector abundance is a time-varying input, whereas vector propensity and behaviour are summarized in a constant factor. The model of SIGVALD (1986) treats vector abundance as a time-varying input and vector propensity as a species specific parameter. Vector abundance is estimated by yellow water trap catches. The vector propensity parameters were corrected for species specific attractivity of the yellow water traps (Dr. R. Sigvald, Swedish Univ. of Agricultural Sciences, S-750 07 Uppsala, Sweden, pers. comm.). Apart from vector abundance and propensity, the model of RUESINK & IRWIN (1986) includes one behaviour parameter, namely the mean number of interplant movements, performed by an aphid before leaving the field. The model published by BERGER & FERRIS (1989) includes parameters of the spatial virus dynamics, such as kind and

distance of vector movement, but does not allow the investigation of other behaviour characteristics, like probing frequency and duration, walking and flight frequency. The model of VAN DER WERF *et al.* (1989) includes probing and dispersal behaviour of the vectors. However, it is not applicable to the potato-PVY system, because the virus is semi-persistently transmitted, and the behaviour characteristics involved are thus different from those relevant for nonpersistent transmission. Moreover, the model takes only apterae and colonizing species into consideration.

Finally we may conclude that none of the currently available epidemic models calculates the rate of virus spread as a function of all behaviour parameters believed to be relevant for nonpersistent virus epidemics. Hence, these models cannot be used to study the effect of the vector behaviour on PVY epidemics.

4. THE ROLE OF VECTOR DISPERSAL IN SPATIO-TEMPORAL VIRUS SPREAD - A SIMULATION STUDY

The spatial pattern of virus diseased plants is often nonrandom (SCOTT, 1985; MADDEN *et al.*, 1987) and gradients of disease intensity around the infection source have been observed for many virus diseases (THRESH, 1976), among them for PVY and PLRV (GREGORY & READ, 1949). The effect of the mechanisms leading to this nonrandom pattern is investigated in this chapter.

In apparent contradiction to the field observations, most epidemic models of plant diseases implicitly assume a random spatial pattern (CAMPBELL & MADDEN, 1990), e.g. the logistic model of VANDERPLANK (1963), the PVY-epidemic model in potatoes (SIGVALD, 1986) or the citrus tristeza virus epidemic model of MARCUS *et al.* (1989). Since experimental evidence indicates that the spatial pattern of virus infected plants is often nonrandom, this assumption is questionable for these diseases. To describe temporal epidemic progress appropriately, we ought to know, on what conditions the assumption of randomness is acceptable.

Nonrandom patterns of diseases can be caused by the processes leading to virus spread, i.e. vector dispersal, or by inhomogeneities in the field, e.g. unequal susceptibility for virus infections or different attractivity of plants. It is difficult to determine whether vector dispersal or inhomogeneity of the field caused the spatial pattern in field experiments (MADDEN, 1989). Since this study focuses on the role of vector dispersal in virus spread, inhomogeneity of the field was excluded by using a simulation model. This model allows further to change dispersal parameters easily.

Different models have been used to describe nonrandom disease spread. Some modellers attempted to account for the nonrandom pattern by correcting the equation for the rate of temporal disease spread, using a known distribution, e.g. the negative binomial (WAGGONER & RICH, 1981; RUESINK & IRWIN, 1986). The limitation of this approach lies in the temporal variation of the spatial distribution parameters (MADDEN, 1989). Physical dispersal models (CAMPBELL & MADDEN, 1990) describe fungal spore dispersal by physical laws. The mathematical treatment of these models is difficult. Empirical dispersal models (CAMPBELL & MADDEN, 1990) assume a statistical distribution of inoculations around a source. This approach was chosen here, since vector dispersal can hardly be described by physical laws only.

A new model, called 'Spatial Epidemic', was constructed using a stochastic two-dimensional cellular automaton (WOLFRAM, 1986). Studies of the model 'Spatial Epidemic' served three purposes: to study the relationship between vector dispersal distance and epidemics, using three statistical distributions of dispersal distances with different parameter values, to determine on what conditions an assumption of randomness could be accepted and as a preliminary study of the EPOVIR-model (Chapter 6).

4.1. Model Description

The model is based on the following assumptions:

- a cell represents an individual plant, which can at a time be in one state of a set of discrete states
- the field is a torus
- the inoculations occur in a certain distance and direction from an infection source, the distances and directions follow a statistical distribution
- the sources outside the field do not influence the epidemic in the field
- a constant latent period exists
- the infectious period is infinite.

Cell States

The model 'Spatial Epidemic' represents plants as cells of a cellular automaton. The cells are numbered from 1 to n_r in x-direction (row index), and from 1 to n_p in y-direction (plant index within row). The state of a cell $[X,Y]$ in relation to the virus disease is the *disease state* D_{XY} .

$$D_{XY} \in \{\delta_h, \delta_l, \delta_p, \delta_s, \delta_r\}$$

The disease states are summarized in Tab. 4.1. Cells in the states δ_p and δ_s act as infection sources.

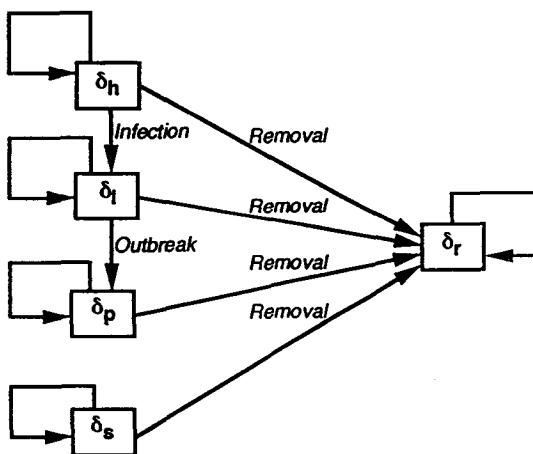


Fig. 4.1: Kinematic graph (state transition diagram) showing discrete states and transitions between them.

Three processes can change the disease state of a cell (Fig. 4.1): *infection*, *outbreak* and *removal* (roguing or death)¹. The three processes are irreversible, δ_r is an absorbing state (Fig. 4.1). Infection and removal are stochastic processes.

¹To make the interpretation of the results easier, removal was not simulated in the presented simulation experiments.

Tab. 4.1: Disease states of individual cells (plants).

Symbol	Meaning
δ_h	healthy
δ_l	latent infection (infected, but not an infection source)
δ_p	primary infection source
δ_s	secondary infection source
δ_r	plant lacking, dead or removed

Initial State

The initial state of a cell D_{XY} at simulation start time ($t_0 = 0$) is:

$$D_{XY}(t_0) \in \{\delta_h, \delta_s, \delta_r\}$$

p_i is the initial proportion of infection sources, the so called *initial disease incidence*. To initialize the model $Int(p_i n_r n_p + 0.5)$ ¹ plants were chosen randomly in the field to be in state δ_s . All other plants were in state δ_h .[†]

Torus

The simulated field was treated as a torus. A torus simulates an infinite field by a finite number of cells. The border effect is eliminated by connecting the left with the right and the upper with the lower border of a rectangular field. Each cell has thus the same number of neighbouring cells. E.g. the cell [1,1] and [1,n_r], or [1,5] and [n_r,5] are neighbouring cells.

The calculated cell coordinates² of an inoculation lie outside the simulated area if:

$$i_X < 1 \vee i_X > n_r \vee i_Y < 1 \vee i_Y > n_p \quad (4.1)^3$$

In case (4.1) the cell coordinates of inoculated cells $[i_X, i_Y]$ are transformed by:

$$\begin{aligned} i_X' &= \begin{cases} i_X - Int((i_X-1)/n_r) \cdot n_r & \text{if } i_X > n_r \\ i_X - Int((i_X-1)/n_r) \cdot n_r & \text{if } i_X < 1 \end{cases} \\ i_Y' &= \begin{cases} i_Y - Int((i_Y-1)/n_p) \cdot n_p & \text{if } i_Y > n_p \\ i_Y - Int((i_Y-1)/n_p) \cdot n_p & \text{if } i_Y < 1 \end{cases} \end{aligned} \quad (4.2)$$

Inoculations and Infections

i is the number of inoculations originating from an infection source plant (vector intensity). To determine the number of plants to be inoculated, the real number ($i \in \mathbb{R}$)⁴ is transformed into an integer number ($i^* \in \mathbb{I}$)⁵ for any source plant. Since i is constant

¹ $Int(x)$ is the integer part of x .

[†] The initial state δ_r of a plant would mean a plant that has not emerged. This initial state was not chosen for the simulations in this chapter.

² The metric coordinates of a point in the field will be denoted by lower case index letters x and y . E.g. the vector $[i_X, i_Y]$ defines a point in the field, with the origin [0,0] at the lower left corner of the field. Cell coordinates (i.e. the row index (X) and the plant index within the row (Y)) will be denoted by upper case letters. Thus the cell $[i_X, i_Y]$ is the Y -th plant in the X -th row. The cell in the lower left corner of the field is the cell [1,1].

³ The symbol ' \vee ' is used for 'OR'.

⁴ \mathbb{R} is the set of real numbers.

⁵ \mathbb{I} is the set of integer numbers.

during a simulation run, simple rounding of i ($i^* = \text{Int}(i+0.5)$) would cause a deviation between the calculated number of inoculations ($i \times$ number of infection sources) and the actual one ($i^* \times$ number of infection sources), especially for low values. Hence i^* was determined by stochastic sampling. For each infection source plant ($D_{XY}(k) = \delta_s$ or δ_p) i^* plants are sampled to be inoculated.

$$\begin{aligned} i^* &= \begin{cases} \text{Int}(i)+1 & \text{if } u < i-\text{Int}(i) \\ \text{Int}(i) & \text{otherwise} \end{cases} \\ u &= U[0,1) \end{aligned} \quad (4.3)$$

If a healthy plant ($D_{XY}(k) = \delta_h$) is inoculated, it becomes a latently infected plant ($D_{XY}(k+1) = \delta_l$). An inoculation of a plant in another state than δ_h does not change its state.

Statistical Distribution of the Inoculations

The direction i_ϕ of the inoculation in relation to the infection source is sampled from a uniform circular distribution:

$$i_\phi = U[0, 2\pi) \quad (4.4)$$

The distribution of the distances between an infection source and an inoculation can follow a normal, exponential or Weibull distribution (Tab. 4.2).

Tab. 4.2: Probability density function, distribution function and random number generating function of the three statistical distributions of the dispersal distances (after HASTINGS & PEACOCK, 1975).

	Normal	Exponential	Weibull
Probability density function $f(x) = \frac{d(F(x))}{dx}$	$\frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$	$\frac{e^{-x/\mu}}{\mu}$	$\alpha\beta^{-\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha}$
Distribution function $F(x) = \int f(u) du$	not solvable	$1 - e^{-x/\mu}$	$1 - e^{-(x/\beta)^\alpha}$
Random number generating function	BOX & MULLER (1958)	$-\mu \ln(U[0,1])$	$\beta (-\ln(U[0,1]))^{1/\alpha}$
Parameters	$\mu = \text{mean} (= 0 \text{ in the model})$ $\sigma = \text{standard deviation}$	$\mu = \text{mean}$	$\alpha = \text{shape parameter}$ $\beta = \text{scale parameter}$

Exponentially and Weibull distributed random numbers were generated using the inversion method, the normal deviates with the acceptance-rejection method of BOX & MULLER (1958).

The distance i_d is sampled from one of the three statistical distributions shown in Tab. 4.2. Given the metric coordinates of an infection source $[s_x, s_y]$, the metric coordinates of an inoculated cell are:

$$\begin{aligned} i_x &= s_x + i_d \cos(i_\phi) \\ i_y &= s_y + i_d \sin(i_\phi) \end{aligned} \quad (4.5)$$

where the metric coordinates of the source $\{s_X, s_Y\}$ are calculated as:

$$s_X = (s_X - 0.5) d_r \quad (4.6)$$

$$s_Y = (s_Y - 0.5) d_p$$

The cell coordinates $\{i_X, i_Y\}$ of the inoculated cell are:

$$i_X = \text{Int}\left(\frac{i_X}{d_r}\right) + 1 \quad (4.7)$$

$$i_Y = \text{Int}\left(\frac{i_Y}{d_p}\right) + 1$$

Disease State Transitions

The disease state transitions are given by Eq. 4.8 (see also Fig. 4.1):

$$D_{XY}(k+1) = \begin{cases} \delta_l & \text{if } D_{XY}(k) = \delta_h \wedge \text{inoculation occurs at } k \quad (\text{infection}) \\ \delta_p & \text{if } D_{XY}(k) = \delta_l \wedge k \geq k_{iXY} + m_l \quad (\text{outbreak}) \\ \delta_s & \text{if plant is removed at } k \quad (\text{removal}) \\ D_{XY}(k) & \text{otherwise} \quad (\text{no state change}) \end{cases} \quad (4.8)^1$$

where k_{iXY} is the infection time of the cell $[X, Y]$ and m_l the latent period.

Standard Run

Tab. 4.3: Parameters in the standard run of the model 'Spatial Epidemic'.

Parameter	Meaning	Unit	Value
i	number of inoculations per source plant	# plant ⁻¹ week ⁻¹	1
p_i	initial disease incidence	%/100	0.01
n_r	number of rows	#	40
n_p	number of plants in row	#	80
m_l	latent period	week	3
d_r	distance between rows	m	0.75
d_p	distance between plants in a row	m	0.375

The standard run parameter set (Tab. 4.3) defines a model, which was simulated from $k=0$ to $k=9$, with a step of one week. Two output variables were used for the statistical analyses: the disease incidence D (proportion of cells in states δ_l , δ_p and δ_s) and the evenness E of the distance class distribution (see Chapter 2.4.2, plants in states δ_l and δ_p were used instead of plants with infected tubers τ_l). Simulation results were monitored at three times t_1 to t_3 (Eq. 2.1), i.e. at $k = 3, 6$ and 9 . Each simulation experiment consisted of 50 simulation runs.

¹The symbol ' \wedge ' is used for 'AND'.

4.2. Vector Dispersal and Virus Epidemics

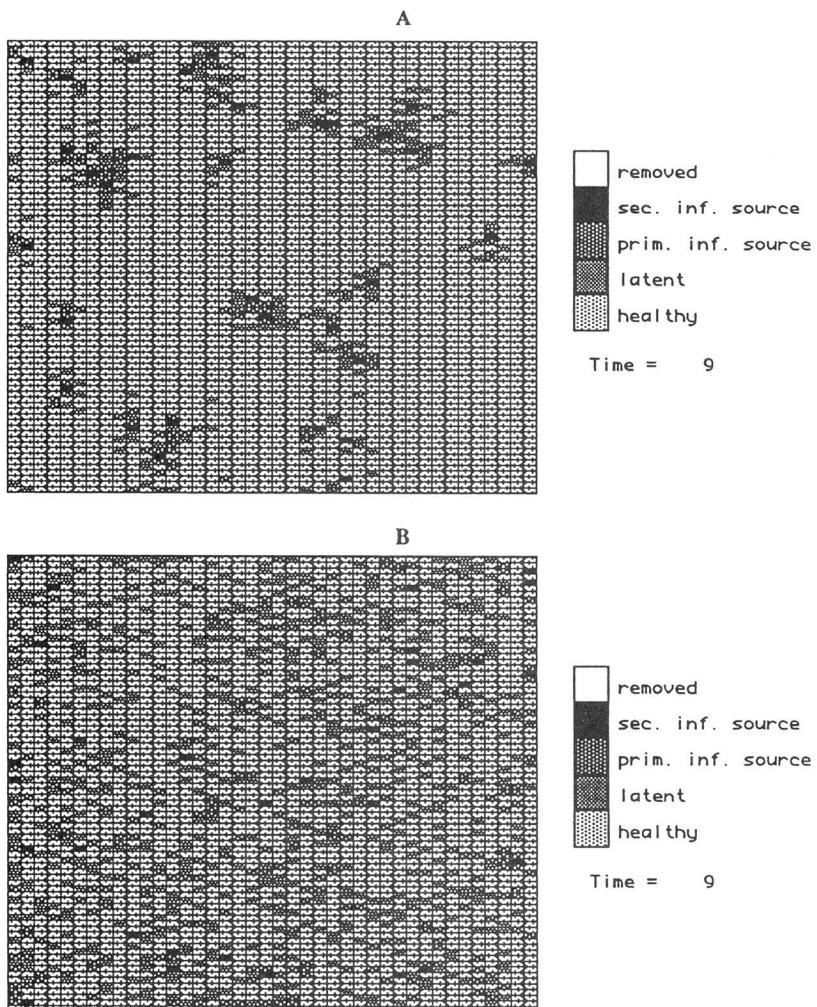


Fig. 4.2: Examples of a clumped (A) and nearly random pattern (B). The patterns were actually produced by an exponential distribution of the dispersal distances, where the mean μ was 0.5 m in A and 20 m in B.

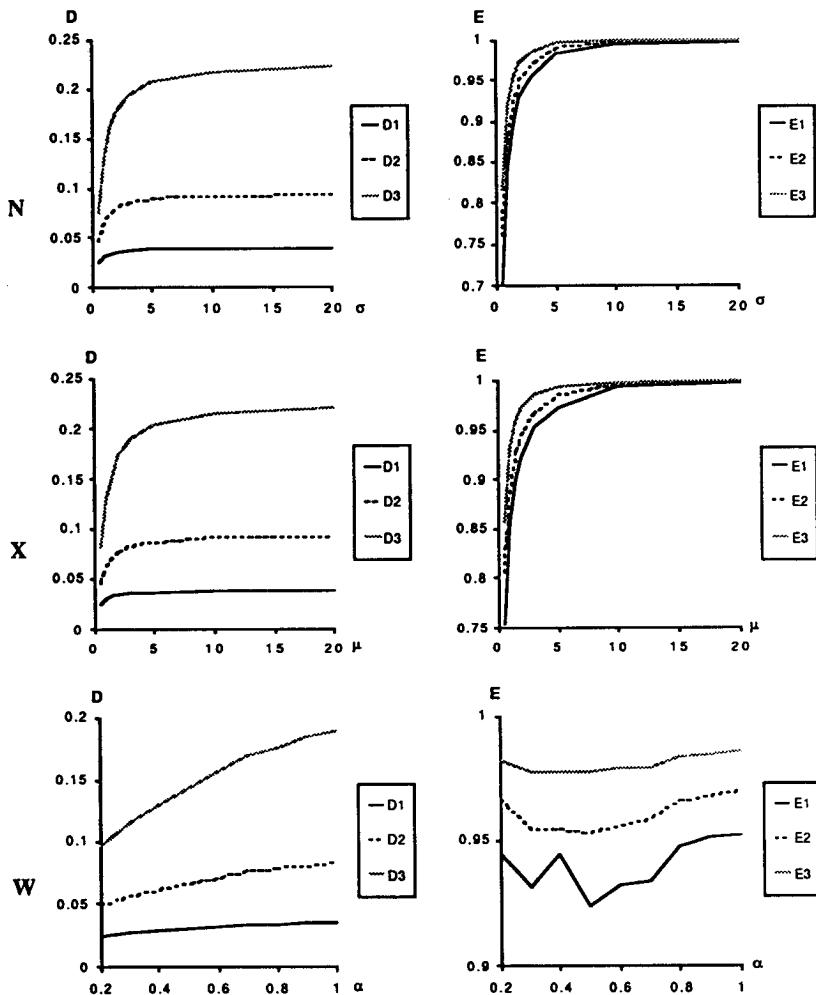


Fig. 4.3: Relationship between the parameters of the dispersal distance distributions and the disease incidence (D) resp. evenness of the distance class distribution of diseased plants (E) for three statistical distributions: N: normal distribution ($\mu = 0$, σ = standard deviation [m]), X: exponential distribution (μ = mean of exponential distribution [m]) and W: Weibull distribution (α = shape parameter of Weibull distribution, $\beta = 3$). D1 to D3 are the disease incidence, E1 to E3 the evenness values for the three monitoring times t_1 to t_3 . Note that the scales of D and E are different for the three distributions.

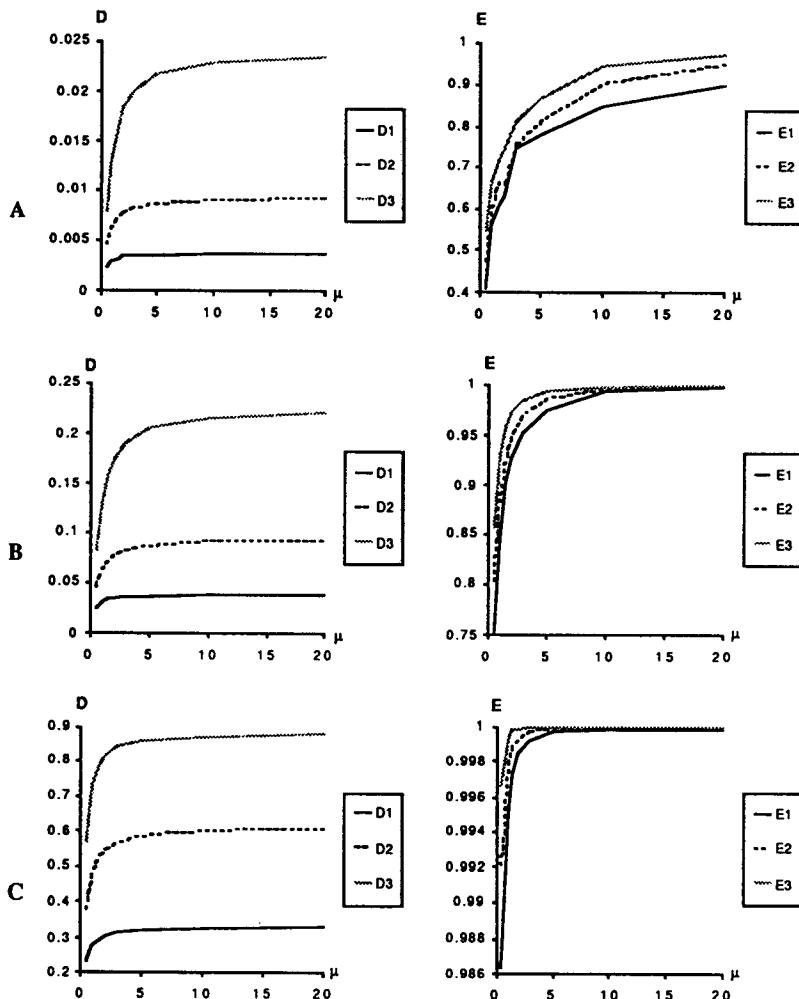


Fig. 4.4: Relationship between the mean μ of the exponential distribution of dispersal distances and the disease incidence (D) resp. the evenness (E) of the distance class distribution of diseased plants for three values of the initial disease incidence p_i : A: $p_i = 0.001$, B: $p_i = 0.01$ and C: $p_i = 0.1$. D1 to D3 are the disease incidence, E1 to E3 the evenness values for the three monitoring times t_1 to t_3 . Note that the scales of D and E are different for the three initial disease incidences.

Fig. 4.2 shows that the short dispersal distances (A) produced a high degree of clumping around the infection sources, leading to less plants becoming actually infected, than for longer dispersal distances (B). The disease incidence was positively correlated

with the mean dispersal distance, i.e. σ or μ (Fig. 4.3)¹. In a torus the disease incidence should be maximal for infinite distances, which would correspond to completely random spread. With increasing distance, the disease incidence should asymptotically approach this maximal value. The evenness of the distance class distribution was low for short dispersal distances and approached 1 for longer distances, i.e. the spatial pattern became nearly random. The normal distribution produced a slightly more aggregated pattern than the exponential distribution. Increasing the value of α of the Weibull distribution increased the disease incidence markedly and the evenness of the distance class distribution slightly.

The initial disease incidence considerably affected the spatial disease pattern (Fig. 4.4). A low initial disease incidence tended to yield more marked clumping, indicated by lower E -values. Moreover, the lower the initial disease incidence, the longer dispersal distances were needed to yield E -values near 1. The ratio of the disease incidences for the shortest and longest dispersal distance (0.5 and 20 m) was smaller for $p_i = 0.1$ than for the other two values. This difference disappeared, if the multiple infection transformation was applied to the disease incidences (GREGORY, 1948), showing that the smaller difference was due to the fact that the disease incidence approached 1 in experiment C.

4.3. Discussion

The simulation experiments showed that the mean distance of vector dispersal can considerably affect the spatial and temporal epidemics. Different processes will contribute to aphid-borne virus dissemination in the field: wingless aphids will transmit the disease by walking, winged aphids by walking and flight and by migration between fields. Wingless aphids will hardly walk over distances longer than 1 m (THYGESEN, 1968; FERRAR, 1969) and the retention time for nonpersistently transmitted virus diseases is short enough to prevent infections by apterae in a longer distance. Their contribution to virus epidemics is limited, because most plants in the proximity of the source, which can be reached by wingless aphids, become infected. The mean distance of trivial flights of alatae was estimated to be approximately 13 m (Chapter 5.4). The same number of movements by alatae would lead to about three times more infections at harvest time compared with movements by apterae. The inoculations caused by alatae immigrating from another field will probably be distributed randomly over the field. New foci can be created, from where further virus spread can occur.

The evenness values seem to allow discrimination between marked clumping, caused by short dispersal distances and random patterns, caused by long distances. They can thus be used as a measure of clumping. Low values indicate marked clumping.

E increased with time (Fig. 4.3 and 4.4), probably due to two causes: a) the distance class distribution tends towards uniformity with increasing disease incidence levels (Fig. 2.6) and becomes uniform as soon as all plants are infected. This 'saturation of the

¹To limit the number of simulations, the parameter β of the Weibull distribution was not varied. It was assumed that the response of the output variables to changes in β will be similar to the response to changes in μ of the exponential distribution, since if $\alpha = 1$, the Weibull distribution is equal to the exponential distribution with $\mu = \beta$.

'focus' has been observed in real systems (THRESH, 1978) and could also be reproduced with the EPIMUL-model (ZADOKS & KAMPMEUR, 1977). An example shall illustrate this effect: provided all plants in the first distance class are infected and the disease incidence in the whole field is 0.05, the ratio of observed and expected number of infected plants in the first class is $f_I = 20$ ($= 1/0.05$, Eq. 2.4). If the disease incidence in the field achieves 0.5, f_I equals 2 ($= 1/0.5$). The closer the f_c -values are to 1, the higher is the evenness of the pattern. b) As soon as primary sources appear in the field, further virus spread from these sources makes the pattern more uniform (primary sources were not used as origins for the distance class distribution analysis).

The differences in the evenness values for different initial disease incidences (Fig. 4.4) might be surprising, since the evenness values were standardized (corrected for the mean disease incidence). The phenomenon can be explained as follows: if the initial disease incidence is low, the infection sources are in a longer distance from each other. Therefore the foci that develop will hardly overlap. If the initial disease incidence is high (e.g. 10% in experiment C), the foci around the initial sources will soon overlap, covering the whole surface towards the end of the simulation. The pattern becomes nearly random. High initial disease incidences are likely to lead to the infection of nearly all plants. In this case the effect of clumping will disappear.

The assumption of randomness in space in most epidemic models, is justified only if the mean dispersal distance exceeds 5 to 10 m and the initial disease incidence is not too low. In the case of PVY in potatoes, this assumption seems not to be justified, since walking aphids (apterae and alatae) can contribute to PVY spread. Furthermore the initial disease incidence is usually low (p_I is typically lower than 0.01) and clumping can thus play an important role. Therefore the usage of a cellular automaton is indicated for the investigation of the epidemic mechanisms of PVY in seed potatoes. To estimate the corresponding dispersal distance parameters, accurate estimates are needed, if the distances are short, but rough estimates seem to be sufficient, if the distances exceed 5 to 10 m.

5. EXPERIMENTAL AND MARKOV MODEL STUDIES OF WINGED APHID BEHAVIOUR

In this chapter, parameters describing dispersal and probing behaviour of aphids in potato fields were estimated and the properties of the behaviour models derived were studied. The behaviour study was concentrated on alatae populations, which are likely to come from their (primary or secondary) hibernation hosts into a potato field during the spring migration.

Behaviour sequences of winged aphids of known age were recorded under controlled and field conditions. Two species were chosen as model species for the groups of the colonizing (*M. persicae*) and noncolonizing (*Aphis fabae* SCOP.), because they are abundant at many sites and therefore available in sufficient quantities for the study, they are considered as important PVY vectors (DERRON & GOY, 1990) and well known from other studies in the literature. Additional observations with two other species (*Aphis nasturtii* KALTENBACH - colonizing and *Brachycaudus helichrysi* KALTENBACH - noncolonizing) were carried out to gain some information on whether the behaviour of *M. persicae* and *A. fabae* could be considered as representative for the respective groups.

To reduce the huge amount of information contained in the behaviour sequences and to allow easy comparisons between species and age classes the sequences were visually grouped into three behaviour types characterizing the aphid-host plant relationship.

Behaviour sequences have often been described satisfactorily by Markov chains (CANE, 1978; METZ *et al.*, 1983). Using the experimental results, parameters for a discrete Markov chain model were estimated. The discrete Markov model was used in a first step, since statistical methods are much better developed for this model class.

The properties of the derived Markov models were studied in the second part. The discrete Markov models were transformed into continuous Markov models. Models of this class are better suited for simulation studies, since the time step can be easily changed. Behaviour characteristics were calculated for the steady state of the models and the sensitivity of the behaviour sequences to changes in transition parameters was assessed.

Post-migratory flight behaviour of aphids in a potato field was studied in the third part, to estimate parameters describing flight direction and distance and their dependence on environmental conditions. As most flights could not be observed until landing, minimal and probable flight distances were assessed with the aid of a simple simulation model, based on observations. The dependence of flight direction on wind direction and sun position was investigated. To determine prevailing wind directions, the frequency distribution of the wind directions for the Agricultural Research Station in Changins was analyzed for the whole growing season and for those periods, when aphid flight was supposed to be possible.

5.1. Material and Methods

5.1.1. OBSERVATION OF BEHAVIOUR SEQUENCES

Aphid Material

Alatae aphids used in the experiments were collected as nymphs with the aid of a fine brush either on leaves of primary hosts or from a rearing on secondary hosts in a climatic chamber. *M. persicae* originated from colonies on *Sinapis arvensis* L. collected in March 1989 and were reared on chinese cabbage (*Brassica chinensis* L.). *A. fabae* were collected on the primary hosts *Eryngium europaea* L. and *Viburnum opulus* L. or in a rearing on *Vicia faba* L., which was started with colonies taken from the primary hosts *E. europaea* and *V. opulus*. *A. nasturtii* was taken from the primary host *Rhamnus catharticus* L. and *B. helichrysi* from the primary host marian plum (*Prunus cerasifera x munsoniana x angustifolia*).

The nymphs were kept in plastic petri dishes (diameter of 8.5 cm) on leaves of the plant species, where the individuals had been sampled, underlaid with wet filter paper. The dishes were kept at daylight at a mean temperature of 20 °C and a mean rel. humidity of 60 %. Hatched alatae were collected and transferred every evening to another petri dish, containing leaves of the plant species, where they had been sampled. The alatae stayed overnight in these dishes, which ensured that they had passed a teneral period of at least 12 h, which seems necessary for alatae to be able to fly (ROBERT, 1987). The following day the alatae were transferred to petri dishes without leaves, where they stayed for 2 h. According to KLINGAUF (1976), a fasting period has similar effects on aphid behaviour as a distance flight. The alatae were thus in a state corresponding to the state after a distance flight.

Observation of the Sequences of Behaviour under Controlled Conditions

Potato plants of the Bintje variety were grown in plastic pots (diameter 14 cm) in a green house and used for aphid behaviour observations at an age of 10 to 15 days after emergence. The aphids were transferred one by one to these plants enclosed in a cylindrical cage of plexiglass (diameter 30 cm, height 50 cm). The observations were performed at a temperature of 22°C and a rel. humidity of 75% in a climatic chamber illuminated by 11 40W-neon tubes. An aphid was observed during 15 min with the aid of a glass and a dentist's mirror. Any change of the aphid's behaviour state was recorded together with the corresponding time (time resolution 1 s) using a programmable event-recorder (PSION Organiser II, Model XP). This corresponds to a complete record according to SLATER (1978). The data were later transferred to an Apple™ Macintosh™ computer for further analysis. Six behaviour states were distinguished in the observations (according to LEHMANN *et al.*, 1975 modified):

- Flight
- Walking
- Resting (aphid remains motionless, usually with rostrum directed backwards)
- Wing beating (movement of the wings, often followed by flight)

- Tapping (aphid touches the plant surface with the tip of the rostrum, *the antennae are directed upwards*)
- Penetration (according to TARN & ADAMS, 1982, stylets penetrate into plant tissue, includes probing and feeding. The rostrum is perpendicular to the plant surface, *the antennae are directed backwards and motionless*)

For the analysis tapping and wing beating was counted as resting, because aphids in these three behaviour states do not contribute to virus dissemination. Tapping and wing beating often occurred in sequence with resting.

After the first observation, performed immediately after the hunger period, the aphids were kept singly in petri dishes on potato leaves with wet filter paper and subjected to further observations (whenever possible three times a week) until death.

The ethograms of the behaviour sequences were plotted in colour using a Dialog Machine program (FISCHLIN, 1986) to compare the sequences visually.

Observation of the Sequences of Behaviour in the Field

The same observation method as under controlled conditions was applied in the field experiments, except that no cage was used and the individuals were observed only once (most individuals were lost due to flight). The experiments were performed between May and July 1989 in a potato field, variety Bintje (planting and emergence dates given in Tab. 2.3, 1989). The alatae were observed immediately after the hunger period, or stored in groups on potato leaves with wet filter paper in the laboratory until observation. Each aphid was observed until it was lost during flight, or penetrated longer than 15 min. In the latter case, the aphid's location was marked with a pin and controlled in 15 min intervals as long as possible. If the aphid was still penetrating and had not changed its location, it was assumed that it had been in state 'long penetration' during the whole interval. Otherwise, the end of the penetration was estimated with the midpoint between the last two controls. Contrary to the behaviour sequences recorded under controlled conditions, the behaviour sequences recorded in the field were of varying duration.

Splitting Penetration into Probing and 'Long Penetration'

Simulation studies of the Markov models showed that it was not possible to satisfactorily describe the behaviour sequence with only one state 'penetration' with an exponential distribution of the durations. An exponential distribution of durations is inherent to Markov chains (METZ *et al.*, 1983). The frequency distribution of the penetration durations indicated that a satisfactory description of the sequences might be achieved by splitting penetration acts into short penetrations (probing) and 'long penetrations' (Fig. 5.1). A statistical criterion was chosen, based on the duration of the acts, because visual observation does not allow to distinguish these two behaviour states. A nonlinear model was fitted to the log frequency data (Fig. 5.1, SIBLY *et al.*, 1990). The Quasi-Newton method was used to minimize the least squares of the deviations. The threshold was determined with the criterion minimizing the number of acts assigned to the wrong state, the so-called misassigned acts (SIBLY *et al.*, 1990).

Estimation of the Transition Matrix and Statistical Tests

The transition probabilities for a first order Markov chain (discrete time) were estimated according to ANDERSON & GOODMAN (1957) by the proportion of individuals in state I passing to state J in the following step¹:

$$\hat{p}_{IJ} = \frac{\sum_{k=a}^{b-1} n_{IJ}(k)}{\sum_{k=a}^{b-1} n_I(k)} \quad (5.1)$$

where \hat{p}_{IJ} is the probability to pass from state I to state J in the time interval $[a, b]$, k is the time, $n_{IJ}(k)$ is the number of transitions from state I (at time k) followed by J (at time $k+1$), $n_I(k)$ is the number of occurrences of state I at k .

To test the Markov chains for stationarity, i.e. the constancy of the transition probabilities in time, the number of state transitions was calculated for the time intervals: 0-300 s (I_1), 300-600 s (I_2) and 600-900 s (I_3) after the begin of the record. A log-linear model was fitted to the observed number of transitions, arranged in a three-dimensional matrix (according to BISHOP *et al.*, 1975):

$$\ln(m_{IJT}) = \mu + \lambda_I + \lambda_J + \lambda_T + \lambda_{IJ} + \lambda_{JT} \quad (5.2)$$

where m_{IJT} is the expected number of transitions from state I to state J in time interval T . μ is the natural logarithm of the overall mean and λ_I , λ_J and λ_T are the 'main effects'. Two second order interaction terms (λ_{IJ} and λ_{JT}) are included. The model omits the second order interaction term λ_{IT} for the different time intervals, thus testing whether the model still fits the data with $\lambda_{IT} = 0$. If the expected numbers of transitions m_{IJT} (Eq. 5.2) do not significantly deviate from the observed cell counts, the null hypothesis that the T intervals have the same transition probabilities will be retained and the Markov chain is assumed to be stationary in the tested time intervals. If the null hypothesis is rejected, the Markov chain is considered as not stationary, i.e. different transition matrices must be used for the different time intervals. The following transitions were treated as structural zeros (expected cell count = 0) for the test: flight→'long penetration', walking→'long penetration', resting→'long penetration' and 'long penetration'→probing, because these transitions could not occur by definition of the state 'long penetration'. The log-linear model was fitted after the addition of 0.5 to all cell counts (otherwise the number of degrees of freedom would have been too small for the test, due to the zero values). The likelihood ratio values (G^2) were used to test for significance.

¹The indices of array elements will be used as follows: small caps letters I and J will denote the rows resp. columns in the markov matrices of behaviour sequences. The index variable C stands for the colonization group, G for the vector group and V for the virus strain. Thus $I, J \in \{f, w, s, p, l\}$, $C \in \{col, ncol\}$, $G \in \{1, 2a, 2b, 3\}$ and $V \in \{PVY0, PVYn\}$ (see also Appendix II).

5.1.2. ANALYSIS OF MARKOV CHAINS

Continuous Markov Chain Model

The continuous Markov chain model of the behaviour sequence is a linear differential equation system of fifth order for colonizing, and analogous for noncolonizing species. The differential equation system for colonizing species is independent from the one for noncolonizing species. The temporal dynamics of the proportion p_{JC} of aphids in behaviour state J is given by:

$$\frac{d p_{JC}}{dt} = \sum_{I \neq J}^l p_{IC} q_{IJC} \quad J \in \{f, w, s, p, l\} \quad (5.3)$$

p_{JC} and p_{IC} are the proportions of aphids in behaviour state J resp. I , q_{IJC} is the transition rate from state I to state J . The frequency f_{JC} of behaviour state J , i.e. the number of occurrences per time unit, is:

$$f_{JC} = \sum_{\substack{I=1 \\ I \neq J}}^l p_{IC} q_{IJC} = \frac{d p_{JC}}{dt} \cdot p_{JC} q_{JJC} \quad (5.4)$$

Probing frequency f_{pC} was calculated differently, taking into account only probes not followed by 'long penetration':

$$f_{pC} = \sum_{\substack{I=1 \\ I \neq p}}^l (p_{IC} q_{IpC}) - p_{pC} q_{pIC} \quad (5.5)$$

This value can be considered as the probing frequency relevant for nonpersistent virus transmission, since the transmission probability declines with increasing probing duration (VÖLK, 1959; ZETTLER, 1967; KATIS & GIBSON, 1985).

The steady state of the system (proportions of states to which the system converges) were calculated by matrix operations (SNELL, 1989) with the spreadsheet program Microsoft Excel™ version 2.2.

Transformation of Discrete to Continuous Markov Matrices

The transition rates q_{IJC} of a continuous Markov chain (forming the Q_C -matrix) were derived from the transition probabilities p_{IJC} of the discrete Markov chain (P_C -matrix) according to JEFFERS (1988). The off-diagonal elements q_{IJC} (Eq. 5.6) are given by the quotient of the transition probability p_{IJC} and the sum of all off-diagonal elements in the I -th row of the P_C -matrix (s_{IC} , Eq. 5.7), multiplied by M_{IC} (Eq. 5.8). Δt is the time step of the discrete Markov chain.

$$q_{IJC} = \frac{p_{IJC}}{s_{IC}} M_{IC} \quad (5.6)$$

$$s_{IC} = \sum_{\substack{J=1 \\ J \neq I}}^l p_{IJC} \quad (5.7)$$

$$M_{IC} = \frac{-\ln(p_{IC})}{\Delta t} \quad (5.8)$$

The diagonal elements q_{IIC} are given by the negative sum of the off-diagonal elements of row I by Eq. 5.9.

$$q_{IIC} = - \sum_{\substack{J=1 \\ J \neq I}}^l q_{IJC} \quad (5.9)$$

Changing Markov Transition Rates

To assess the sensitivity of the behaviour sequences to changes in the Markov transition rates, the parameters were modified in two ways:

- The elements of a matrix *row* were divided by a factor k_{dl} (mean residence time prolonged by factor k_{dl} , d stands for duration).
- The elements of a matrix *column* were multiplied by a factor k_{fj} (modifies the frequency of behaviour state J , f stands for frequency).

In the matrix of a continuous Markov chain the row sum must equal 0. To conserve this property, changing only one parameter of a row is not possible; at least two matrix elements must be changed simultaneously.

Mean Residence Times: The mean residence time r_{IC} in state I is given by the reciprocal of the transition rate q_{IIC} :

$$r_{IC} = \frac{1}{q_{IIC}} \quad (5.10)$$

To modify the mean residence time of behaviour state I , the elements of the I -th row were divided by a scaling factor k_{dl} :

$$q'_{IJC} = \frac{q_{IJC}}{k_{dl}} \quad J \in \{f, w, s, p, l\} \quad (5.11)$$

q'_{IJC} is the element of the modified Q_C -matrix. E.g. if $k_{dl} = 2$, the rate of leaving state I will be halved and the mean residence time r_{IC} of I will be doubled.

Frequencies: Multiplication of the elements of a matrix column by a scaling factor k_{fj} changes the frequency of state J , but this change is not proportional.

$$q'_{IJC} = k_{fj} q_{IJC} \quad I \in \{f, w, s, p, l\}, I \neq J \quad (5.12)$$

The diagonal elements q'_{IIC} of the rows with modified elements were calculated by Eq. 5.9. Changes of the scaling factor of probing frequency k_{fp} required a special treatment. These changes had virtually no effect on the probing frequency, due to the close link between probing and 'long penetration', which acts as an 'quasi absorbing' state. To compensate for this, the transition rate probing → long penetration q_{plC} was divided by k_{fp} .

$$q'_{plC} = \frac{q_{plC}}{k_{fp}} \quad (5.13)$$

The model, where q'_{plC} was calculated by Eq. 5.13 ('corrected model') was compared with the 'uncorrected model', where q_{plC} remained unchanged ($q'_{plC} = q_{plC}$). The diagonal element q'_{ppC} was again calculated by Eq. 5.9.

5.1.3. OBSERVATION OF FLIGHT BEHAVIOUR

Analysis of the Wind Directions

The mean hourly wind directions, measured at the height of 12 m (Chapter 2.7.5) during the months May to July of the years 1983 to 1989 were used for the following comparison: from the set of all hourly values those values were extracted that fulfilled the following conditions: daylight (at least 30 min of the hour interval should lie between the sunrise and sunset times) and mean wind speed (measured at 12 m) ≤ 2.5 m/s. These conditions were assumed to be suitable for aphid flight, because aphids rarely fly in the dark and under high wind speeds (JOHNSON, 1969; WALTERS & DIXON, 1984; BOUCHERY, 1987; ROBERT, 1987).

Aphid Material

The alatae originated from primary hosts or rearings on secondary hosts that were conditioned as described in Chapter 5.1.1. Other alatae were collected on the upper leaf side¹ of potato plants in the field. Only aphids that did not penetrate were used, which should ensure that these aphids were unsettled.

Flight Observations and Analysis

A single person carried out the observations in a field plot of 15×15 m, surrounded by potatoes on two sides. Each plant within the plot was labelled, to allow the calculation of the flight distances. Groups of five to ten alatae were placed on a detached potato leaf fixed with a wire in a water-filled petri dish, which was positioned on a post of 80 cm height (measured from the bottom of the furrow). The water prevented the aphids from walking off the leaf. The observations were performed during optimal flight conditions (no strong wind, no rain). If one of the aphids took off, the observer tried to follow it until landing. The flight direction (between the release and the landing site or the site, where the aphid was lost), the wind direction (measured simultaneously by a turnable fisherline trap, LABONNE & QUIOT, 1988) and the time were recorded. The directions were determined by a compass. In case of a landing, the number of the plant was noted for later calculation of the flight distance and direction. In case of landings outside the plot, the distance was estimated. In case the aphid was lost during flight, the distance (d_l , distance of loss) and the height (h_l , height of loss) of the location, where the aphid was lost, was estimated. The flight direction was compared with the wind direction and with the solar azimuth by circular-circular rank correlation (BATSCHLET, 1981), implemented with a Microsoft Excel™ macro.

Wind Profile

The following wind profile above the potato canopy was assumed (after ZADOKS & SCHEIN, 1979):

$$v_w(z) = a_v \ln\left(\frac{z-D_\theta}{z_o}\right) \quad (5.14)$$

¹Aphids settle usually on the lower leaf side.

where $v_w(z)$ is the wind speed at the height z above ground, a_v is a scaling factor (proportional to the wind speed at a certain time), D_o and z_o are parameters describing the effect of the crop canopy on the wind profile (Tab. 5.1). Eq. 5.14 was used in the simulation model and to determine the proportion of alatae that had left the boundary layer. The boundary layer (TAYLOR, 1960) is the air layer, where the wind speed is lower than the maximum flight speed, i.e. the air layer, where aphids can fully control their movement relative to the soil. The maximum flight speed was assumed as 0.5 m/s (ROBERT, 1987). a_v was calculated by Eq. 5.14, setting $v_w(z)$ to the measured wind speed and z to the anemometer height of 12 m. The wind speed was measured in intervals of 10 min. The recorded flight times were rounded to obtain the next measurement of wind speed.

Estimation of the Flight Distance Distribution with Simulation Models

Two simulation models were constructed to estimate a range of possible flight distances: A) the minimal and B) the probable flight distances of those aphids that were lost during flight. The movements in the three-dimensional space were reduced to movements in a vertical plane parallel to the wind direction. The way each aphid had travelled from the site of loss was simulated. The models are based on the following assumptions:

Model A: the aphids are actively flying downwards with the sinking speed v_s , but make no active horizontal movement, i.e. they are carried away with the wind.

Model B: the aphids are actively flying upwind with a flight speed slightly faster than the wind speed, whenever possible (KENNEDY & THOMAS, 1974), but cannot exceed a maximum flight speed. The vertical movement of an aphid is a stochastic process, which can be described by the uniformly distributed random variable $U[v_s, v_c]$ between the sinking speed v_s and the rate of climb v_c (Tab. 5.1).

The movement of the aphids was described by two differential equations:

Model A

$$\frac{d z}{dt} = -v_s \quad (5.15a)$$

$$\frac{d df}{dt} = v_w(z) \quad (5.16a)$$

where z is the height above ground, df is the flight distance travelled from the site of release ($df > 0$ means downwind, $df < 0$ means upwind). The wind speed $v_w(z)$ is given by Eq. 5.14.

Model B

$$\frac{d z}{dt} = U[v_s, v_c] \quad (5.15b)$$

$$\frac{d df}{dt} = v_w(z) + v_f \quad (5.16b)$$

where v_f is the flight speed of the aphid relative to the air masses ($v_f < 0$, since the aphid is flying upwind) and t is the time in seconds.

Tab. 5.1: Parameters of the wind profile and the flight simulation model.

Parameter	Meaning	Unit	Value	Source
D_0	Effect of crop on wind profile	m	0.3	ZADOKS & SCHEIN (1979), canopy height assumed as 0.5 m
z_0		m	0.05	
v_s	sinking speed	m/s	-0.3669	mean maximum speed of <i>A. fabae</i> flying actively downwards, THOMAS <i>et al.</i> (1977)
v_c	rate of climb	m/s	0.2525	mean rate of climb of <i>Rhopalosiphum padi</i> L. after take-off from potato plants, WIKTELius (1982)
Δv_f	difference between active flight speed and wind speed	m/s	0.1	KENNEDY & THOMAS (1974)
v_{fmax}	maximum flight speed	m/s	0.5	ROBERT (1987)

Since durations of trivial flights are short, the wind profile was assumed to be constant during this time ($a_v = \text{const.}$). The active flight speed v_f is:

$$v_f = -\text{MIN}(v_w(z) + \Delta v_f, v_{fmax}) \quad (5.17)$$

where MIN is the minimum function, defined as:

$$\text{MIN}(a, b) = \begin{cases} a & \text{if } a \leq b \\ b & \text{if } a > b \end{cases}$$

v_{fmax} is the maximum flight speed and Δv_f the difference between active flight speed and wind speed. The initial values for z and d_f were the coordinates of the point, where the aphid was lost:

$$z(0) = h_l \quad (5.18)$$

$$d_f(0) = \begin{cases} d_l & \text{for } |\text{flight direction-wind direction}| \geq \frac{\pi}{2} (\text{downwind}) \\ -d_l & \text{for } |\text{flight direction-wind direction}| < \frac{\pi}{2} (\text{upwind}) \end{cases} \quad (5.19)$$

Landing occurred if $z \leq 0.5$ m (crop height). The total flight distance was d_f at the moment of landing. The models were integrated with Heun's method using an integration step of 0.05.

The absolute values of the observed and simulated flight distances of colonizing and noncolonizing species were compared by a Mann-Whitney-U test. It was assumed that all aphids collected in the field were noncolonizing aphids¹. Most of the aphids caught during this period in fisherline traps in the field were *Brevicoryne brassicae* L., a noncolonizing species.

A Weibull function with two parameters (Tab. 4.2) was fitted to the frequency distribution of the probable flight distances after adding a constant of 0.00001 to all values. The observed and the fitted distributions were compared with the Kolmogorov-Smirnov goodness of fit test. These analyses were performed with the statistical package StatGraphics™, version 2.6, running on a Mac286™ 80286 Coprocessor Card installed in a Macintosh™ IIx computer.

¹It was not possible to identify aphids in the field reliably and fast enough, therefore it was not known to which species the aphids belonged.

5.2. Quantification of Behaviour Sequences

Longevity

Alatae of *M. persicae* lived longer on potato leaves than those of *A. fabae* (*M. persicae*: 10.7 ± 6.4 [mean and standard deviation in days after emergence of the alata], $N = 75$ aphids; *A. fabae*: 4.0 ± 1.3 , $N = 95$ aphids). To test, whether the potato leaves influenced the longevity of *A. fabae*, additional 44 *A. fabae* were conditioned in the way described above and kept in petri dishes on wet filter paper, but without potato leaves. Their longevity (3.8 ± 1.2) did not differ significantly from that of *A. fabae* kept on potato leaves in a t-test. Therefore the potato leaves did neither increase nor decrease the longevity of *A. fabae*.

Splitting Penetration into Probing and 'Long Penetration'

The log frequency plot according to SIBLY *et al.* (1990) showed that the penetration acts were probably not due to a single process (all points would lie on a line), but to at least two processes (Fig. 5.1). A threshold duration of 179.2 s was calculated. For further analysis a penetration act was assumed to be probing until 3 min and to switch to 'long penetration' afterwards. Note that due to this definition the state 'long penetration' can only be reached via the state probing and that probing cannot follow immediately after 'long penetration' ($q_{lpC} = 0$).

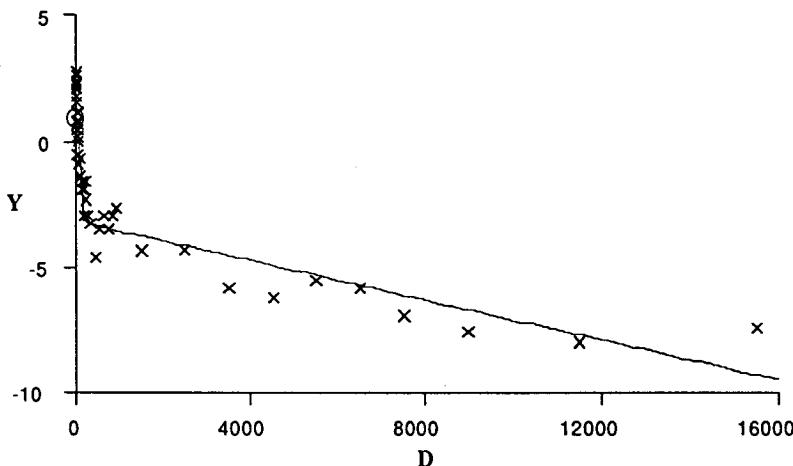


Fig. 5.1: Logarithmic frequency plot of the penetration act durations in the field. D = duration in seconds, Y = natural logarithm of the number of acts N per interval, weighted by the interval length ($Y = \ln \frac{N}{(D_u - D_l)}$, where D_u and D_l are the upper and lower limits of the respective interval (D_l, D_u)). The first point (circle) was excluded from the analysis. The line represents the fitted nonlinear function.

Classification of Behaviour Sequences

Tab. 5.2: Definition of the behaviour types under controlled and field conditions.

Behaviour type	Controlled conditions	Field	Name
I	>50% of time on cage or at the end of the observation on cage	lost by flight	<i>Host-rejection behaviour</i>
II	>50% of time on plant, but not in state 'long penetration' at the end of the observation	on plant at the end of the observation, but not in state 'long penetration'	<i>Searching behaviour</i>
III	in state 'long penetration' at the end of the observation	in state 'long penetration' at the end of the observation	<i>Settling behaviour</i>

The plotting of the sequences with the colour plot program revealed that similar behaviour sequences existed. The sequences were repeatedly rearranged and finally three behaviour types were derived (Fig. 5.2, Tab. 5.2), named according to the aphid-plant relationship. Aphids showing type I behaviour typically took off from the plant after a few probes. Type II behaviour was characterized by frequent probes interrupted by walking, type III by initiation of 'long penetration' after a few probes.

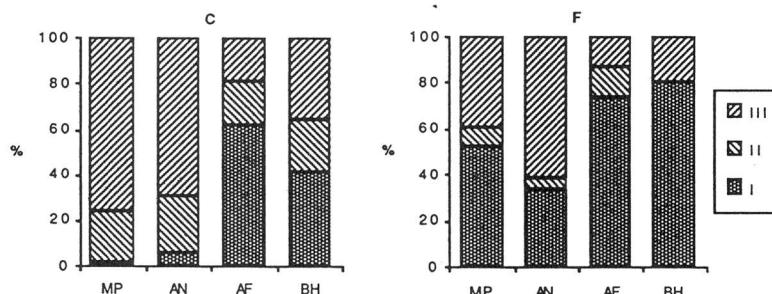


Fig. 5.3: Proportions of sequences observed under controlled conditions (C) and in the field (F) classified by the three behaviour types (I, II, and III) of the following species (number of observations given in brackets): MP = *M. persicae* (C: 202, F: 61), AN = *A. nasturtii* (C: 36, F: 14), AF = *A. fabae* (C: 72, F: 42), BH = *B. helichrysi* (C: 17, F: 8).

Colonizing species showed predominantly settling behaviour (Fig. 5.3, III), non-colonizing species mainly host-rejection behaviour (Fig. 5.3, I), i.e. they left the plant and spent most time on the cage. Only those field sequences were classified, where the final behaviour state had been recorded. The results revealed marked differences between the laboratory and field sequences. Particularly colonizing species had a lower flight frequency and less frequent host-rejection behaviour (I) under controlled conditions than in the field. These differences were similar but less marked for noncolonizing aphids.

The differences in behaviour within the groups of colonizing resp. noncolonizing species were smaller than the differences between the groups. This was confirmed also by visual comparison of the behaviour sequences.

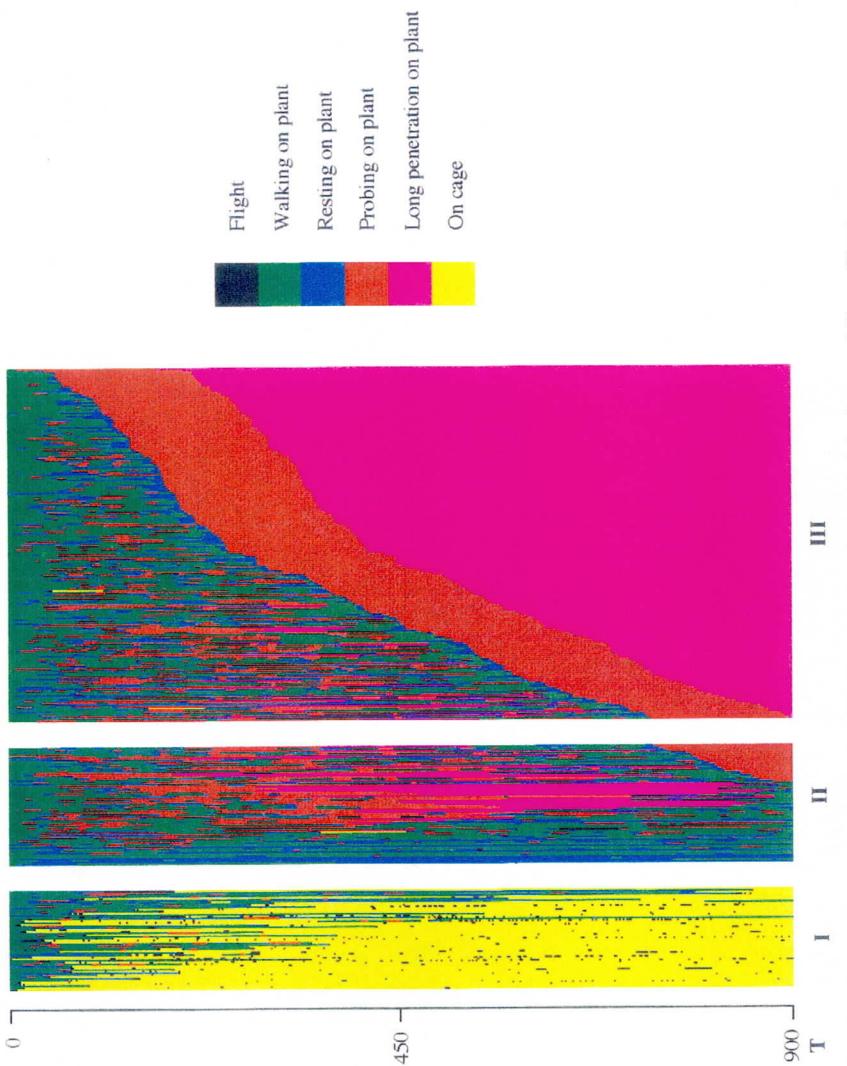


Fig. 5.2: Behaviour sequences observed under controlled conditions assigned to the behaviour types I, II resp. III and sorted within the types. Each line represents one behaviour sequence. T = time after beginning of observation in seconds.

Leer - Vide - Empty

The age classes were chosen differently for the two species due to the much shorter longevity of *A. fabae*. The proportion of host-rejection behaviour (Fig. 5.4, *I*) was highest in young individuals and decreased with age. The proportion of settling behaviour (*III*) increased, whereas the proportion of searching behaviour (*II*) decreased with age in *M. persicae*.

M. persicae performed flights only during the first two days in the climatic chamber and during the first three days in the field. Yet, most *A. fabae* remained able to fly until death. Even the oldest *A. fabae* (seven days) were able to fly.

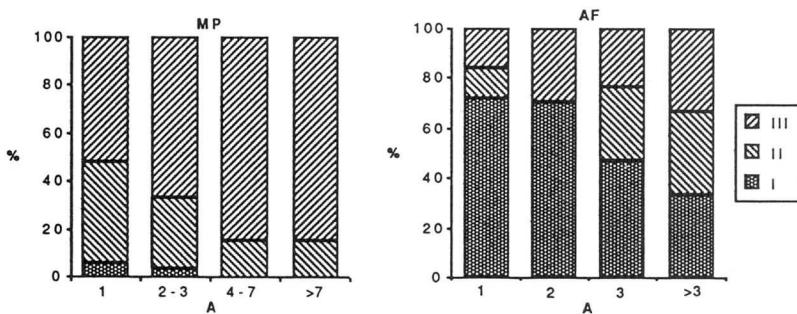


Fig. 5.4: Proportions of sequences observed under controlled conditions classified by the three behaviour types (*I*, *II*, and *III*) in function of aphid age (*A* = days after hatching) of the following species (number of observations in the four age classes in brackets): MP = *M. persicae* (35, 27, 67, 73), AF = *A. fabae* (39, 10, 17, 6).

Markov Matrices for Discrete Markov Chain Model

Only data of the first two age classes were used for the following analyses, i.e. 1-3 days for *M. persicae* and 1-2 days for *A. fabae*. The stationarity of the sequences was tested to eliminate the initial phase, when the aphids might be disturbed by the handling procedure. The transition probabilities differed significantly between the intervals I_1 (0-300 s) and I_2 (300-600 s) (*M. persicae*: df = 12, G^2 = 51.02, P < 0.0005; *A. fabae*: df = 12, G^2 = 71.23, P < 0.0005), but not between intervals I_2 and I_3 (600-900 s, *M. persicae*: df = 12, G^2 = 11.13, P = 0.518; *A. fabae*: df = 12, G^2 = 13.12, P = 0.361). The transition probabilities for the time interval $I'_2 = [D_k, D_k+300]$ were recalculated several times, changing D_k in steps of 25 s, and compared with I_3 , to determine the threshold D_k , where the sequences became stationary. The smallest value D_k , for which I'_2 did not differ from I_3 at the 5% significance level was 125 s for *M. persicae* (df = 12, G^2 = 18.87, P = 0.092) and 200 s for *A. fabae* (df = 12, G^2 = 18.14, P = 0.111). For following analyses, only the stationary part of the sequences ($k \geq D_k$) was retained.

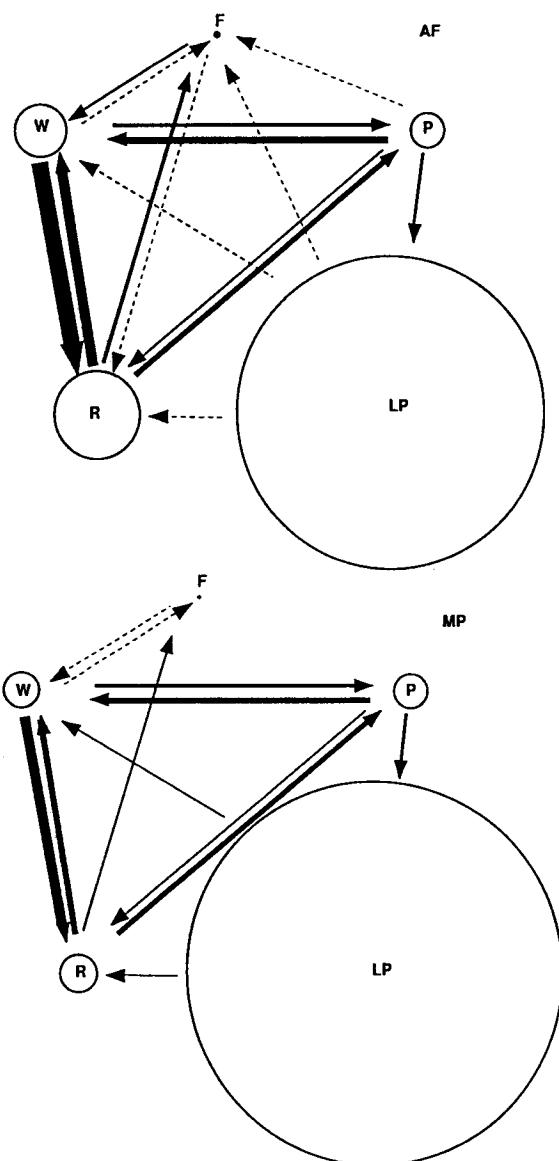


Fig. 5.5: Kinematic graphs of the stationary part of the behaviour sequences of *A. fabae* (AF) and *M. persicae* (MP) observed under field conditions. The widths of the arrow lines are proportional to the observed transition frequencies and the diameter of the circles to the proportion of total time spent in the behaviour state. Broken lines indicate sparse transitions (< five times observed). F = flight, W = walking, R = resting, P = probing, LP = 'long penetration'.

The kinematic graphs (SUSTARE, 1978; Fig. 5.5) are a representation of the transition frequencies between behaviour states and the proportion of the total time spent in a state, which has not to be confounded with the mean residence time. It seems that the behaviour sequences of the two species were qualitatively similar, but that there existed quantitative differences. *M. persicae* spent most time in state 'long penetration'. The same was true for *A. fabae* but the proportion of time spent in other states was higher. *A. fabae* showed the behaviour states walking, resting and flight more frequently than *M. persicae*.

The transition probabilities for the discrete Markov chains are given in Tab. 5.3. To test the ergodicity of the Markov chains, higher order powers of the matrices were calculated. The resulting matrices had nonzero entries in all cells, indicating that it is possible to reach any state in a finite number of steps from any other state (SNELL, 1989). The transition matrices are thus ergodic for both species, i.e. no absorbing state exists and the matrices are not periodic. Therefore the proportions of the states will converge towards a steady state with increasing time.

Ethograms of sequences, simulated with the estimated transition matrices were compared visually with the observed sequences. The comparison showed that the simulated sequences were similar to the observed ones.

Tab. 5.3: Transition probability matrices (P_C) of behaviour sequences of $MP = M. persicae$ (P_{col}) and $AF = A. fabae$ (P_{nco}) for the single-step dependence discrete Markov chain, estimated for a step of 1 s. The probabilities to pass from a state I at time k (shown in the left column) to a state J at time $k+1$ (shown in the top row) are given. The matrices were estimated for the age of ≤ 3 days (*M. persicae*, 68 sequences) and ≤ 2 days (*A. fabae*, 46 sequences) after hatching, and the stationary part of the sequences, ($k \geq 125$ s for *M. persicae* and $k \geq 200$ s for *A. fabae*).

MP	Flight	Walking	Resting	Probing	'Long penetration'
Flight	0.96179*	0.03821	0	0	0†
Walking	0.00037	0.97939	0.01497	0.00527	0†
Resting	0.00142	0.00822	0.98382	0.00654	0†
Probing	0	0.00886	0.00106	0.98613	0.00395
'Long penetration'	0	9.68E-05	4.40E-05	0†	0.99986
AF	Flight	Walking	Resting	Probing	'Long penetration'
Flight	0.96179*	0.03439	0.00382	0	0†
Walking	0.00057	0.97929	0.01691	0.00323	0†
Resting	0.00195	0.00631	0.98865	0.00310	0†
Probing	0.00029	0.01077	0.00175	0.98369	0.00349
'Long penetration'	3.22E-05	0.00013	3.22E-05	0†	0.99981

* Parameter estimated indirectly from the mean flight distance (Chapter 5.4) assuming a mean flight speed of 0.5 m/s.

† Impossible transition by definition.

5.3. Analysis of the Behaviour Model

Continuous Markov Chain Model

The parameters for the continuous Markov chains are given in Tab. 5.4.

Tab. 5.4: Transition rate matrices (Q_C) of behaviour sequences of $MP = M. persicae$ and $AF = A. fabae$ for a continuous Markov chain. The transition rates [d^{-1}] from state I (shown in the left column) to state J (shown in the top row) were calculated by Eqs. 5.6 and 5.9 from the matrices P_C in Tab. 5.3.

MP	Flight	Walking	Resting	Probing	'Long penetration'
Flight	-3365.8	3365.8	0.0	0.0	0.0
Walking	32.3	-1799.5	1307.2	460.0	0.0
Resting	123.3	716.5	-1409.8	570.1	0.0
Probing	0.0	771.0	92.2	-1206.8	343.6
'Long penetration'	0.0	8.4	3.8	0.0	-12.2
AF	Flight	Walking	Resting	Probing	'Long penetration'
Flight	-3365.8	3029.2	336.6	0.0	0.0
Walking	49.8	-1808.2	1476.4	282.0	0.0
Resting	169.4	548.1	-986.5	269.1	0.0
Probing	25.4	938.6	152.2	-1420.6	304.4
'Long penetration'	2.8	11.1	2.8	0.0	-16.7

Steady State

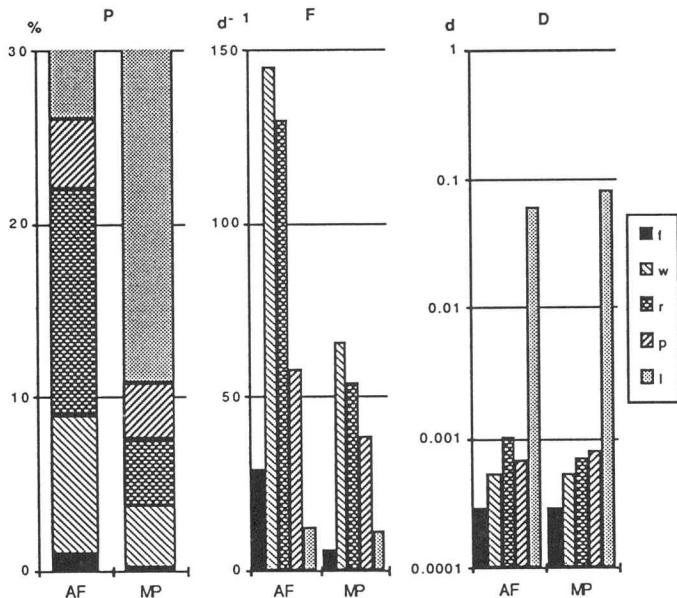


Fig. 5.6: Proportions P , frequencies F and mean residence times D (logarithmic scale) of the five behaviour states in steady state of the Markov model: $f = \text{flight}$, $w = \text{walking}$, $r = \text{resting}$, $p = \text{probing}$, $l = \text{'long penetration'}$ for $A. fabae$ (AF) and $M. persicae$ (MP).

Simulation studies of the continuous Markov models showed that the steady state was nearly reached after 1 h. In steady state, $M. persicae$ spent nearly 90 % of the total

time in state 'long penetration' (Fig. 5.6, P). *A. fabae* spent a higher proportion of time flying, walking, resting and even probing than *M. persicae*.

Marked differences between the walking, probing and especially flight frequencies of the two species were found (Fig. 5.6, F). The total frequency of behaviour state changes (= sum of all frequencies $\sum_{I \neq f} f_{IC}$) was twice as high for *A. fabae* (375.1 d^{-1}) than for *M. persicae* (174.6 d^{-1}), which confirms that noncolonizing species are in fact more 'restless' than colonizing ones.

The mean residence times seem to be similarly long in the two species, except that *M. persicae* had a longer mean residence time of 'long penetration' (Fig. 5.6, D).

Sensitivity to Changes in Transition Rates

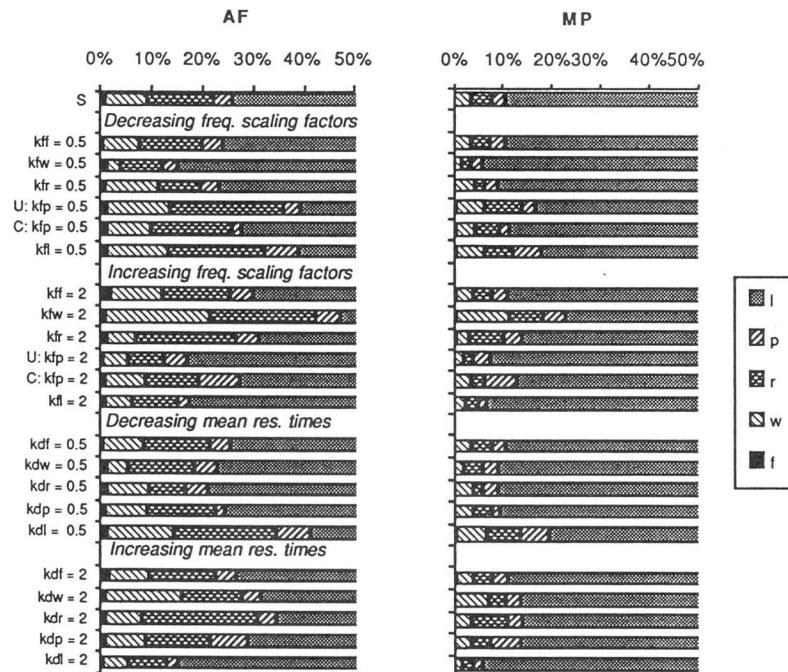


Fig. 5.7: Proportions of the five behaviour states in steady state (I = 'long penetration', p = probing, r = resting, w = walking, f = flight) for *A. fabae* (AF) and *M. persicae* (MP) after modifications of the transition rate parameters. k_{fJ} ($J \in \{f, w, s, p, l\}$) are the scaling factors of the frequencies, k_{dI} ($I \in \{f, w, s, p, l\}$) are the scaling factors of the mean residence times. S shows the steady states of the standard matrices. U: 'uncorrected model' (without correction of Eq. 5.13), C: 'corrected model'.

Reducing or increasing the scaling factor of walking frequency k_{fw} had the most significant effect on the proportions in steady state (Fig. 5.7). An increase in the scaling

factor of 'long penetration' frequency $k_{f\eta}$ reduced the proportion of I . Modifications of k_{fp} in the 'uncorrected model' (without application of Eq. 5.13) had a similar effect, whereas the proportions of states were hardly changed in the 'corrected model'.

The effect of changed mean residence times on the proportions of the states was less pronounced (Fig. 5.7). Only modifications in the mean residence time of I had a visible effect.

k_{fw} was the most important factor in determining the state change frequency. Reduction of k_{ff} , k_{fw} and k_{fs} reduced the frequency of state changes relative to the standard sequences (RF), whereas reduction of $k_{f\eta}$ increased it (Fig. 5.8). Modifications of k_{fp} in the 'uncorrected model' had a similar effect as changes in $k_{f\eta}$, whereas the total state change frequency was little affected in the 'corrected model'. Note that changing a frequency scaling factor of behaviour J does not affect only the frequency of J , but also the frequencies and durations of all other states.

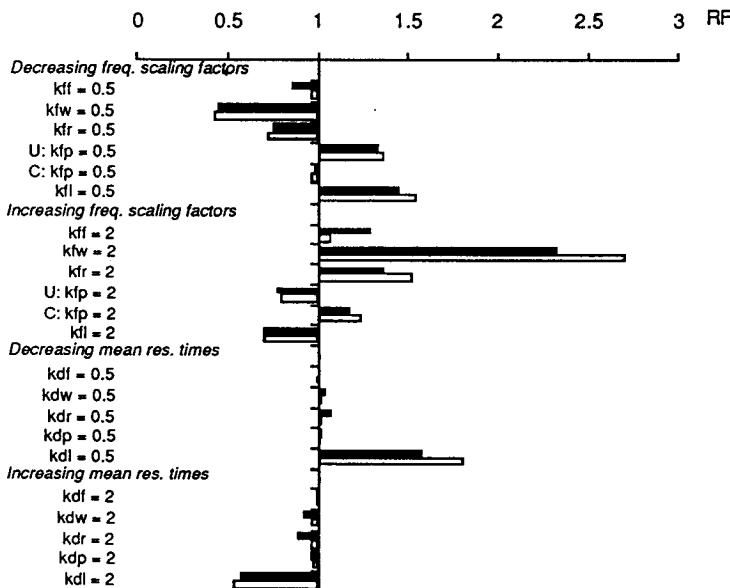


Fig. 5.8: Relative total frequencies of behaviour state changes in steady state (RF = total frequency in changed sequence) for *A. fabae* (black bars) and *M. persicae* (white bars). k_{fJ} ($J \in \{f, w, s, p, I\}$) are the scaling factors of the frequencies, k_{dI} ($I \in \{f, w, s, p, I\}$) are the scaling factors of the mean residence times. U: 'uncorrected model' (without correction of Eq. 5.13), C: 'corrected model'.

Changes of k_{fp} in the 'uncorrected model' hardly affected the probing frequency (Fig. 5.9, U), whereas probing frequency was nearly proportional to k_{fp} in the 'corrected model' (Fig. 5.9, C).

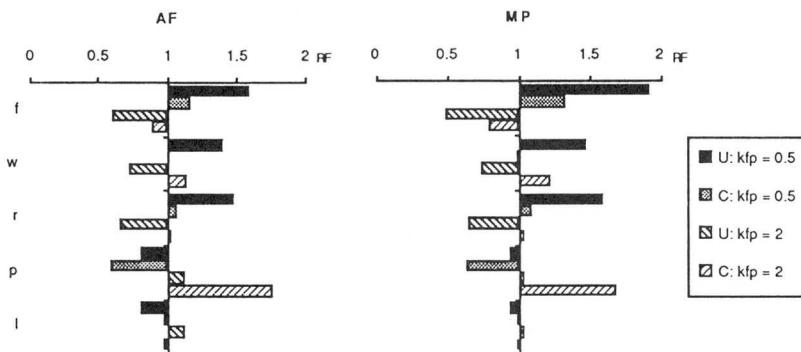


Fig. 5.9: Relative frequencies (RF) of behaviour states in steady state (f = flight, w = walking, r = resting, p = probing, l = 'long penetration') for *A. fabae* (AF) and *M. persicae* (MP). k_{fp} is the scaling factor of the probing frequency. U: 'uncorrected model' (without correction of Eq. 5.13), C: 'corrected model'.

5.4. Flight Distance and Flight Direction in Relation to Weather

Analysis of the Wind Directions

The main direction of winds suitable for aphid flight (south-east) differed markedly from the overall wind directions (north-west, south-west and south-east, Fig. 5.10, see also Fig. 5.11).

Flight Direction

Most of the aphids were flying downwind, i.e. mainly from south-east to north-west (Fig. 5.11). The differences between the flight and the simultaneous wind direction were grouped around 0 (Fig. 5.12), which means that downwind flights were the most common. Although the correlation was highly significant, the correlation coefficient was small. A comparison showed no dependence of the differences between flight direction and simultaneous wind direction (Fig. 5.12) on the wind speed measured at the Swiss Federal Agricultural Research Station of Changins.

No correlation existed between the flight directions and the azimuth of the sun (direction in the horizontal plane, $r = 0.03$, $P = 0.91$, $N = 354$).

Relating the height of loss h_f to the current wind profile showed that 97 % of those aphids that have been lost during flight, had left the boundary layer. These aphids were unable to fly upwind, because the wind speed exceeded their active flight speed, assumed as 0.5 m/s.

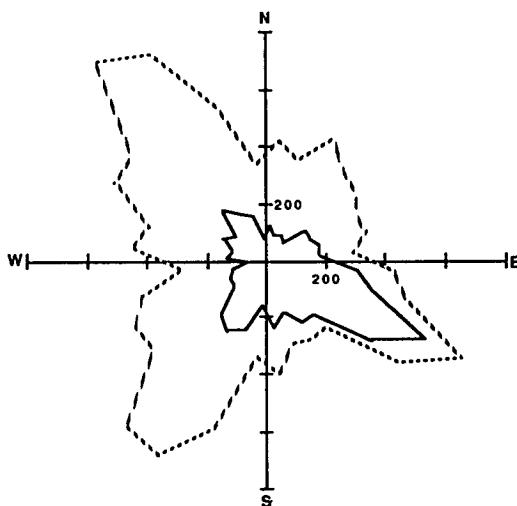


Fig. 5.10: Frequency distribution of the mean hourly wind directions measured at 12 m height at the research station in Changins from May to July of the years 1983 to 1989 of all winds (broken line) and winds suitable for aphid flight (unbroken line, hourly measurements during the day with wind speeds ≤ 2.5 m/s).

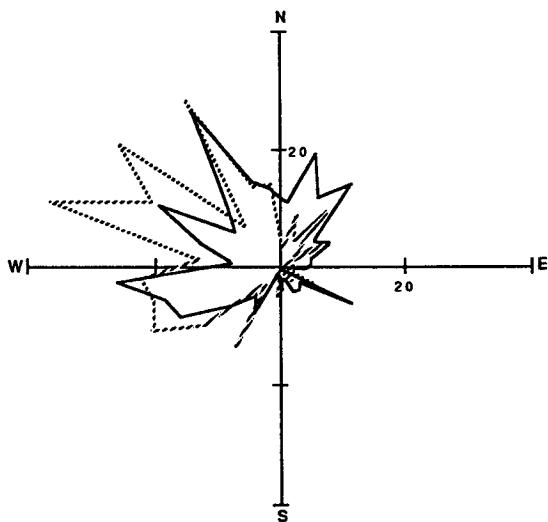


Fig. 5.11: Frequency distribution of the flight directions (solid line) of the aphids and the simultaneously measured wind directions (broken line) rotated by π (for better comparison). The distance from the origin shows the number of observations per class.

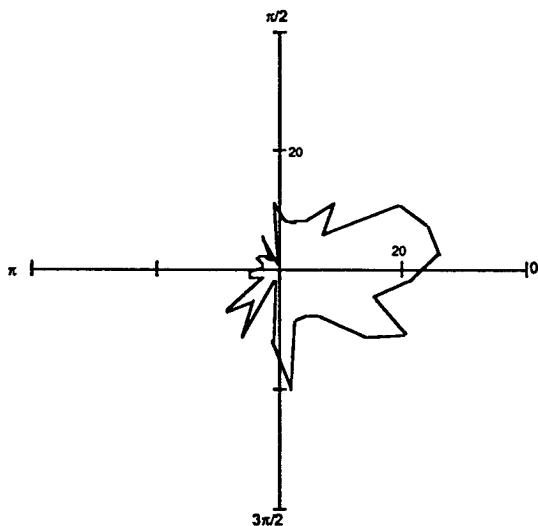


Fig. 5.12: Frequency distribution of the difference angles between the flight directions of the aphids and the simultaneously measured wind directions (rotated by π). Correlation: $r = 0.17$, $P = 8 \cdot 10^{-5}$, $N = 341$. The distance from the origin shows the number of observations per class.

Flight Distance

386 flights were observed. In 32 cases (8.3 %) the landing could be observed in other 31 cases (8.0 %) the probable flight distance could be estimated, because the aphid was lost very near the plants during downward flight. No differences between migratory and trivial flights were observed. 349 observations with complete data were retained for statistical analysis.

To estimate the minimal and probable flight distances (Fig. 5.13), the effectively observed distances were used for the observations, where aphids could be followed until landing and the results of the deterministic simulations (model A) or the averages of 20 stochastic simulation runs (model B) were taken for the flights, where the aphids were lost during flight.

The minimum flight distances, estimated by model A, had an average of 5.2 m. The estimated probable flight distances (model B) were longer (average 13.2 m). Nevertheless, short flights were prevailing: 20 % of the probable flight distances were within 1 m from the take-off site and 48 % within 5 m. Only in 13 % of the cases, the estimated landing site lied upwind of the take-off site. The comparison of the probable flight distances of colonizing and noncolonizing aphids yielded no significant difference (N colonizing = 115, N noncolonizing = 234, test value of Mann-Whitney-U-test = 14271, $P = 0.357$). Hence all observations were pooled for further analysis. A Weibull distribution with the parameters $\alpha = 0.6569$ and $\beta = 9.613$ was fitted to the results

(Fig. 5.13). The Kolmogorov-Smirnov goodness of fit test indicated that the fit was satisfactory (test value $T = 0.054$, $P = 0.256$).

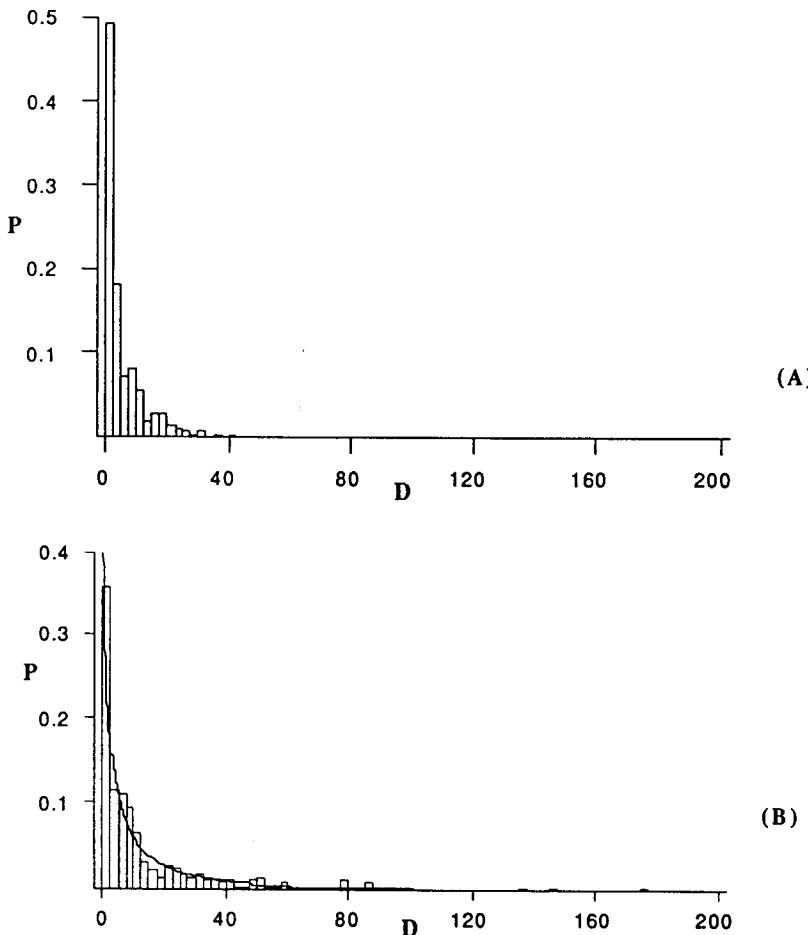


Fig. 5.13: Frequency distribution of observed and simulated flight distances of aphids. A = minimal flight distances, B = probable flight distances. Fitted Weibull distribution shown by curve. D = flight distance in m, P = proportion of observations in the respective class.

If only the observed flight distances had been used to determine the flight distance distribution, the distance would have been substantially underestimated (average distance 2.5 m), because shorter flights are more likely to be followed until landing than longer ones.

5.5. Discussion

5.5.1. BEHAVIOUR SEQUENCES

The penetration acts could be divided into short penetration (probing) and 'long penetration', which allowed an appropriate description of the behaviour sequences by the Markov model. The biological meaning of 'long penetration' is unclear. It cannot be equal to feeding for two reasons: 'long penetration' started after 3 min, but aphids need longer to reach the phloem (KLINGAUF, 1987). Moreover, some *A. fabae* also performed long penetrations (Fig. 5.3, III and Fig. 5.5), although aphids seem not to ingest phloem sap from nonhosts (MCLEAN & KINSEY, 1968). It is possible that these individuals were not able to recognize the plant as unsuitable host during short probes. Another explanation is that the handling procedure had increased the aphid's tendency to settle (JOHNSON, 1969). *M. persicae* in the state 'long penetration' penetrated towards the phloem and either started feeding or retired the stylets before reaching the phloem. Separating short and 'long penetrations' could be useful in an epidemiological model, because nonpersistent viruses are transmitted with a higher probability by short probes (VÖLK, 1959; ZETTLER, 1967; KATIS & GIBSON, 1985).

The three behaviour types that were derived (Fig. 5.2) seem to have an epidemiological interpretation. Host-rejection behaviour (*I*) is probably the most important for PVY spread, since the alatae disperse by flight and can transmit viruses over longer distances, if they probe before take-off and after landing. Successful virus transmission would require that an aphid visits other potato plants in the proximity of the source, which is likely to occur (see Chapter 5.4). Searching behaviour (*II*) can be relevant, if the aphids walk onto another plant. Otherwise virus spread is restricted to other stems of a plant. Since walking over long distances is unlikely (FERRAR, 1969), mainly the neighbouring plants can be infected by aphids showing behaviour type *II*. Settling behaviour (*III*) will rarely lead to virus transmissions, because the aphids are most time performing 'long penetrations' and hardly change to other behaviour or location.

The experiments have shown that behaviour observation under controlled conditions can yield results substantially different from field observations (Fig. 5.3). The laboratory sequences could thus not be used for the parameter estimations, as intended in the beginning. The differences in flight frequency could be explained by the fact that the air was completely calm in the cages, whereas in the field, there was always some air motion. KENNEDY (1990) has shown that more aphids take off in slight wind and in sequences of wind and calm than in continuous calm. Behaviour observations in a climatic chamber should always be compared with field observations.

The procedure of taking the aphids from a petri dish and transferring them to a potato plant in the field is artificial. Disturbance can increase the probability of flight (DIXON & MERCER, 1983) and the handling procedure the tendency of settling (JOHNSON, 1969). Taking only the stationary parts of the sequences can eliminate such effects, so that the matrices probably represent average behaviour sequences on potato plants.

Older alatae moved less and were more likely to settle than younger individuals (Fig. 5.4). Settled aphids of many species, among them *A. fabae* and *M. persicae*, autolyse their wing muscles and become unable to fly after two to three days (JOHNSON,

1957 & 1959). Later they can disperse only by walking. Alatae of colonizing species therefore contribute to virus spread mainly during the first three days after hatching. As settled aphids rarely move, they can be ignored in epidemiological studies. Unsettled *A. fabae* on potato plants (nonhost) were able to fly until death. Their wing muscles were apparently not autolyzed, because they could not find a suitable host. Noncolonizing aphids remain potential vectors, until they settle on a suitable host. Older *A. fabae* were excluded from the parameter estimations, since their behaviour might be unnatural, due to starvation in petri-dishes during days. The estimated transition matrices can be considered as representative for the epidemiologically relevant behaviour of unsettled alatae of the two species.

Markov models are simple models and probably rough simplifications of the behaviour sequences of aphids. Nevertheless, they have often been successfully applied to behaviour sequences (CANE, 1978; METZ *et al.*, 1983). Comparison of observed and simulated sequences indicated that the underlying assumptions of the Markov model (stationarity, exponential distribution of the residence times and single-step dependence) were consistent with the data.

The behaviour sequences of *M. persicae* and *A. fabae* (Fig. 5.5) seem to be qualitatively similar. The sequence of walking and probing is characteristic after landing on a plant (KLINGAUF, 1987) and is observed even on materials such as glass. *A. fabae* showed nevertheless more frequently the relevant behaviour sequences for epidemiology, namely probing → (walking, resting) → flight → (walking, resting) → probing. The epidemiologically relevant behaviour types I and II occurred more frequently in *A. fabae* than in *M. persicae* (Fig. 5.3).

Since the differences in behaviour within the groups of colonizing resp. noncolonizing species were small (Fig. 5.3), it seems possible to extrapolate the results of *M. persicae* to other colonizing and of *A. fabae* to other noncolonizing species.

5.5.2. BEHAVIOUR MODEL

The observation periods were relatively short. This might cause a bias, which can be eliminated by using the steady state of the Markov models for simulation studies. If the observations had been long enough, the observed behaviour characteristics should approach the values calculated for the steady state.

Although the behaviour sequences were qualitatively similar (Fig. 5.5), there existed important quantitative differences in flight, walking and probing frequencies and in the time spent in 'long penetration' (Fig. 5.6). Similar differences have also been observed for other aphid-host systems (KENNEDY *et al.*, 1959; MCLEAN & KINSEY, 1968). The observed differences might be merely due to the negative correlation between the proportion of time spent in 'long penetration' and the frequency of all other states, including probing frequency. Since colonizing aphids are more inclined to settle, they will penetrate longer than noncolonizing aphids, resulting in less frequent movement and probing.

Out of the scaling parameters, k_{fw} and k_{dl} influenced the behaviour sequence most. k_{fw} was the most important parameter determining the frequency of behaviour state changes, probably because walking was the most frequent behaviour (Fig. 5.6, F). Increasing 'long penetration' duration by k_{dl} led to a reduced frequency of state changes.

The mean residence time of 'long penetration' was so long that this state acted as a 'quasi-absorbing' state, leading to a reduced frequency of all states. Increasing k_{η} had the same effect as increasing k_{dl} , which was exactly the opposite of the effect of other frequency scaling factors. Walking and 'long penetration' mark therefore the extremes on a scale of 'restlessness'. Aphids having high walking frequency can be considered as restless, whereas long mean residence times of l mark 'settling' behaviour. To characterize behaviour sequences of aphids, it might be sufficient to quantify (1) the frequency of walking and (2) the mean duration of the penetrations, resp. the proportion of time spent penetrating. It seems that the most frequent behaviour and the behaviour with the longest mean residence time determine the characteristics of a behaviour sequence. Instead of the walking frequency, the total frequency of behaviour state changes could be estimated.

Changes of k_{fp} in the 'corrected model' seem to reflect better what will happen, if probing frequencies are changed in nature, than in the 'uncorrected model' for the following reasons: probing frequency seems to be proportional to k_{fp} in the 'corrected model' but not in the 'uncorrected model'. *A. fabae* had a higher probing frequency than *M. persicae*, but spent less time in state 'long penetration', had a higher state change frequency and a similar absolute transition rate probing → 'long penetration' ($= p_{pC} q_{plC}$) as *M. persicae*. All these findings are consistent with the 'corrected model' but in contradiction with the 'uncorrected model'. The former model was thus preferred to the latter and applied in the simulation studies of the EPOVIR-model in Chapter 7.

5.5.3. FLIGHT BEHAVIOUR

Flight and Wind Direction

An analysis of the wind directions showed that winds were blowing mainly from north-west during the night. North-eastern and south-western winds were usually strong and therefore unsuitable for aphid flight, whereas south-eastern winds were weak and the most frequent ones during summer days. This breeze, suitable for aphid flight, blows from the lake of Geneva towards the Jura mountain chain, due to different warming by sunshine. The results (Fig. 5.10) emphasize the necessity to consider only the wind directions during the phases, when aphids fly.

KENNEDY & THOMAS (1974) have observed that flying aphids orient upwind. If the wind speed exceeds their flight speed, they are carried with the wind, but their bodies are orienting upwind. The field observations showed that almost all aphids had left the boundary layer and therefore were carried downwind (Fig. 5.11 and 5.12). Lateral movement in a direction not parallel to the wind direction was also common (Fig. 5.12). In 51% of the observations the flight direction deviated more than $\pi/4$ from the wind direction. A part of this variation could be explained by measurements errors, i.e. the actual deviation of the flight direction from the wind direction might be smaller than estimated. The wind direction changes quickly during periods of low wind speed. It is possible that the fisherline trap does not react immediately to a changed wind direction. Further the wind direction at the height of the fisherline trap (1 m) could differ from that at a height of several meters above soil. KENNEDY & THOMAS (1974) have observed that the higher the wind speed, the stronger the correlation between flight and wind direction. This could

not be confirmed for the presented data set, perhaps because the wind speed was measured 2 km away and not exactly at the same time. The direction of the flight movement seemed not to be influenced by the position of the sun. Probably most aphids performed trivial flights and were not attracted by short wave light like during the distance flight.

Since a prevailing wind direction exists, a concentration of infections downwind of the infection sources might occur, as it has been found by IRWIN (1981) for soybean mosaic virus in a soybean field. In another system (white-fly transmitted viruses in the cassava crop) a concentration of infections upwind within a short distance from the source has been observed (N'GUETTA *et al.*, 1986). The vectors of the cassava virus diseases used the boundary layer to fly upwind and therefore most infections occurred upwind of the sources. However, the boundary layer is much deeper in the cassava, due to the crop height of about 1.8 m. In the presented field observations only few aphids had travelled upwind. Although south-east was the prevailing wind direction in the experimental fields during aphid flight periods, neither in south-east nor in north-west direction of the infection sources the infections were more frequent than in the other directions (see also Fig. 7.17). It must be kept in mind that even if there is a prevailing wind direction during aphid flight periods, there remains a considerable variation of wind directions (note that the hourly means were used in Fig. 5.10). Moreover, although highly significant, the coefficient of the correlation between flight and wind direction is small. This might explain, why PVY appears to spread in all directions from the source with similar frequency. The topic is further discussed in Chapter 8.3.

Flight Distance

It can be assumed that the aphids were already in a post-migratory flight phase, because they had passed a hunger period or were collected in the field (most of these individuals had immigrated from outside the field). Hence the observed flights were probably trivial. Trivial flights are of highest interest for within-field epidemic studies.

Short flight distances were prevailing (Fig. 5.13). The estimated flight distances were much shorter than the post-migratory flights observed by TAIMR & KRIZ (1978) for *Phorodon humuli* (SCHRANK) in a hop garden. The authors found marked aphids in a distance of 49 to 149 m from the release site during the days after the release. However, from the results shown in Chapter 5.4 it is concluded that most of these aphids had probably performed more than one trivial flight. A further indication that trivial flight distances are actually short, are the steep spatial disease gradients of many virus diseases (THRESH, 1976 & 1978), including PVY and PLRV. These gradients can only partly be attributed to wingless aphids. It can be concluded that most aphids performing trivial flights will land again in the same field and can potentially transmit viruses.

The parameters describing the flight distance distribution are certainly rough estimates. However, the simulation studies of the model 'Spatial Epidemic' have shown that the disease incidence and the spatial pattern of the infections are insensitive to mean dispersal distances exceeding 10 m. These rough estimates seem thus to be sufficiently accurate for the model EPOVIR (Chapter 6).

Although the epidemic is insensitive to mean dispersal distances exceeding 10 m, assuming the virus spread to be completely random in space for the EPOVIR-model is inadequate for the following reasons: virus transmission by walking might play a role and

significantly affect the spatial pattern. Nonrandom distribution is more important, if the initial disease incidence is low, which is the usual case in seed potatoes. Lower values of α of the Weibull distribution (like the one found in the experiments) leads to lower disease incidences and more marked clumping (Fig. 4.3). Finally, a random model cannot reproduce the nonrandom patterns observed in the field experiments (GREGORY & READ, 1949).

The flight distances of colonizing and noncolonizing aphids might have been different, if different parameters had been assumed for the two aphid groups. However, studies of KENNEDY (1965) showed no clear differences in the rates of climb of *A. fabae* after landing on host and non-host leaves. Thus the assumption of the same parameters for both groups seems justified. Colonizing and noncolonizing aphids will have a similar chance to land again in the same field and both groups can potentially contribute to within-field virus spread.

The flight distance might be influenced by the wind speed, but no clear dependence could be found, probably because aphids flew only when the wind speed was low. The estimated Weibull distribution describes flight distances during conditions suitable for flight, i.e. when wind speed is low.

6. EPOVIR: MODEL OF THE EPIDEMIOLOGY OF POTATO VIRUSES

6.1. The Concept of Inoculation and Infection Submodels

Existing epidemic models of vector transmitted plant diseases combine all modelled epidemic processes in one model (e.g. MARCUS & RACCAH, 1986; RUESINK & IRWIN, 1986; SIGVALD, 1986). Although useful for small models, this procedure becomes more and more intractable for larger and more complex models. Pathosystems of aphid-borne virus diseases tend to be more complex than fungal pathosystems because of the vectors as an additional element. An alternative concept is proposed, which will subsequently be applied in the EPOVIR¹-model.

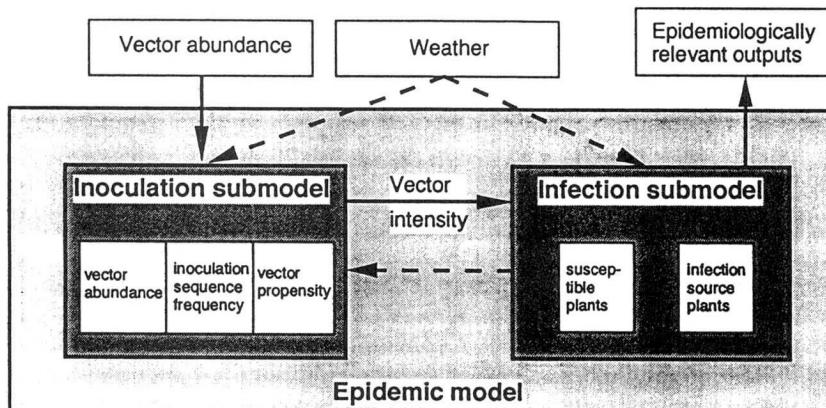


Fig. 6.1: Concept of inoculation and infection submodels. Unbroken lines symbolize compulsory, broken lines optional inputs, resp. outputs.

According to the concept, schematized in Fig. 6.1, aphid-borne virus epidemic models are divided into two submodels:

- *Inoculation submodel*: calculates the vector intensity, which is the number of inoculations caused per infection source and time unit. RUESINK & IRWIN (1986) defined an inoculation as the 'successful insertion of virus into a host plant' and an infection as the inoculation of a susceptible plant. The inoculation submodel for PVY is independent of the infection submodel and acts as an input model for the latter.

¹EPOVIR is an acronym for epidemiology of potato viruses.

- *Infection submodel*: calculates the number of infections using vector intensity as input. Optionally it can also define the spatial pattern of the infections. The model calculates the epidemiologically relevant outputs, such as proportions of diseased, healthy and infection source plants and possibly the spatial pattern.

The concept can be applied to simple models (e.g. MADDEN *et al.*, 1990), as well as to more complex models. It is applicable to all currently known epidemic models of virus-vector systems. The advantages of this approach are more clarity due to the modularization, easier comparison between models and the possibility to exchange and to couple different submodels.

6.2. Model Structure

The border of a potato field was taken as the spatial system boundary. The time domain of the model is one growing season from crop emergence t_{em} to haulm-killing t_{hk} .

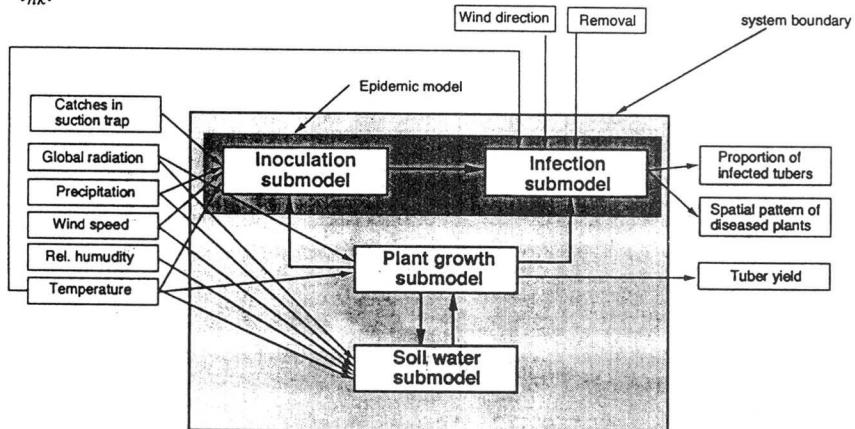


Fig. 6.2: Structure of the EPOVIR-model.

The EPOVIR-model consists of four submodels (Fig. 6.2):

- *Inoculation submodel*: calculates the number of inoculations per infection source (vector intensity) caused by the vector population by different types of interplant movement. Differential equation system of forth order.
- *Infection submodel*: determines the spatial distribution of the inoculations, the occurrence of infections and the disease state of individual plants. The proportion of infected tubers is a model output. Discrete time stochastic cellular automaton (WOLFRAM, 1986) with discrete states.
- *Plant growth submodel*: calculates the physiological state of the canopy, the leaf area and the tuber yield of the potato crop. The physiological state of the leaves

is used to estimate the age resistance function in the infection submodel.
Difference equation system of sixth order.

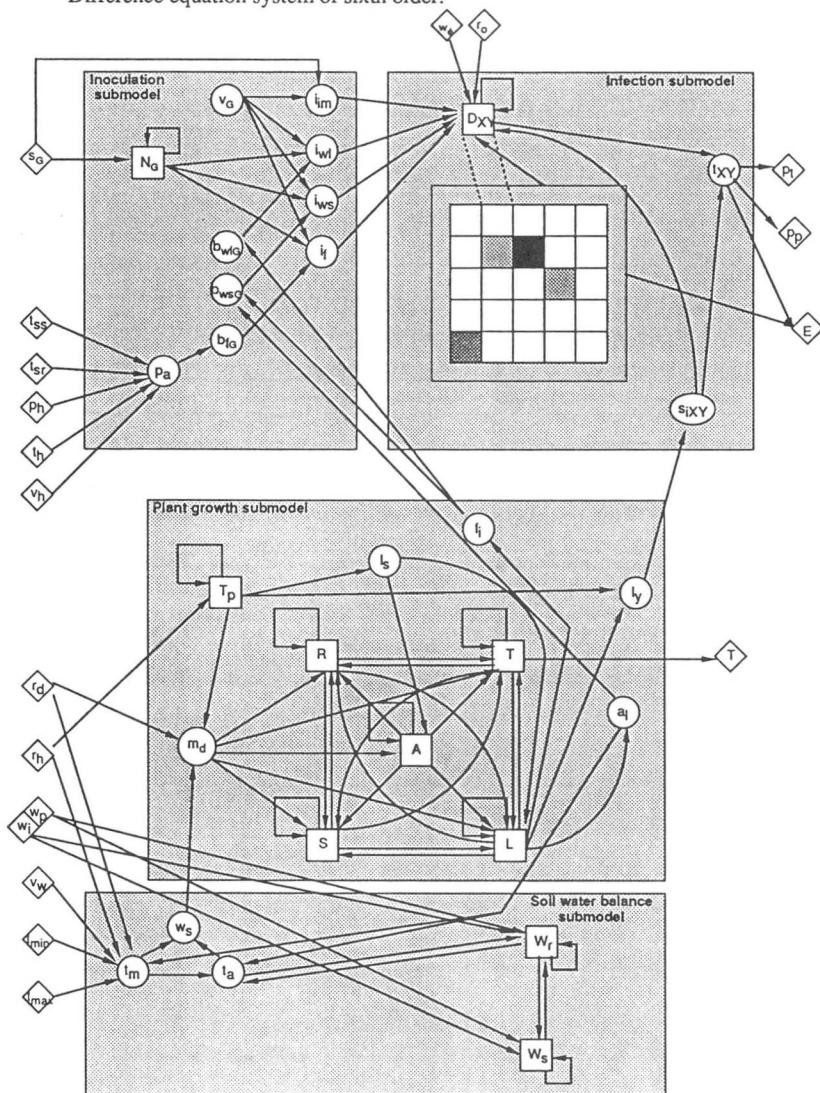


Fig. 6.3: Relational graph (for the predicate $P(x, y) ::= x \text{ influences } y$) of the EPOVIR-model. Boxes = state variables, circles = auxiliary variables, rhombi = global inputs resp. outputs. The relations are shown only for one vector group and one cell of the cellular automaton.

- *Soil water submodel:* calculates the water content of the soil and the water stress of the potato plants. Difference equation system of second order.

The plant growth and soil water balance submodels act as input models for the epidemic model, but are independent of the latter. Fig. 6.3 shows the relational graph of the EPOVIR-model.

6.3. Model Equations

6.3.1. INOCULATION SUBMODEL

Vector intensity, i.e. the potential number of infections, caused by the alatae located on one source plant per time unit, is calculated as the product of vector abundance N_G , inoculation sequence frequency and vector propensity v_G . An inoculation sequence is defined as the sequence of behaviour events necessary for a successful inoculation, i.e. probing, movement to another plant and further probing. Vector propensity after IRWIN & RUESINK (1986) is a measure of the average species specific ability to transmit a virus disease. It is the probability to transmit a virus disease if a vector moves from a source plant to a susceptible plant and probes on both. The three factors represent the different requirements for successful virus inoculations: vectors must be present in the field (abundance), they must probe on the plants and move (inoculation sequence frequency) and they must be able to transmit the virus (vector propensity).

Vector intensity is calculated separately for three types of within-field movement: by flight i_f , by walking over leaf bridges i_{wl} and by walking on bare soil i_{ws} and for immigrating aphids i_{im} .

vector intensity [#plant ⁻¹ d ⁻¹]	vector abundance [#/plant]	inoculation sequence frequency [d ⁻¹]	vector propensity [%/100]	
i_f	$= \sum_{G=1}^3 (N_G p_{piC} f_{pfC})$	$p_{piC} f_{pfC}$	v_G)	(6.1)
i_{wl}	$= \sum_{G=1}^3 (N_G p_{piC} f_{pwIC})$	$p_{piC} f_{pwIC}$	v_G)	(6.2)
i_{ws}	$= \sum_{G=1}^3 (N_G p_{piC} f_{pwSC})$	$p_{piC} f_{pwSC}$	v_G)	(6.3)

	[#plant ⁻¹ d ⁻¹]	[#plant ⁻¹ d ⁻¹]	[%/100]	[%/100]	
i_{im}	$= \sum_{G=1}^3 (m_{iG} p_{piC})$	p_{piC}	v_G)		(6.4)

p_{piC} is the average probability that an aphid probes on the infection source plant, f_{pfC} , f_{pwIC} and f_{pwSC} are the frequencies of interplant movement by flight, walking over leaf bridges resp. walking on soil. Eq. 6.4 is slightly different from the other three equations, since the frequency of movement is included in the immigration rate m_{iG} . In the following the three factors in the equations are developed.

Vector Abundance

The model includes only alatae, which are believed to be responsible for most PVY transmissions (BROADBENT & TINSLEY, 1951). Virus transmission by apterae is ignored. The abundance of unsettled alatae is calculated for four vector groups, derived from the transmission tests of DERRON & GOY (1990). The authors found no significant differences in PVY-transmission within vector groups 1, 2 and 3. Only those species were considered that appear in significant numbers in potato fields. This constellation might be specific for Swiss conditions and must be reevaluated, if the model is applied under different conditions. Vector group 2 was divided into group 2a (colonizing) and 2b (noncolonizing) vectors. Groups 1 and 3 have only colonizing resp. noncolonizing species (Tab. 6.1).

Tab. 6.1. Vector groups in the EPOVIR-model.

Vector group (G)	Vector species	Colonization group
1	<i>Myzus persicae</i>	
2a	<i>Aphis nasturtii</i> <i>Macrosiphum euphorbiae</i>	colonizing
2b	<i>Aphis fabae</i> <i>Brachycaudus helichrysi</i> <i>Phorodon humuli</i>	noncolonizing
3	<i>Acyrtosiphon pisum</i> <i>Rhopalosiphum padi</i>	

The model assumes alatae to be uniformly distributed over all plants in the field. The abundances N_1 , N_{2a} , N_{2b} and N_3 of the vector groups are the four state variables of the inoculation submodel. Each vector group could also be considered as a distinct subsystem, since these states have no interactions in between.

Unsettled alatae abundance N_G is calculated by the immigration rate m_{iG} , the relative emigration, mortality and settling rates (m_{eC} , d_C and s_{rC} , Eq. 6.5). The immigration rate is vector group specific (G), the other three rates colonization group specific (C). It is assumed that settled alatae move no longer and are therefore irrelevant for epidemiology. Settling is thus treated like death in the model and is added to the mortality rate. Since alatae cannot give birth to alatae, the birth rate term is dropped.

$$\frac{d N_G}{dt} = m_{iG} - (m_{eC} + d_C + s_{rC}) N_G \quad (6.5)$$

Since noncolonizing aphid will not settle, their settling rate is 0. The relative emigration rate m_{eC} is defined by the flight frequency f_{fC} , the conditional probability that an alata will leave the field after take-off p_{em} and the fraction of day with favourable flight conditions p_a :

$$m_{eC} = f_{fC} p_{em} p_a \quad (6.6)$$

p_a is calculated assuming that four factors prevent take-off and subsequent flight: too low temperatures, too high wind speed, precipitation and low light intensity during the night. Take-off is possible if the mean hourly temperature t_h exceeds a threshold f_{th} , the hourly precipitation p_h is below a threshold f_{hp} , the mean hourly wind speed v_h is below a threshold f_{hv} and it is daylight (Eq. 6.7).

$$\begin{aligned}
 p_a &= \frac{\sum_{h=1}^{24} c_h}{24} \\
 c_h &= \begin{cases} 1 & \text{if } t_h \geq t_{su} \wedge p_h \leq f_{tp} \wedge v_h \leq f_{tw} \wedge t_{sr} \leq h_m \leq t_{ss} \\ 0 & \text{otherwise} \end{cases}
 \end{aligned} \tag{6.7}$$

h_m is the midpoint of hour h . If h_m is before sunrise t_{sr} or after sunset t_{ss} , no take-off can occur during the hour h . p_{em} was estimated using the estimated flight distance distribution (Chapter 5.4), assuming that aphids emigrate if their flight distance exceeds a threshold distance of 50 m.

The immigration rate m_{iG} (Eq. 6.8) is calculated from the aerial aphid abundance s_G , estimated by a Taylor suction trap. TAYLOR & PALMER (1972) calculated deposition rates, treating aphids as inert particles, under the assumption of a density profile and a mean flight time. This deposition rate s_d is adopted in the model. CARTER *et al.* (1982) have observed that the effective immigration rates were higher than the deposition rates estimated by TAYLOR & PALMER (1972). To account for this, the authors introduced a concentration factor s_C , which is also used in the EPOVIR-model. Since the deposition rate is calculated per area unit, the equation must be divided by the plant density d_{pl} to obtain the immigration rate per plant:

$$m_{iG} = \frac{s_G s_d s_C}{d_{pl}} \tag{6.8}$$

Inoculation Sequence Frequency

The underlying continuous Markov chain behaviour model was described in Chapter 5.1.2. The auxiliary variables flight and probing frequency (f_{fC} and f_{pC}) and the proportion of time spent walking p_{wc}^* used in the EPOVIR-model were estimated from the steady state of the Markov model. The state proportions in steady state p_{IC}^* were estimated by simulation of the continuous Markov chain model (see Eq. 5.3). The simula-

tion was stopped, if the relative change of p_{IC} approached 0, i.e. $\left| \frac{dp_{IC}}{dt} \right| / p_{IC} < 0.0005$.

$f_{pfC}, f_{pwlc}, f_{pwsC}$ are the frequencies of interplant movement by flight (f), walking over leaf bridges (wl) and walking on bare soil (ws).

The flight frequency f_{fC} is calculated by Eq. 5.4. To calculate f_{pfC} , the flight frequency is corrected for the proportion of aphids emigrating during a flight p_{em} and the proportion of time, when take-off is possible p_a .

$$f_{pfC} = f_{fC} (1 - p_{em}) p_a \tag{6.9}$$

The frequency of interplant movement by walking could neither be estimated from literature nor from experimental data. The estimation was based therefore on the proportion of time spent walking in steady state of the model p_{wc}^* , the rate of interleaf movement during walking f_{lmC} and the probability that an interleaf movement leads to an interplant movement (p_{wl} resp. p_{ws}).

$$f_{pwsC} = p_{wc}^* f_{lmC} p_{ws} \tag{6.10}$$

$$f_{pwlc} = p_{wc}^* f_{lmC} p_{wl} \tag{6.11}$$

The conditional probability p_{ws} that an aphid walks off a plant, given an interleaf movement occurs, was estimated based on the assumption that aphids located on the lowest two leaves of each stem will leave the plant and walk off the plant on bare soil, if they move downwards. The vertical distribution of aphids on the plant is assumed uniform and the probabilities of upward and downward moves to be equal. p_{ws} was estimated by the ratio of the area of the lowest leaf a_{ll} of each stem to the total leaf area a_l of the plant.

$$p_{ws} = \text{MIN} \left(\frac{a_{ll}}{a_l}, 1 \right) \quad (6.12)$$

The conditional probability p_{wl} of an interplant movement by leaf bridges, given an interleaf movement occurs, was estimated by the spatial arrangement of the plants and the proportion of area covered by the canopy f_i . p_{wp} is the probability that an interleaf movement leads to a movement to a neighbouring plant, given these two plants touch. A plant is in leaf contact with two plants, if the plants touch within the row and with four plants, if they touch also between rows (see also Fig. 6.6).

$$p_{wl} = \begin{cases} 0 & \text{if } f_i < t_{tw} \\ 2 p_{wp} & \text{if } t_{tw} \leq f_i < t_{tb} \\ 4 p_{wp} & \text{otherwise} \end{cases} \quad \begin{matrix} (\text{plants don't touch}) \\ (\text{plants touch within row}) \\ (\text{plants touch also between rows}) \end{matrix} \quad (6.13)$$

t_{tw} and t_{tb} are the thresholds for f_i , where the plants begin to touch within, resp. between rows. p_{piC} (Eqs. 6.1 to 6.4) is the probability that an aphid will make at least one probe on the infection source plant, not longer than the 'long penetration' threshold. It is estimated by the probability that an aphid probes during the mean time spent on a plant t_{plC} . The latter is the reciprocal of the frequency of interplant movement. Eq. 6.14 assumes that probes occur randomly in the behaviour sequence (Poisson probability). The probing frequency f_{pC} is calculated by Eq. 5.5.

$$p_{piC} = 1 - e^{-f_{pC} t_{plC}} \quad (6.14)$$

$$t_{plC} = \frac{1}{f_{pC} P_a + f_{pwlC} + f_{pwsC}} \quad (6.15)$$

Vector Propensity

Vector propensity v_G is different for each vector group. It is calculated from the parameters vector efficiency e_G , serial inoculation probability t_{sG} and retention rate r_{fG} .

$$v_G = \frac{e_G}{1 - t_{sG} e^{-r_{fG}/f_{pC}}} \quad (6.16)$$

e_G is the probability of PVY transmission during the first probe immediately after virus acquisition. To account for the fact that aphids can still be infective after several probes, e_G is corrected by the serial inoculation probability t_{sG} , which is the probability that a vector will transmit the virus during a probe, compared with the probability during the previous probe. An aphid having probed on an infection source will thus have a probability of a successful inoculation of $e_G t_{sG}^0 = e_G$ during the first probe, $e_G t_{sG}^1$ during the second probe and $e_G t_{sG}^{n-1}$ during the n -th probe. $\mathbb{V}(1, I - t_{sG})$ is the sum of the powers

$$\sum_{n=0}^{\infty} t_{sG}^n, \text{ i.e. the cumulated transmission probability after an infinite number of probes.}$$

The inoculation probabilities are added to give the total number of inoculations (Eqs. 6.1 to 6.4).

Since infectivity by nonpersistent viruses is lost fast, vector propensity is corrected by the retention rate r_{fG} . SYLVESTER (1988) has proposed to use a negative exponential model for the inoculation probability after virus acquisition (Eq. 6.32 and Fig. 6.5), which is supported by experimental results (KOSTIW, 1975; PRÖSELER & WEIDLING, 1975; VAN HOOF, 1980; KATIS & GIBSON, 1985). t_{SG} is multiplied by $e^{-r_{fG} t_{fpc}}$ to account for the decline in inoculation probability between two probes.

The interval between the last probe before take-off and the first probe after landing did not exceed few minutes in most behaviour sequences (Chapter 5.2), therefore the loss of infectivity until the first probe on the target plant was ignored (otherwise v_G should have been multiplied by $e^{-r_{fG} t_{fpc}}$).

6.3.2. INFECTION SUBMODEL

The infection submodel of the EPOVIR-model is an extension of the model 'Spatial Epidemic' (Chapter 4.1). The disease states D_{XY} and possible transitions are given by Tab. 4.1 resp. Fig. 4.1, the initial state of the cellular automaton is defined as in Chapter 4.1 (except in the validation experiments, see Chapter 7.2.1).

The field is treated as a torus analogously to the model 'Spatial Epidemic' (Eq. 4.1 and 4.2) or as a finite field. In the latter case inoculations with coordinates $[i_X, i_Y]$ fulfilling the condition in Eq. 4.1 are ignored. The processes of immigration and emigration are unaffected by the simulation of a torus or a finite field.

The model uses the number of inoculations caused by aphids flying in the field i_f , walking over leaf bridges i_{wl} , on bare soil i_{ws} and caused by immigrating aphids i_{im} as inputs (see Eqs. 6.1 to 6.4). The four variables were distinguished, since each type of inoculation follows another spatial probability distribution. The number of inoculations (real number) is transformed to integer numbers by Eq. 4.3. A healthy plant, which is inoculated, becomes a latently infected plant. An inoculation of a cell with another state than δ_h does not change the state of this cell.

Inoculations Caused by Aphids Flying in the Field

Using the number of inoculations i_f per source plant, plants are sampled i_f^* -times (according to Eq. 4.3) to become inoculated. This is repeated for each source plant, each time determining the number of inoculations by Eq. 4.3. The same procedure is applied also to the inoculations caused by aphids walking over leaf bridges and on soil.

The distance of the inoculation from the infection source i_{df} is sampled from a Weibull distribution with the parameters α and β (Tab. 4.2). The direction $i_{\phi f}$ is sampled from a von Mises distribution (VON MISES, 1918), sometimes also referred as 'circular normal' distribution. Its probability density function is given by:

$$\phi(\theta) = \frac{e^{\kappa \cos(\theta - \mu_0)}}{2\pi I_\phi(\kappa)} \quad (6.17)$$

$$i_{\phi f} = VM(\mu_0, \kappa) \quad (6.18)$$

where VM is a von Mises distributed random variable with the mean angle μ_0 and the so-called 'concentration parameter' κ . $I_\phi(\kappa)$ is a modified Bessel function, which

makes the integral over the circle equal to 1. The mean angle μ_0 (ZAR, 1984) is calculated by the mean wind direction during the favourable period for flight, calculated from hourly mean wind directions.

The metric coordinates $[i_x, i_y]$ of the inoculated cell are given by Eq. 4.5, the cell coordinates $[i_X, i_Y]$ by Eq. 4.7.

Inoculations Caused by Aphids Walking over Leaf Bridges

As soon as the plants touch within the row ($f_i \geq t_{lw}$), walking over leaf bridges on the neighbouring plants in the row is possible. If the plants touch also between the rows ($f_i \geq t_{lb}$), an aphid can reach also the two neighbouring plants across the rows.

$$\begin{aligned} i_X &= s_X + p_2 \frac{p_1}{|p_1|} & (6.19) \\ i_Y &= s_Y + (1-p_2) \frac{p_1}{|p_1|} \\ p_1 &= u_1 - 0.5 \\ p_2 &= \begin{cases} 0 & \text{if } u_2 > 0.5 \vee f_i < t_{lb} \quad (\text{movement within row}) \\ 1 & \text{otherwise} \quad (\text{movement between rows}) \end{cases} \\ u_{1/2} &= U[0,1] \end{aligned}$$

s_X and s_Y are the coordinates of the infection source plant. p_2 defines, whether movement is within the row or across rows. p_1 determines the sense of the movement (positive or negative).

Inoculations Caused by Aphids Walking on Soil

The distance walked by an aphid on soil i_{dws} is sampled from an exponential distribution (Tab. 4.2) with the mean walking distance d_w . The directions $i_{\phi ws}$ are uniformly distributed (Eq. 4.4). i_X and i_Y are calculated by Eqs. 4.5 and 4.7.

Inoculations Caused by Immigrating Aphids

These inoculations are distributed uniformly over the field, i.e. each plant has the same probability to become inoculated. The number of inoculations i_{im}' in the field is calculated by the vector intensity i_{im} per plant, multiplied by the number of plants in the simulated field $n_r n_p$ and the proportion of immigrating aphids coming directly from an infection source outside the field v_{im} . Contrary to the previous three types of inoculations, i_{im}' is the number of inoculations per plant, whereas i_f , i_{wl} and i_{ws} are the number of inoculations per source plant.

$$i_{im}' = i_{im} n_r n_p v_{im} \quad (6.20)$$

The metric coordinates of the inoculated cell $[i_X, i_Y]$ are sampled from a two-dimensional uniform distribution $U[0, n_r d_r]$, resp. $U[0, n_p d_p]$, where $n_r d_r$ and $n_p d_p$ are the width and length of the field.

$$\begin{aligned} i_X &= U[0, n_r d_r] & (6.21) \\ i_Y &= U[0, n_p d_p] \end{aligned}$$

The cell coordinates $[i_X, i_Y]$ are given by Eq. 4.7.

Latent Period and Disease State Transitions

The age resistance function is determined by the susceptibility of a plant to virus infections s_{IXY} . Since this susceptibility is believed to be related to the physiological age and activity, s_{IXY} is estimated by the proportion of leaves l_y younger than a physiological age threshold t_{ary} at infection time t_{IXY} of the cell $[X,Y]$. Different thresholds were assumed for different virus strains; a high value means that age-resistance arrives late. l_y is an input from the plant growth submodel.

s_{IXY} is used to calculate 1) the probability of an infected plant to become an infection source and 2) the fraction of tubers infected. The age resistance is assumed to affect virus multiplication and translocation in the plant, but not the infection process, contrary to the model of SIGVALD (1986), where the infection process is affected by age resistance.

$$s_{IXY} = l_y(t_{IXY}) \quad (6.22)$$

The latent period follows a normal distribution with mean m_l and standard deviation s_l (in physiological time units, Eq. 6.23). Since virus multiplication is highly dependent on plant physiology, the physiological time scale of the plant growth model is adopted. t_{IXY} is the physiological age, when a plant becomes an infection source, which has previously been infected at t_{IXY} :

$$t_{IXY} = t_{IXY} + N(m_l, s_l) \quad (6.23)$$

The disease state transitions of a plant are given by Eq. 6.24 (see also Fig. 4.1):

$$D_{XY}(k+1) = \begin{cases} \delta_l & \text{if } D_{XY}(k) = \delta_h \wedge \text{inoculation occurs at } k \wedge k \leq t_{hk} \\ \delta_p & \text{if } D_{XY}(k) = \delta_l \wedge T_p(k) \geq t_{IXY} \wedge u \leq s_{IXY} p_{il} \wedge k \leq t_{hk} \\ \delta_r & \text{if plant is removed at } k \\ D_{XY}(k) & \text{otherwise} \end{cases} \quad (6.24)$$

$$u = U[0, 1)$$

It is assumed that only a certain proportion (namely $s_{IXY} p_{il}$) of latently infected plants become infection sources and that all other will remain latently infected until harvest, i.e. their latent period is considered as infinite. The random number u is sampled only once for each plant in state δ_l , at the time of infection. $T_p(k)$ is the physiological time at k . After haulm-killing infection and outbreak stop.

Outputs of the Infection Submodel

The model assumes that all tubers of a secondarily infected plant (state δ_s) will be infected. The final proportion of infected tubers of primarily infected plants equals the susceptibility s_{IXY} , resp. age-resistance at the time of infection t_{IXY} . The proportion of infected tubers t_{XY} of cell $[X,Y]$ is:

$$t_{XY}(k) = \begin{cases} 1 & \text{if } D_{XY}(k) = \delta_s \\ s_{IXY} & \text{if } D_{XY}(k) \in \{\delta_l; \delta_p\} \wedge T_p(k) \geq t_{IXY} \wedge t_{IXY} \leq t_{hk} \\ 0 & \text{otherwise} \end{cases} \quad (6.25)$$

Tuber infection occurs simultaneously for that fraction of tubers that become infected at t_{IXY} (Fig. 6.4). No tuber infection occurs after haulm-killing in the model. The delay between plant infection (at t_{IXY}) and tuber infection (at t_{IXY}) is normally distributed with mean m_t and standard deviation s_t (in physiological time units):

$$t_{tXY} = t_{iXY} + N(m_t, s_t) \quad (6.26)$$

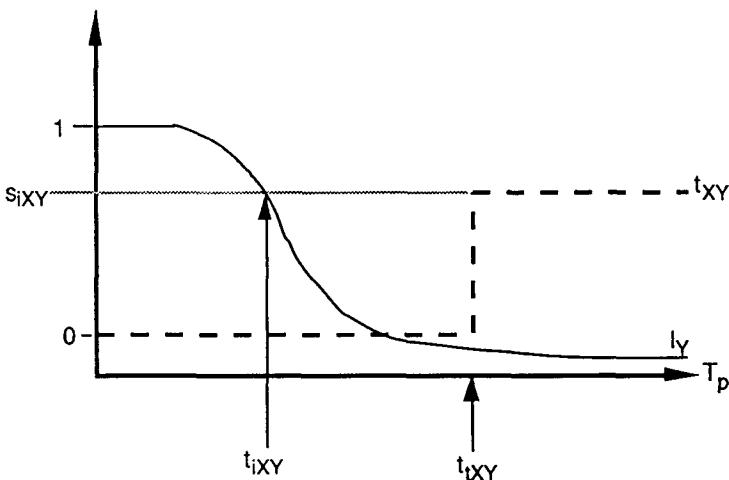


Fig. 6.4: Proportion of infected tubers (t_{XY} , broken line) of a plant infected at t_{iXY} . l_y = proportion of physiologically young leaves (unbroken line), t_{iXY} = time of tuber infection, T_p = physiological age.

The proportion of infected tubers p_t of the whole field is the average of all t_{XY} , except of removed plants:

$$p_t = \frac{\sum_{X=1}^{n_r} \sum_{Y=1}^{n_p} t_{XY}}{n_r n_p (1-p_r)} \quad (6.27)$$

The proportion of infected tubers is calculated regardless of the tuber initiation, thus p_t represents the potential tuber infection. The proportion of plants p_p having at least one infected tuber in the sample is:

$$p_p = \frac{\sum_{X=1}^{n_r} \sum_{Y=1}^{n_p} 1 - (1-t_{XY})^{n_t}}{n_r n_p (1-p_r)} \quad (6.28)$$

$1 - (1-t_{XY})^{n_t}$ is the binomial probability that at least one of the n_t tubers sampled of the plant $\{X, Y\}$ will be infected. t_{XY} is the fraction of tubers infected in this plant. This output variable is introduced for validation purposes.

The disease incidence p_d is a further output variable, which is calculated as the proportion of cells in state δ_l , δ_p , δ_s and δ_r . The proportion of sources p_s is the proportion of cells in state δ_p and δ_s .

6.3.3. PLANT GROWTH SUBMODEL

The plant growth submodel is an implementation of the potato model published by JOHNSON *et al.* (1986). The structure of the model, its coupling with the soil water balance model and the ModelWorks implementation is described in ROTH *et al.* (in press). Only the basic characteristics are summarized.

The submodel has six state variables, namely the dry matter [g/plant] of leaves L , stems S , roots R , tubers T and assimilate pool A and the physiological age T_p (calculated with the formula of SANDS *et al.*, 1979).

The physiological age T_{pi} of the leaves which are produced during day i and their dry weight L_i are retained for the calculation of senescence. This leaf state vector is also used for the calculation of the proportion of physiologically young leaves l_y :

$$l_y(k) = \frac{\sum_{i=t_{em}}^k L_i'(k)}{L(k)} \quad (6.29)$$

$$L_i'(k) = \begin{cases} L_i(k) & \text{if } T_{pi}(k) \leq t_{arv} \\ 0 & \text{otherwise} \end{cases} \quad (6.30)$$

t_{em} is the date of 50 % emergence of the crop.

6.3.4. WATER BALANCE SUBMODEL

This submodel, which is described in ROTH *et al.* (in press), was introduced to replace the measurements of soil water potential needed in the original model of JOHNSON *et al.* (1986). The submodel simulates the soil as a single compartment and has two state variables: the water content in the root zone W_r [mm] and the surface water storage W_s [mm]. The water stress factor w_s is calculated as the ratio of the actual daily transpiration t_a and the potential transpiration t_m .

6.4. Parameter Estimation

6.4.1. INOCULATION SUBMODEL

The relative mortality rate d_C (Tab. 6.2) was calculated (under the assumption of an exponential distribution for longevity) as the reciprocal of the mean longevity \bar{l}_C :

$$d_C = \frac{1}{\bar{l}_C} \quad (6.31)$$

The parameter e_G and r_{fG} were estimated by nonlinear regression, fitting the model

$$I = e_G e^{-r_{fG} T} \quad (6.32)$$

to the observed proportion of infected plants I (Fig. 6.5) after a fasting time T . Where more than one aphid per plant was used in transmission tests, the values were transformed with the formula of GIBBS & GOVER (1960).

The rate of interleaf movement f_{lmC} was estimated by the frequency of the sequence leaf-stem-leaf observed under controlled conditions, divided by the proportion of time

spent walking. The laboratory observations were used, because interleaf movements were relatively rare in the sequences and more observations were available under controlled conditions.

Tab. 6.2: Parameter values of the inoculation submodel.

Parameter	Value	Unit	Source
a_{ll}	0.069	m^2	DERRON (1983, unpublished data): area of the lowest leaves of each stem under the assumption of 10 stems per Bintje plant
d_{col}	0.09346	d^{-1}	Field and laboratory experiments 1989, Chapter 5.2
d_{ncol}	0.25	d^{-1}	Field and laboratory experiments 1989, Chapter 5.2
e_1	0.5086	%/100	data for <i>M. persicae</i> of KOSTIW (1975) (apterae and alatae), PRÖSELER & WEIDLING (1975), VAN HOOF (1980), KATIS & GIBSON (1985)
$e_{2a/b}$	0.1006	%/100	for <i>A. nasturtii</i> and <i>M. euphorbiae</i> of PRÖSELER & WEIDLING (1975), <i>A. nasturtii</i> of KOSTIW (1975) (apterae and alatae), <i>A. fabae</i> and <i>P. humuli</i> of VAN HOOF (1980)
e_3	0.01701	%/100	data for <i>R. padi</i> of KATIS & GIBSON (1985)
$t_{lm\ col}$	67.72	d^{-1}	Behaviour sequences under controlled conditions for <i>M. persicae</i> , 1989, Chapter 5.2
$t_{lm\ ncol}$	33.05	d^{-1}	Behaviour sequences under controlled conditions for <i>A. fabae</i> , 1989, Chapter 5.2
t_{lp}	2.0	mm/h	ROHITHA & PENMAN (1986)
t_{lt}	17	$^{\circ}\text{C}$	KRING (1972), threshold for <i>A. fabae</i>
t_{lw}	2.5	m/s	see Chapter 5.4, wind speed measured at 12 m height
p_{em}	0.05	%/100	Chapter 5.4, proportion of flight distances estimated to be longer than 50 m
p_{wp}	1/18	%/100	see text
r_1	10.57	d^{-1}	same source as e_1
$r_{2a/b}$	26.21	d^{-1}	same source as $e_{2a/b}$
r_3	4.5106	d^{-1}	same source as e_3
s_c	40	%/100	CARTER et al. (1982)
s_d	0.0237	m^2	TAYLOR & PALMER (1972), for density gradient of -1 and mean flight duration of 2 h
$s_{r\ col}$	0.2629	d^{-1}	Field experiments 1989, Chapter 5.2
$s_{r\ ncol}$	0.0	d^{-1}	assumption: no settling possible for noncolonizing aphids
t_{s1}	0.5517	%/100	FRITSCHE et al. (1972), data for <i>M. persicae</i>
$t_{s2a/b}, t_{s3}$	0.2803	%/100	FRITSCHE et al. (1972), data for <i>A. nasturtii</i>
t_{tb}	0.785	%/100	$\frac{\text{area of ellipse}}{\text{area of rectangle}} (= \frac{\pi}{4}, \text{ see Fig. 6.6})$
t_{tw}	0.367	%/100	$\frac{\text{circle area}}{\text{rectangle area}} (= \frac{\pi d_p^2}{4 d_r}, \text{ see Fig. 6.6}), \text{ value for: } d_r=2d_p$ (*)

The thresholds t_{lw} and t_{tb} were calculated as the ratio of the area of the vertical projection of the plant (Fig. 6.6, shaded) and the area $d_p d_r$ (rectangle).

The probability p_{wp} that an aphid moves on a neighbouring plant, provided it performs an interleaf movement, was estimated assuming that 1/3 of the leaves of each plant will be in contact with leaves of a neighbouring plant in case two neighbouring plants in a row touch each other. Assuming that aphids can walk in three dimensions with two directions each, the probability that an aphid will walk on a neighbouring plant during an interleaf movement is 1/18.

The serial inoculation probability t_{sG} was estimated on data from experiments, where aphids made a probe on an infection source plant and were subsequently transferred to healthy plants. After a first probe, they were transferred to the next healthy

(*) The parameter t_{tw} is calculated from the actual values of d_r and d_p .

plant. $v_i(n)$ is the proportion of aphids that infected n plants out of those aphids that transmitted virus at all. t_{SG} was estimated by nonlinear regression using the model:

$$v_i(n) = e^{-t_{SG}^*(n-1)} \quad (6.33)$$

where an auxiliary rate parameter t_{SG}^* was introduced. The serial inoculation probability t_{SG} was then calculated as:

$$t_{SG} = \frac{v_i(n+1)}{v_i(n)} = e^{-t_{SG}^*} \quad (6.34)$$

Data on serial transmission were available only for *M. persicae* and *A. nasturtii*. Since vectors having high e_G values like *M. persicae* seem to have also higher t_{SG} values, the value estimated for *A. nasturtii* was used for the vector groups 2a, 2b and 3, which have lower e_G -values than *M. persicae*.

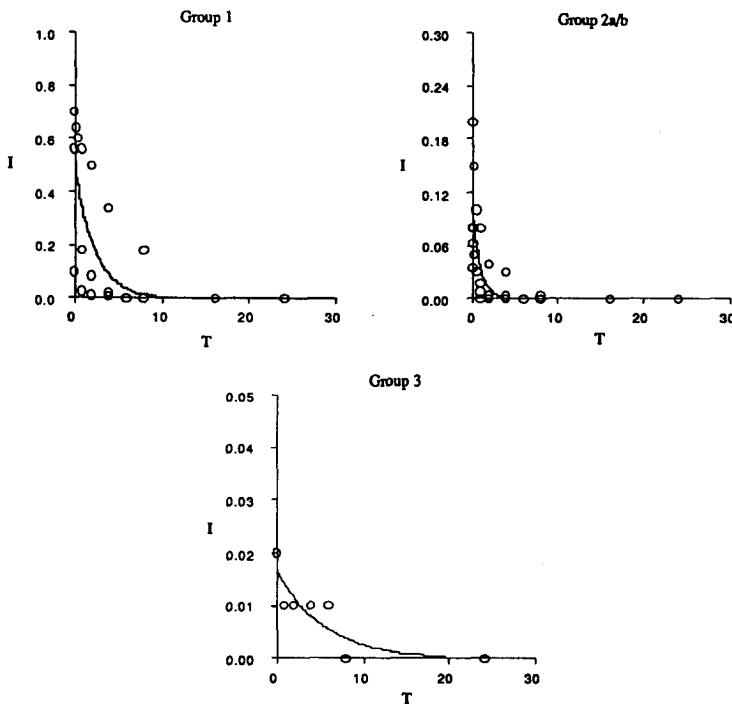


Fig. 6.5: Estimation of vector efficiencies (e_1 to e_3) and retention factors (r_1f_1 to r_3f_3) for vector groups 1 to 3 (see Tab. 6.1). I = proportion of infected plants, T = fasting time in hours between acquisition and inoculation. Note that the scales on the Y-axis are different.

The settling rate s_{RC} of colonizing species was estimated by fitting an exponential model by nonlinear regression:

$$p_{im}(a) = 1 - e^{-s_{RC}a} \quad (6.35)$$

where $p_{im}(a)$ is the proportion of colonizing aphids showing settling behaviour (behaviour type III) in the age class with mean age a . The behaviour sequences recorded in the field were used for this estimation.

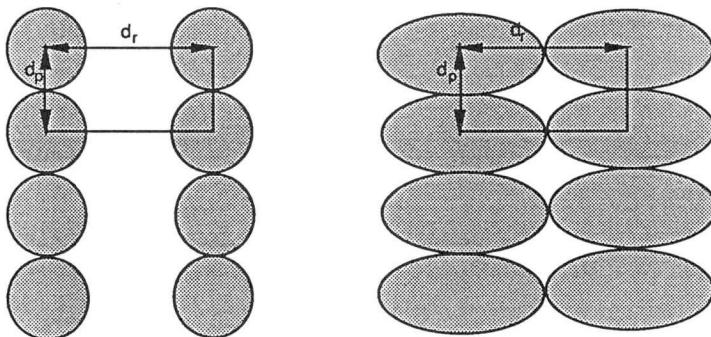


Fig. 6.6: Estimation of thresholds of touching of leaves within the row (t_{rw} , left) and between rows (t_{rb} , right).

6.4.2. INFECTION SUBMODEL

Tab. 6.3: Parameter values of the infection submodel.

Parameter	Value	Unit	Source
α	0.657	-	Field experiments 1989, see Chapter 5.4
β	9.613	m	Field experiments 1989, see Chapter 5.4
d_w	0.81	m	THYGESEN (1968), mean walking distance in first experimental year
κ	0.345	-	Field experiments 1989 (Chapter 5.4)
m_l	233.2	DD	GUGERLI (1985-87, unpublished); artificial inoculation experiment
m_t	63.2	DD	CAMBREZY <i>et al.</i> (1981)
p_{il}	0.692	%/100	GUGERLI (1985-87, unpublished); artificial inoculation experiments
s_l	90.0	DD	GUGERLI (1985-87, unpublished); artificial inoculation experiment
s_t	26.4	DD	CAMBREZY <i>et al.</i> (1981)
$l_{ar} PVYn$	103.6	DD	GUGERLI (1985, unpublished data); artificial inoculation experiment
$l_{ar} PVYo$	95.6	DD	GUGERLI (1986, unpublished data); artificial inoculation experiment

The concentration parameter κ of the von Mises distribution of flight directions was derived from the correlation flight direction - wind direction (Tab. 6.3). The κ -value corresponding to the correlation coefficient r ($r = 0.17$, see chapter 5.3) was taken from a table given by BATSCHELET (1981).

The 'age resistance thresholds' l_{arV} were estimated by identification with the coupled plant growth and soil water submodels (ROTH *et al.*, in press), fitting the curve of the proportion of young leaves l_y to the observed proportion of infected tubers, obtained at harvest from plants inoculated at day k (Fig. 6.7). The identification was performed by minimizing the sum of squares of the deviations between observed and simulated values with a halving doubling algorithm (previously used in the work of ROTH *et al.*, in press).

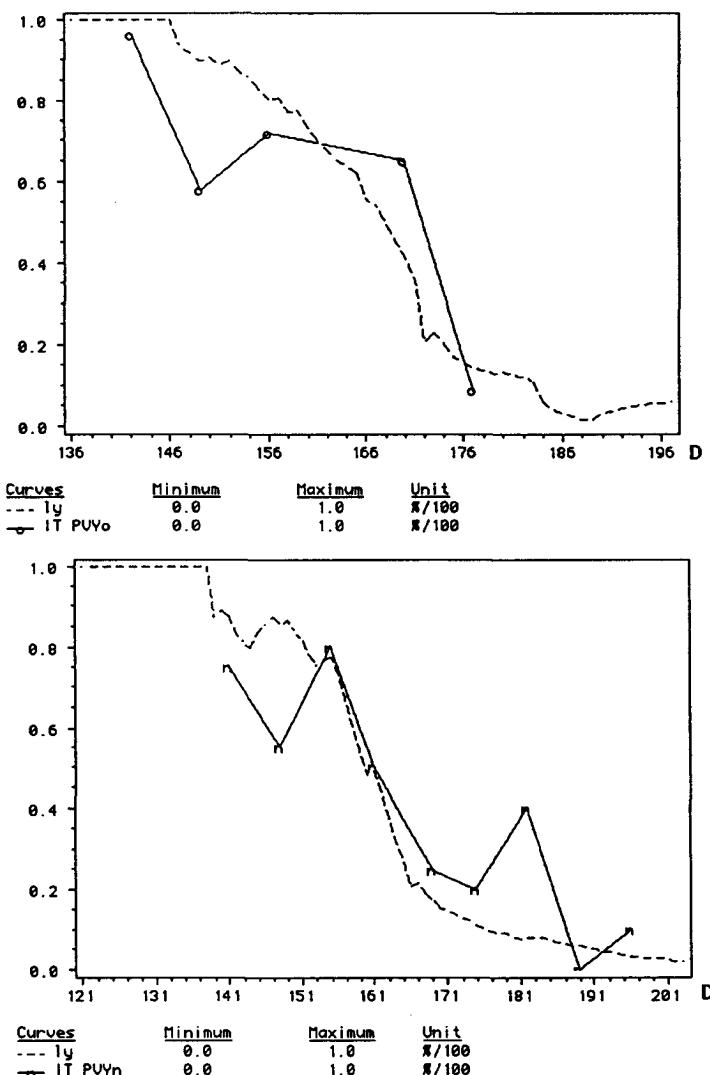


Fig. 6.7: Fit of the proportion of tubers infected by PVY^o (upper graph) resp. PVYⁿ (lower graph) by the proportion of leaf tissue l_y younger than the physiological age thresholds $t_{ar}\ PVY_o$ resp. $t_{ar}\ PVY_n$. The fit was performed by varying the respective age resistance threshold in order to minimize the least squares of the deviations in a parameter identification experiment. D = julian day (days after first Jan), $IT\ PVY_n$, $IT\ PVY_o$ = proportion of infected tubers harvested after haulm-killing from plants inoculated at day D . Data from artificial inoculation experiment (PVY^o: 1986, PVYⁿ: 1985, GUGERLI, unpublished data).

6.4.3. PLANT GROWTH AND WATER BALANCE SUBMODELS

Tab. 6.4: Parameters of the plant growth submodel. The parameter names are adopted from ROTH *et al.* (in press). The identifiers are used in the listings of Appendix IV.

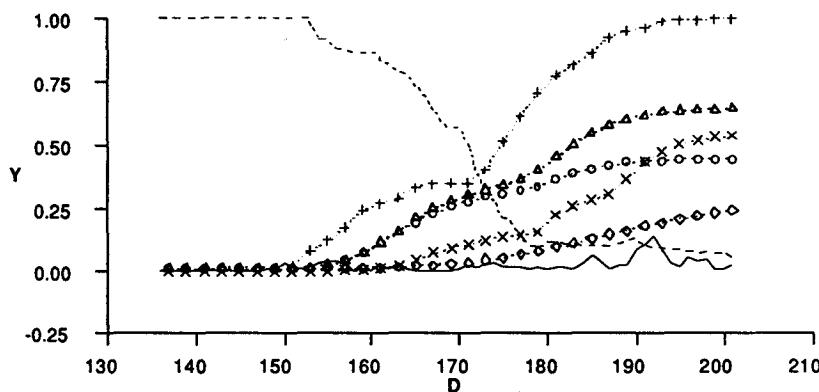
Parameter name	Identifier	Meaning	Unit	Value
K	kGrowth	Plant growth parameter	-	60
μ_L	kLeaf	Partitioning parameter for leaves	-	1
μ_S	kStem	Partitioning parameter for stems	-	15
μ_R	kRoot	Partitioning parameter for roots	-	1
μ_T	kTuber	Partitioning parameter for tubers	-	12
$maxPlantPA$	maxPlantPAge	Maximum physiologic-age	DD	1000
$tuberInitPA$	tuberInitPAge	Physiologic-age at tuber initiation	DD	120
$maxLeafPA$	maxLeafPAge	Maximum physiologic-age of leaves	DD	600
$lpAge1$	lpAge1	maximal leaf age with optimal photosynthesis	DD	150
$lpAge2$	lpAge2	minimum leaf age with optimal photosynthesis	DD	75
$pASeedCont$	pASeedCont	duration of assimilate contribution from mother tuber	DD	120
$propBefE$	propBefE	proportion of assimilates of the mother tuber supplied before crop emergence	%/100	0.2
$propAftE$	propAftE	proportion of assimilates of the mother tuber supplied after crop emergence	%/100	0.65
$propInLeaf$	propInLeaf	proportion of mother tuber assimilates partitioned into leaves	%/100	0.4
$propInStem$	propInStem	proportion of mother tuber assimilates partitioned into stems	%/100	0.4
$propInRoot$	propInRoot	proportion of mother tuber assimilates partitioned into roots	%/100	0.2
$maxPotNetGR$	maxPotNetGR	maximum potential net growth rate	g/MJ	2.2
$minGrT$	minGrowthTmp	minimal growth temperature (development threshold)	°C	4
$optGrT$	optGrowthTmp	optimal growth temperature	°C	20
$maxGrT$	maxGrowthTmp	maximal growth temperature	°C	30

Parameters of the plant growth submodels were adapted for the variety Bintje. The values that have been changed are listed in Tab. 6.4.

7. SIMULATION STUDIES OF THE EPOVIR-MODEL

7.1. Analysis of Model Behaviour

7.1.1. STANDARD RUN



Legend	Simulated time series	Observed time series	Minimum of Y-axis	Maximum of Y-axis	Unit
Proportion of source plants (p_S)	◇		-0.25	1.0	%/100
Fraction of tubers infected (p_I)	○		-0.25	1.0	%/100
Proportion of plants with infected tubers (p_P)	△		-0.25	1.0	%/100
Disease incidence p_d (secondary infection sources, primary infection sources and latent plants)	+		-0.25	1.0	%/100
Number of vectors captured in suction trap		—	-125	500	#/d
Susceptibility of the potato plants to infections (S_{iXY})	- - - - -		-0.25	1.0	%/100
Tuber dry mass	x		-50	200	g/plant

Fig. 7.1: Standard run of the EPOVIR model (averages of 50 simulation runs). D = julian day (days after first Jan.).

In addition to those listed in Tab. 6.2 to 6.4, the standard run was performed with the following parameter values:

$$\begin{aligned}
 p_i &= 0.01 \\
 n_r &= 40 \\
 n_p &= 80 \\
 v_{im} &= 0.0025
 \end{aligned}$$

The input data and all other parameter values that are different between the data frames were taken from data frame 1983n1. The simulation of a single standard run required about 1.5 minutes on an Apple™ Macintosh™ IIfx computer and about 200000 random variates. The cycle length of the random number generator of about $2.78 \cdot 10^{13}$ is sufficiently long to ensure that the random number sequence does not repeat. The results of 50 simulation runs (see Chapter 7.1.2.2) are averaged in the standard run.

The parameter v_{im} was set to 0.0025, assuming that 1/4 of the area surrounding the simulated field is occupied by potatoes with the same initial disease incidence. This parameter was kept constant during the simulation, i.e. the effect of the primary sources in the neighbourhood of the field was ignored.

Fig. 7.1 shows the results of the standard run of the EPOVIR model. Virus spread starts with the occurrence of the first aphids. On average all plants become infected until harvest for the initial disease incidence of 1 %. Due to the age resistance, which interferes with tuber infection, the latter is much lower than the disease incidence.

7.1.2. SIMULATION TECHNICAL ASPECTS

7.1.2.1. *Effect of Torus and Selection of Representative Field Size*

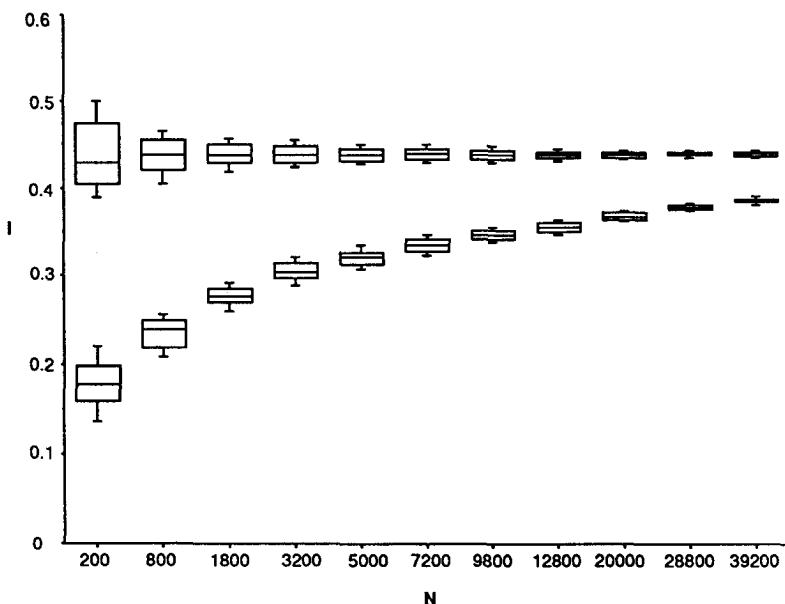


Fig. 7.2: Proportion I of infected tubers at harvest in function of the number of plants in the simulated field. Lower box-plots: finite field, upper box-plots: torus. The box plots represent the 10th, 25th, 50th, 75th and 90th percentile of 100 simulation runs.

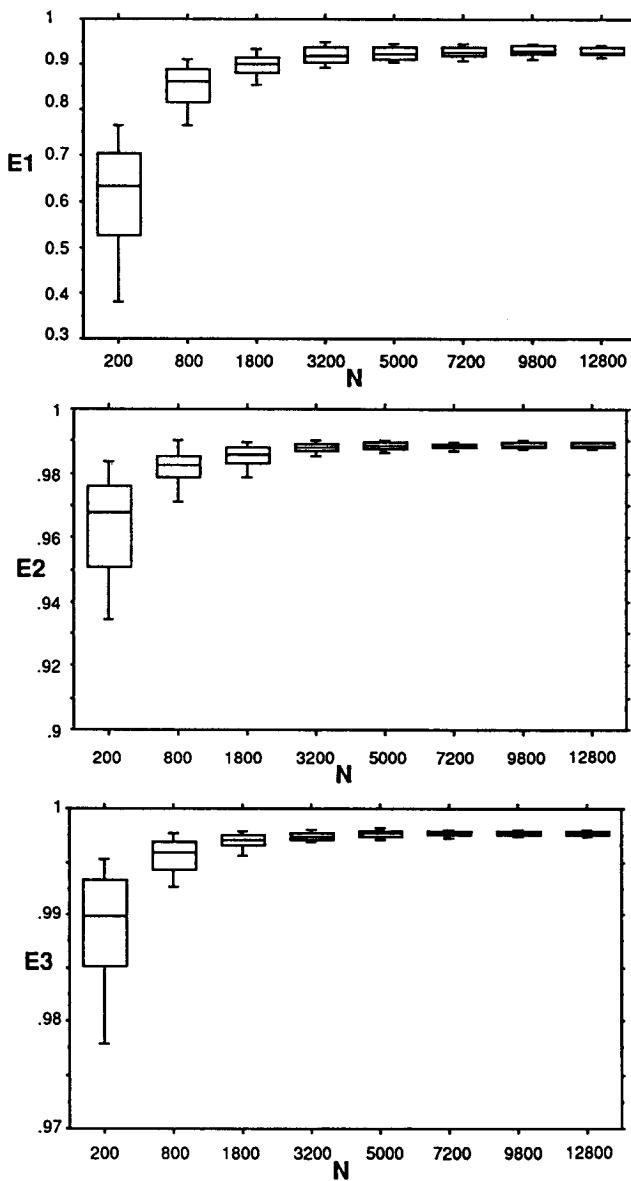


Fig. 7.3: Effect of the number of plants N in the simulated torus, on the evenness $E1$ to $E3$ of the distance class distribution at t_1 to t_3 . The box plots represent the 10th, 25th, 50th, 75th and 90th percentile of 100 simulation runs.

To quantify the effect of the torus and to select a representative number of cells, simulation experiments with different cell numbers were performed. The respective values were (rows x plants): 10x20=200, 20x40=800, 30x60=1800, 40x80=3200, 50x100=5000, 60x120=7200, 70x140=9800, 80x160=12800, 100x200=20000 and 140x280=39200 plants. For the plant density used in the standard run, the last experiment corresponds to an average seed potato field under Swiss conditions (about 1 ha). Since the variance of the output variables is larger for the small cell numbers, the number of simulation runs was increased to 100 per experiment. Thus 100 simulation runs were performed with an infinite field (torus) and 100 runs with a finite field.

In a finite field all inoculations falling outside the field area do not lead to infections. This border effect results in lower disease severities, measured as the proportion of infected tubers (Fig. 7.2). The border effect is highest, if the number of cells is small, and decreases with increasing number of cells, but apparently very slowly. To yield similar results for finite and infinite fields, the number of plants should have been increased far beyond the numbers used in these experiments. The mean disease severity was not affected by the field size, if a torus was assumed, only the variance decreased with increasing cell number. It is thus possible to simulate a large field with a relatively small number of cells, which saves computer memory and computation time. A torus was used in all subsequent simulations, except the validation experiments.

The number of plants for subsequent simulation experiments was selected using following criteria: the number of plants should be large enough to allow the simulation of situations with a small initial disease incidence, the length and width of the area should exceed the mean flight distance and the distribution of the evenness values should not change with increasing field size. For technical and efficiency reasons the evenness of the distance class distribution was calculated only in the experiments up to 12800 cells. Fig. 7.3 suggests that the distribution of the evenness values remained relatively constant above 3200 plants. This field size seems also to fulfil the other criteria, since its width and length would be about three times longer than the mean flight distance and the cell number would be sufficiently high to allow for low initial disease incidences. A size of 40x80 plants was used in all simulation experiments except model validation experiments.

The evenness values showed an increasing tendency with increasing field size (Fig. 7.3), probably due to the small number of plants in the sample for field sizes 200, 800 and 1800, producing an uneven pattern, which does not actually signify clumping. With increasing field size, the distance class distribution became more and more smooth resulting in converging evenness values.

7.1.2.2. Selection of Representative Size of Stochastic Sample

The stochastic sample size was chosen based on the distribution of the proportion of infected tubers p_t . Normal probability plots of p_t (Fig. 7.4) suggest that the values were nearly normally distributed. The confidence intervals for the mean were estimated according to SACHS (1984) by:

$$cl_{wll} = \bar{x} \pm \frac{t_{n-1; \alpha/2} s}{\sqrt{n}} \quad (7.1)$$

where $cl_{u/l}$ is the upper, resp. lower confidence limit, \bar{x} the average, s the standard deviation, n the sample size, $t_{n-1; \alpha}$ the value of the t-distribution for error probability α .

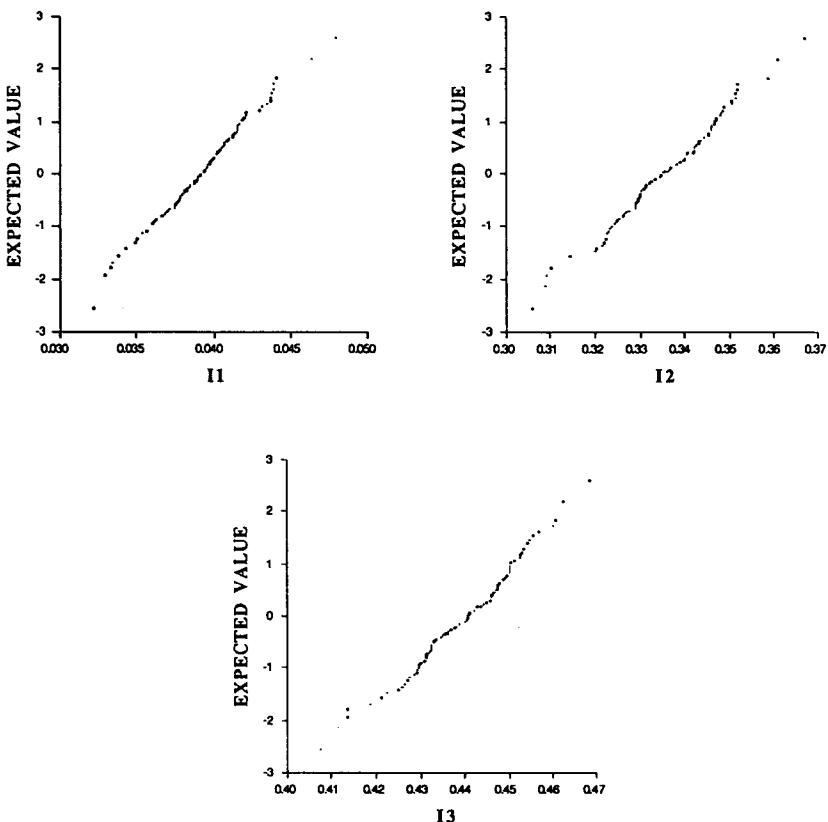


Fig. 7.4: Normal probability plots of the fraction of tubers infected I_1 to I_3 at t_1 to t_3 of 100 simulation runs.

The sample size was chosen so that the relative half confidence interval ($= \frac{|cl_{u/l} - \bar{x}|}{\bar{x}}$) was below 1 % for $\alpha = 0.05$, i.e. the relative deviation of the estimated

mean from the population mean should not exceed 1 % in 95% of the cases. The sample size of 50 simulation runs was found to satisfy this condition at t_2 and t_3 , but not at t_1 . To fulfil the condition also at t_1 , a sample size of over 200 runs would be required. The relative half confidence intervals for 50 runs decreased with simulation time (1.9 % at t_1 , 1.0 % at t_2 and 0.76% at t_3). Since the final disease severity was of main interest, 50 runs were considered as sufficiently representative and were used for the subsequent

simulation experiments, unless stated otherwise. The evenness values were not averaged over the 50 runs, but calculated on the cumulated statistics (see Chapter 2.4.2).

7.1.3. RELATIONSHIP BETWEEN EPIDEMIC PROGRESS AND PREDICTOR VARIABLES

To gain a first insight into the correlation of variables with virus dissemination and to investigate, which variables could potentially be used as single predictors, the increase of disease incidence was related after logit-transformation by simple linear regression to several predictor variables for three data sets:

Data set I: disease incidence p_d simulated by EPOVIR for the data frames 1983n1, 1985n1, 1986n1, 1987n1, 1988o1, 1989o1 and 1990o2 ($p_i = 0.01$ in all simulations, averages of 30 runs).

Data set II: subset of I for the data frames 1985n1, 1986n1, 1987n1, 1988o1 and 1990o2 (for comparison with III).

Data set III: validation data of the same data frames as in II. Since the disease incidence p_d was not available, the proportion of plants with infected tubers p_p was used instead. Only the initial and the final value of p_p was known.

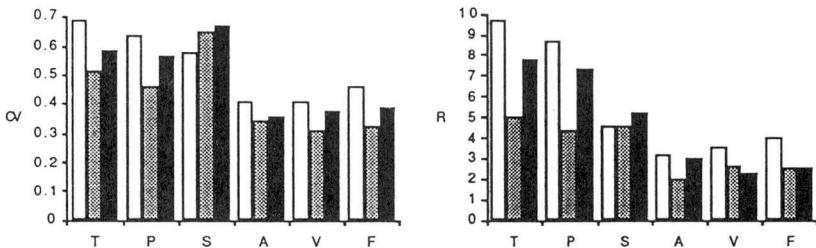


Fig. 7.5: Coefficient of variation CV and ratio $R = \frac{r_{\max}}{r_{\min}}$ (r_{\max}, r_{\min} = maximal resp. minimal regression coefficient) for the regression coefficients of the logit-transformed disease progress curves. Predictor variables in the regression model: T = days after emergence, P = physiological time after emergence, S = cumulated number of aphids caught in the suction trap, A = cumulated number of vectors (Tab. 6.1) caught in the suction trap, V = cumulated number of vectors caught in the suction trap weighted by vector efficiency and F = cumulated number of vectors in the field simulated by EPOVIR weighted by vector efficiency. White bars = data set I, grey bars = II, black bars = III.

$$A = \sum_{k=t_{em}}^t \sum_G s_G, \quad V = \sum_{k=t_{em}}^t \sum_G (s_G e_G), \quad F = \sum_{k=t_{em}}^t \sum_G (N_G e_G)$$

p_d and p_p were logit-transformed, which would linearize a logistic disease progress curve, before estimating the linear regression coefficients. In data sets I and II, the regression line was forced through $\text{logit}(p_i)$. Since only two data points were measured in data set III, forcing through the first point was not necessary (the regression line would

pass through this point). The coefficient of variation CV and the ratio R were calculated (see Fig. 7.5).

The time after emergence T could explain only little of the variation, the physiological time P slightly more. The number of aphids caught in the suction trap S reduced the coefficient of variation slightly in data set I, but not in II and III and the ratio R considerably. The variables A , V and F were the best predictors. Accounting for vector efficiency (in V and F) hardly improved the prediction, compared with A . An analysis showed that A and V were strongly correlated ($r = 0.97$).

7.1.4. STRUCTURAL ANALYSIS

The effect of single processes and of ignoring them in the model was studied in the structural analysis.

7.1.4.1. Nonrandomness of Spatial Pattern

Assuming a random distribution of inoculations overestimated the disease severity by 15-20 % compared with the standard run (Fig. 7.6, B vs A). The evenness values were very close to 1 in experiment B (virus spread random in space), which confirms the results of Chapter 4 that the evenness can discriminate between random and nonrandom spatial patterns. In experiment C the flight directions were randomly distributed over all directions. Neither the disease severity nor the spatial pattern were visibly affected, probably because the von Mises distribution for the value of $\kappa = 0.345$ used in the standard run is close to a circular uniform distribution.

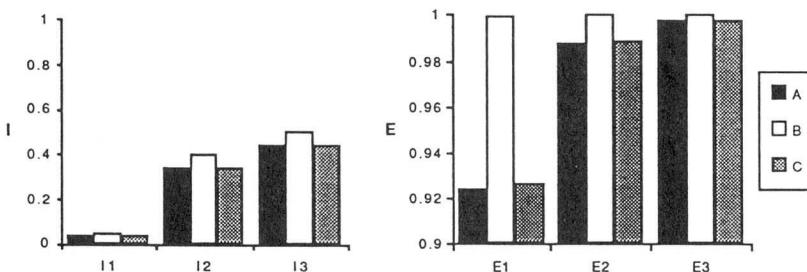


Fig. 7.6: Effect of the nonrandom spatial distribution of the inoculations. A: standard run, B: all inoculations uniformly distributed, C: $\kappa = 0$ (uniform distribution of flight directions). I1 to I3: proportion of infected tubers, E1 to E3: evenness of distance class distribution at t_1 to t_3 .

The overestimation of disease severity by assumption of a random spatial pattern was more marked for low initial disease incidences (Fig. 7.7, see also Fig. 4.4). This must be attributed to a 'saturation effect' of the epidemic, i.e. the proportion of infected tubers was close to 1. The difference reached 25 % at t_3 for $p_i = 0.001$. The ratio RI in-

creased during the season for $p_i = 0.001$. For $p_i = 0.01$ RI remained virtually constant and for $p_i = 0.1$ it nearly disappeared at t_2 and t_3 .

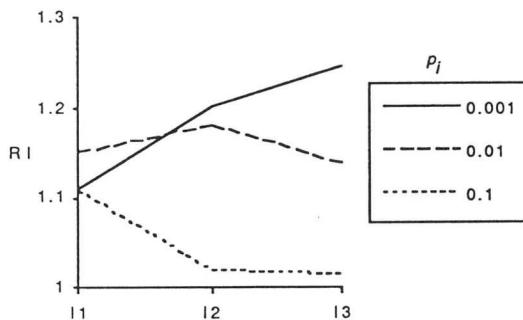


Fig. 7.7: Effect of the initial disease incidence p_i on the ratio of infected tubers $RI = \frac{I_u}{I_s}$. I_u = proportion of infected tubers in experiment with a uniform distribution of all inoculations, I_s = proportion of infected tubers in standard run. t_1 to t_3 = proportion of infected tubers at t_1 to t_3 .

7.1.4.2. Vector Abundance

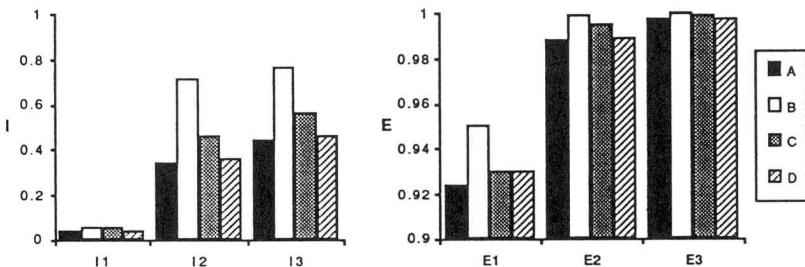


Fig. 7.8: Effect of vector abundance parameters. A: standard run, B: $p_{em} = 0$ (no emigration), C: $d_C = 0$ (no mortality), D: $s_{rC} = 0$ (no settling).

Inhibiting alatae emigration (Fig. 7.8, B) doubled the disease severity in the simulation model. Eliminating the mortality increased the fraction of tubers infected by 28–38 %, whereas the exclusion of settling had only a minor effect on the epidemic progress.

7.1.4.3. Behaviour Characteristics

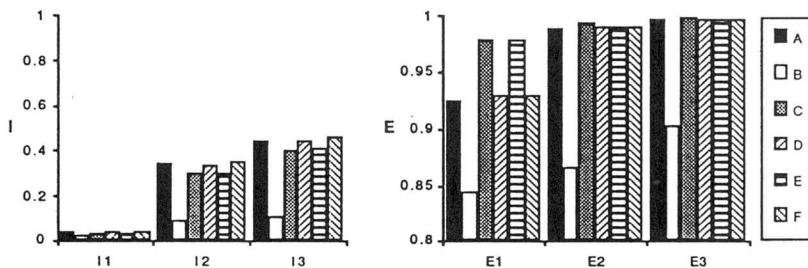


Fig. 7.9: Effect of the different types of inoculations and behaviour sequence parameters:
A: standard run, B: $i_f = 0$ (no inoculations caused by flying aphids), C: $i_{ws} = 0$ and $i_{wl} = 0$ (no inoculations caused by walking aphids), D: $i_{wl} = 0$ (no inoculations by walking over leaf bridges), E: $i_{ws} = 0$ (no inoculations by walking on soil), F: $p_{piC} = 1$ (probing frequency not limiting, p_{piC} = probability to probe on the source plant).

If no virus transmission would occur by flying vectors, tuber infection would be drastically lowered (Fig. 7.9, B), indicating that flying aphids contributed most to the infections in the model. The evenness values were also considerably lower. This more marked clumping was due to the fact that the flight distances were much longer than the walking distances. The elimination of inoculations by walking (C) decreased the disease severity only slightly, but reduced disease clumping considerably. Walking on soil (see E) contributed significantly more to virus infections than walking over leaf bridges (see D). Neither of the output variables was significantly changed by setting the probing probability p_{piC} to 1.

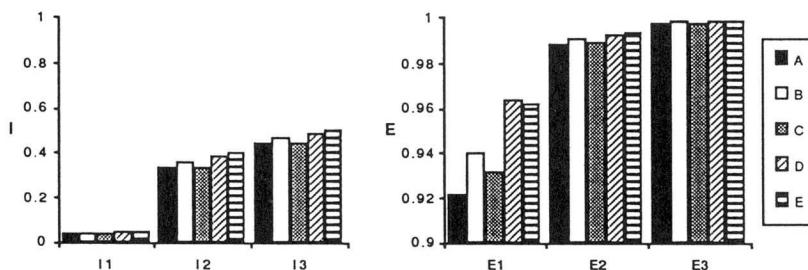


Fig. 7.10: Effect of the thresholds for take-off: A: standard run (take-off limited by wind speed, precipitation and temperature), B: take-off not limited by wind speed, C: take-off not limited by precipitation, D: take-off not limited by temperature, E: take-off possible during the whole day (inhibited only during night), E is a combination of B, C and D.

The temperature threshold was the most restricting one, followed by the wind speed threshold (Fig. 7.10), whereas the precipitation threshold did not affect the epidemic progress at all. The precipitation threshold (2 mm/h) was exceeded only during few hours and usually also the wind speed or the temperature inhibited flight during this periods. The assumption of suitable conditions for take-off during the whole day (E) would overestimate disease severity by 15-35 %.

7.1.4.4. Vector Propensity

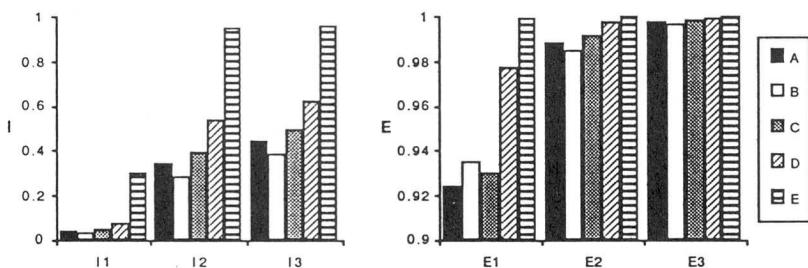


Fig. 7.11: Effect of vector propensity parameters. A: standard run, B: $t_{SG} = 0$ (no serial inoculations), C: $r_{fG} = 0$ (no loss of infectivity during the retention period), D: $v_{im} = 0.1$ (10 % of immigrating aphids come directly from infection sources), E: D: $v_{im} = 1$ (100 % of immigrating aphids come directly from infection sources).

Omitting serial inoculations (Fig. 7.11, B) would underestimate disease severity, whereas omitting retention (C) would slightly overestimate it. Increasing the fraction v_{im} of immigrating aphids coming from source plants to 0.1 increased the final disease severity by 40 %, $v_{im} = 1$ led to 115 % more tuber infection at the haulm-killing date (at t_1 the disease severity was even 7.4 times higher in the latter experiment, the smaller difference at t_3 was due to the saturation effect).

7.1.4.5. Plant-Virus Interaction

Age resistance (Fig. 7.12, B) is apparently very important in reducing tuber infection. The susceptibility of the plants to virus infections s_i is already low, when the large mass of the vectors arrives (Fig. 7.1). Tuber infection was more affected by age resistance at the end of the growing season than during its early phases.

The variance of the latent period played an important role, since it accelerated epidemic progress, compared with a constant latent period (C). The elimination of the latent period (D) increased the disease severity dramatically. The latent period thus considerably slowed down the disease spread.

Primary infection sources contributed also to disease spread (E). Removal of secondary sources by early roguing would even increase the importance of primary sources. Presence of primary sources led also to higher evenness values, due to the fact

that only the secondary sources were used as origins for the distance-class statistics (see Chapter 2.4.2 & Chapter 4.3). The delay to tuber infection (F and G) was important for early tuber infections, but did not affect the final result.

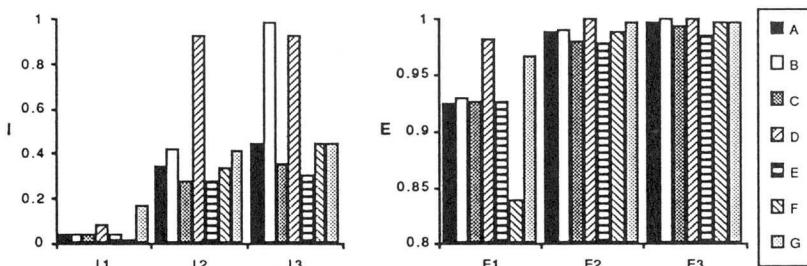


Fig. 7.12: Effect of plant-virus interaction parameters: A: standard run, B: $s_{IXY} = 1$ (no age resistance), C: $s_l = 0$ (constant latent period), D: $m_l = 0$ and $s_l = 0$ (no latent period, i.e. immediate outbreak), E: $m_l = \infty$ (no primary infection sources), F: $s_l = 0$ (constant tuber infection delay), G: $m_l = 0$ and $s_l = 0$ (no tuber infection delay, i.e. immediate tuber infection).

7.2. Validation

7.2.1. VALIDATION EXPERIMENTS

The goal of the validation experiments was to simulate as realistically as possible the infections in a (usually small) experimental plot and not to mimic a large field as in the other simulations. Using a torus in these cases could lead to artefact results, because it would e.g. be possible in the model that an aphid acquires PVY in the experimental plot, takes off and is carried downwind and finally lands again in the experimental plot, upwind from the infection source. Hence it was preferred to simulate a sufficiently large margin area of width d_m (Fig. 7.13, B) around the experimental plot (Fig. 7.13, A) to avoid the border effect. d_m was increased in simulation experiments until the disease severity in plot A remained constant. $d_m = 20$ m was found to be sufficiently large. Only the area of the experimental plot (A) was used for validation. The cellular automaton was initialized differently from the standard run: the positions of secondarily infected, healthy and lacking plants were used to initialize the experimental plot (A). In the margin area (B) the initial disease incidence was set to p_{ib} (Tab. 2.3).

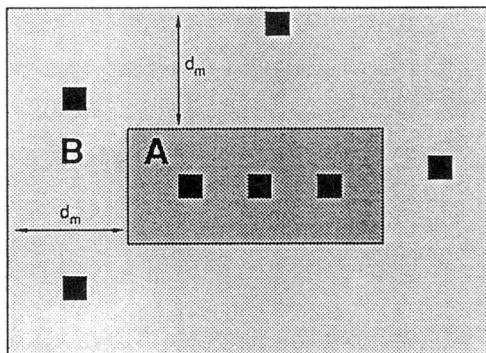
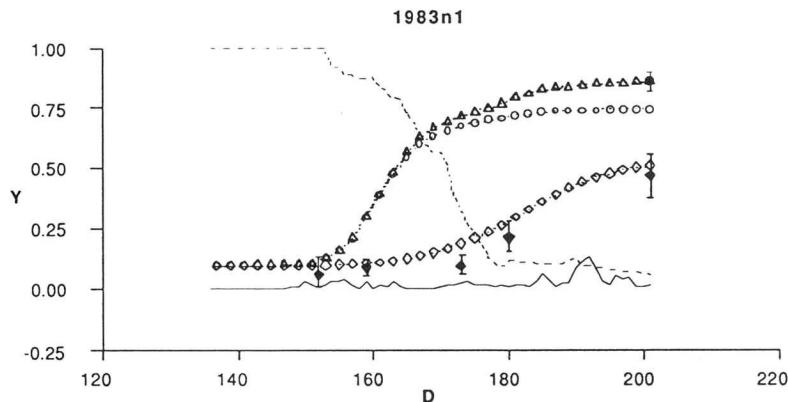


Fig. 7.13: Initialization of simulation runs in validation experiments. Area A is the experimental plot with known positions of infection sources (black boxes). Only this area was used for validation. Area B is the margin area of width d_m , with a proportion p_{ib} of randomly distributed infection sources. Shaded areas symbolize initially healthy plants.

7.2.2. VALIDATION OF TEMPORAL BEHAVIOUR



Legend	Simulated time series	Observed time series	Minimum of Y-axis	Maximum of Y-axis	Unit
Proportion of source plants (p_s)	◇	◆	-0.25	1.0	%/100
Fraction of tubers infected (p_i)	○	●	-0.25	1.0	%/100
Proportion of plants with infected tubers (p_p)	△	▲	-0.25	1.0	%/100
Number of vectors captured in suction trap		—	-125	500	# / d
Susceptibility of the potato plants to infections (s_{iXY})	- - - - -		-0.25	1.0	%/100

Fig. 7.14: Validation of temporal behaviour of the EPOVIR-model on eleven experimental data sets collected at the Swiss Federal Agricultural Research Station of Changins (continued below).

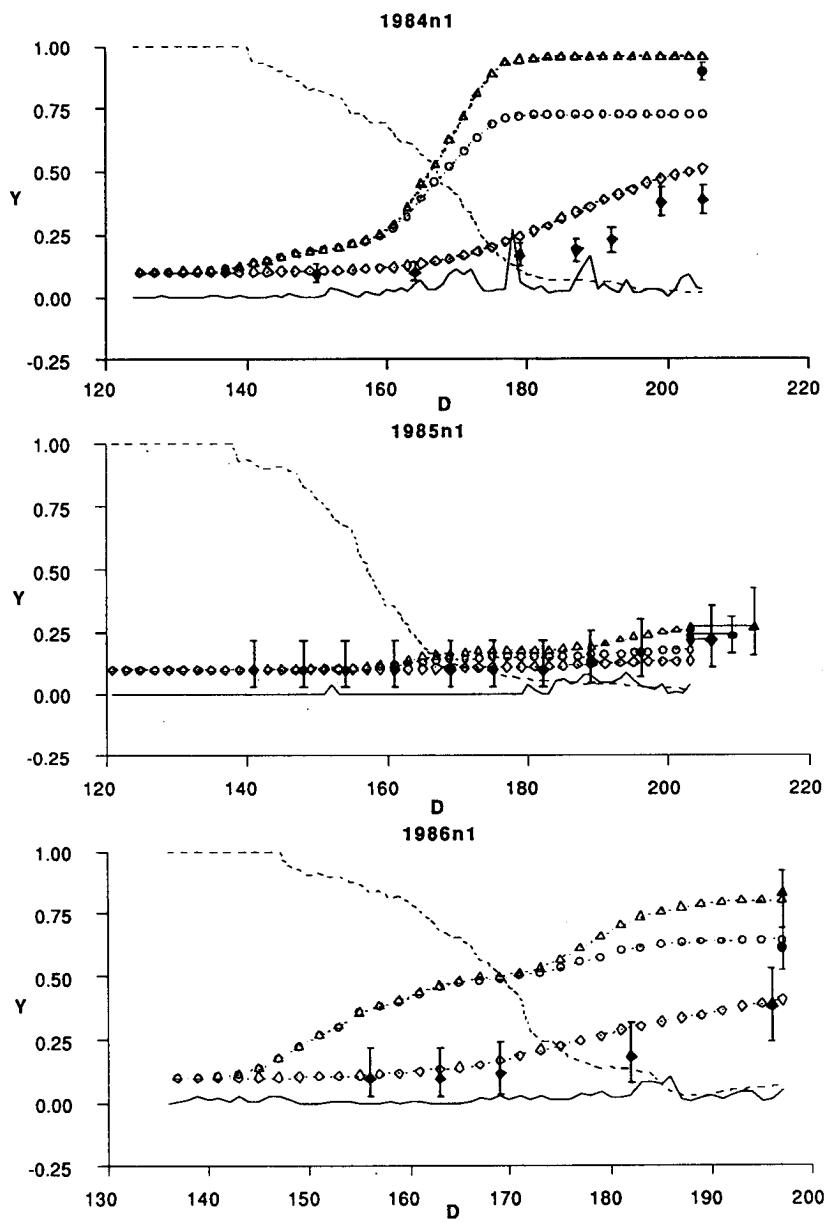
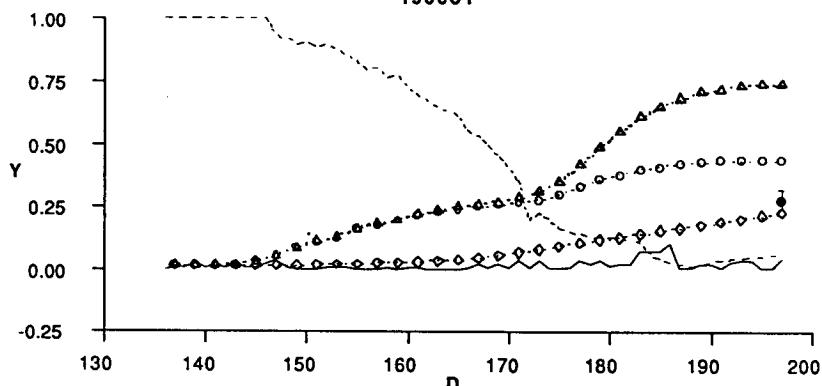


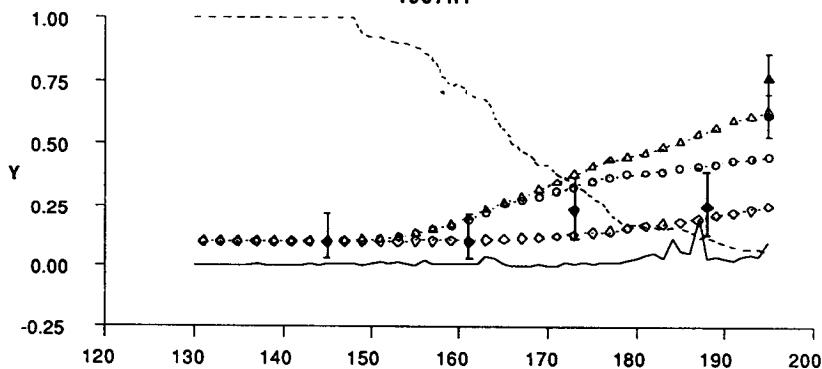
Fig. 7.14 continued.

108

1986o1



1987n1



1987o1

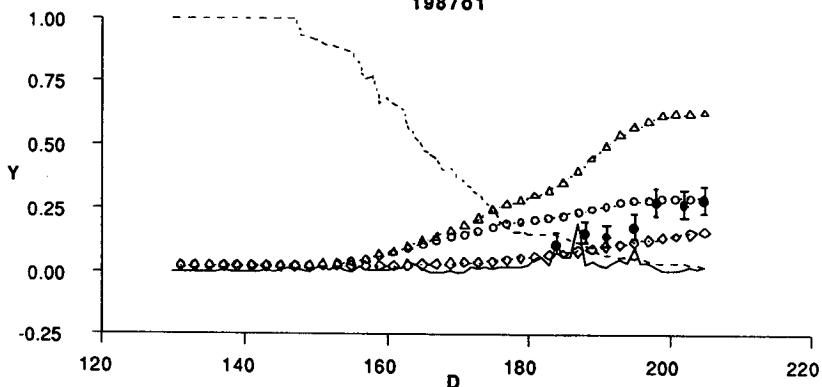


Fig. 7.14 continued.

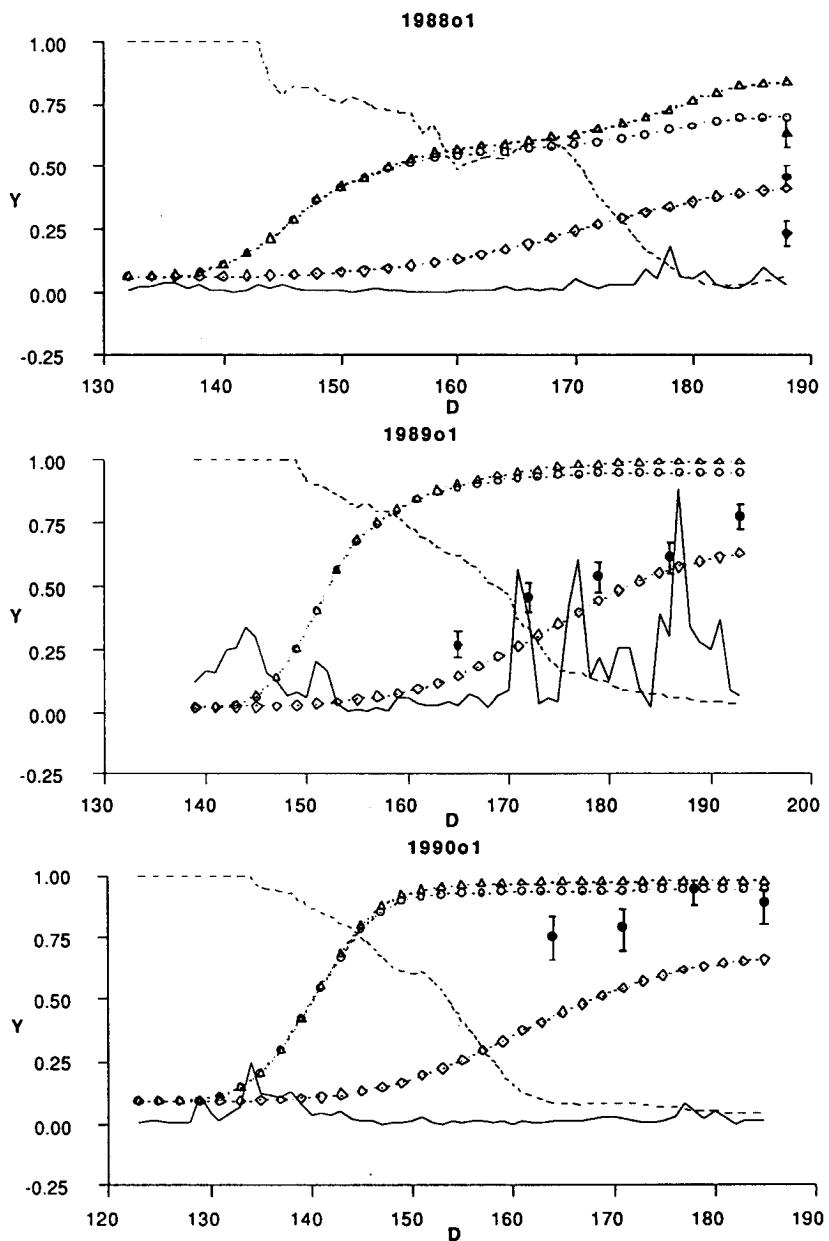


Fig. 7.14 continued.

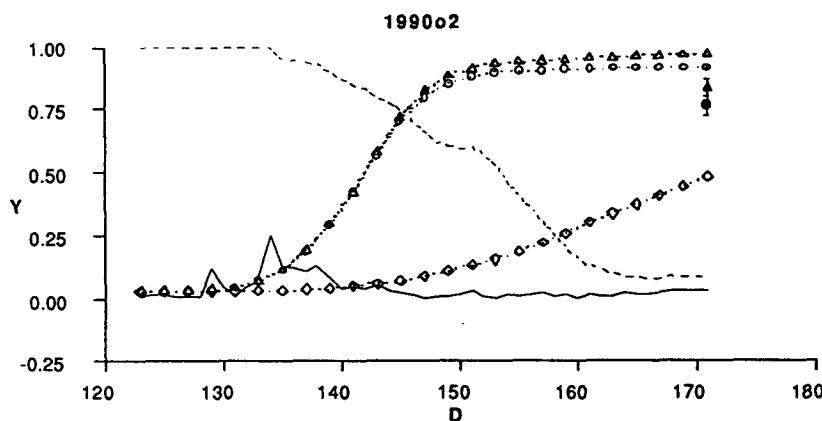


Fig. 7.14 end.

The data sets used for the validation were described in Chapter 2.7.2. The proportion of stems and leaves showing a positive reaction in ELISA was used for the validation of the proportion of source plants p_s (secondary and primary sources), assuming that positive reaction in ELISA corresponds to transmissibility of the viruses by the vectors.

In many cases the temporal behaviour of the model seems to be close to the observed data, but in some situations the model outputs largely deviated from the experimental data (Fig. 7.14). Tuber infection p_t was underestimated in experiments 1983n1, 1984n1, 1987n1 and overestimated in 1986o1, 1988o1, 1989o1, 1990o1 and 1990o2. In general all three variables p_t , p_p and p_s were either overestimated or underestimated, except in data frame 1984n1, where p_t and p_p were underestimated, whereas p_s was slightly overestimated. The ratios of these output variables seem to be close to that of the field data.

7.2.3. VALIDATION OF THE SPATIAL PATTERN

Two experimental field situations could be used to validate the distance class distribution (Fig. 7.15). In these two situations, the secondary infection sources were randomly distributed. In all other experiments the distance class distribution statistics could not be applied, since the secondary sources were regularly distributed. The analysis of such data would yield a periodic distance class distribution.

In 1988 the agreement between simulated and observed distance class distribution was good, except that the model underestimated the disease incidence in the distance classes close to the infection sources. In 1990 the simulated distribution was much more even than the observed one. These deviations can partly be explained by the overestimation of the disease incidence in the two experiments (see Fig. 7.14, 1988o1 and 1990o2). Despite the standardization of the f_c -values, the distance class distribution is not independent of the overall disease incidence.

To make use of the other data sets, the proportion of infected tubers was validated for different positions relative to the infection source. The sampling plan and the labeling

of the positions is schematized in Fig. 7.16. In some experiments, the distance from the source and the direction were known (position P_n and P_{-n} resp. row R_n and R_{-n}), in other cases, only the distance from the source was known (summarized under P_n resp. R_n).

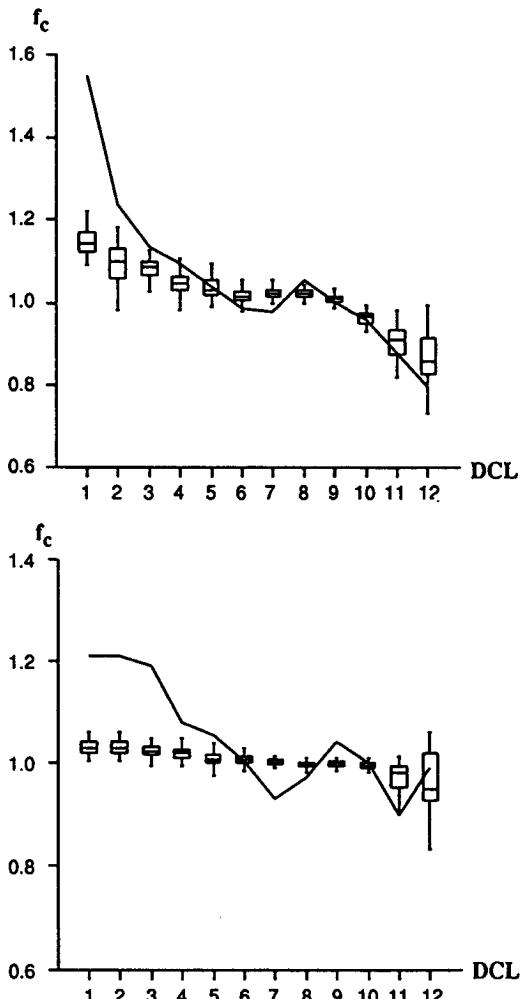


Fig. 7.15: Validation of the distance class distribution in data frame 1988o1 (left) and 1990o2 (right). The box plots represent the frequency distribution of 50 simulation runs, the observed distance class distribution is indicated by the line. f_c = ratio observed/expected disease incidence (Eq. 2.6). DCL = number of distance class (Tab. 2.2).

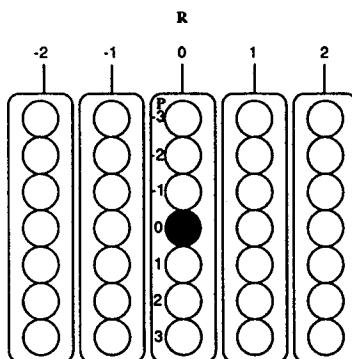


Fig. 7.16: Schematic sampling plan for the validation of tuber infection distributions, shown for a case, where tubers were sampled from three plants in the row on both sides of the source (P_{-3} to P_3) and from two neighbouring rows on both sides (R_{-2} to R_2). The black circle is the secondary infection source plant (P_0).

In data frames 1985n1, 1986n1, 1987n1 and 1990o1 the agreement between the observed and simulated tuber infection in function of the distance from the source was satisfactory, in data frames 1987o1 and 1989o1 the deviations were considerable (Fig. 7.17). Contrary to Fig. 7.15, the values were not standardized. The EPOVIR-model tended to underestimate the infections of the two neighbouring plants in the row of the source and to overestimate the spread to plants in a longer distance from the source, especially across rows.

Over 90 % of the tubers of P_1 -plants were infected, except in 1985n1, where the vector abundance was very low (Fig. 7.14). Virus spread was apparently similarly frequent on both sides of the sources.

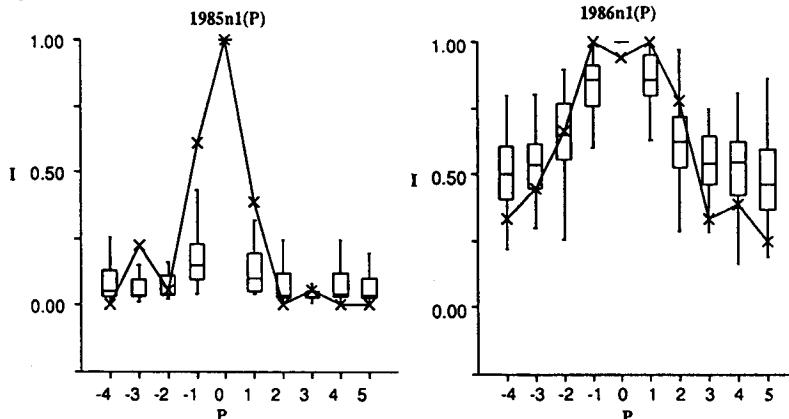


Fig. 7.17: Comparison of the simulated and observed proportion of infected tubers p_I in function of the distance from the source. See Fig. 7.16 for explanation of the labels. x = observed value, boxes = distribution of simulated values.

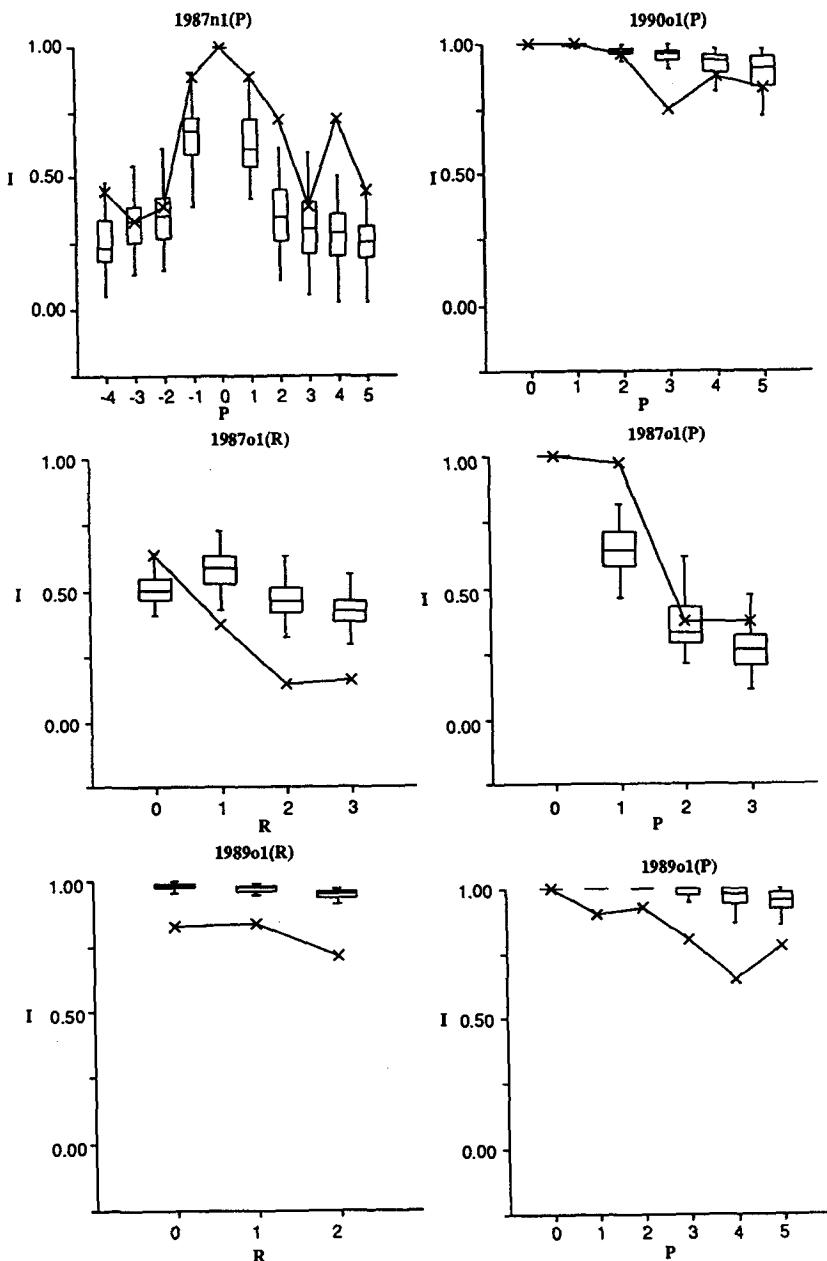


Fig. 7.17 continued.

7.3. Sensitivity Analysis

7.3.1. RELATIVE IMPORTANCE OF VECTOR BEHAVIOUR PARAMETERS

Tab. 7.1: Grouping of the model parameters.

Main group		Subgroup	
Code	Description	Code	Description
B	Behaviour sequence parameters	B ₁	Scaling factors of the frequencies of behaviour states
		B ₂	Scaling factors of the mean residence times of behaviour states
D	Dispersal parameters	D ₁	Parameters affecting vector abundance
		D ₂	Take-off thresholds
		D ₃	Spatial dynamics parameters
		D ₄	Walking parameters
O	Other parameters, not related to vector behaviour	O ₁	Parameters affecting the abundance of unsettled winged aphids
		O ₂	Vector propensity
		O ₃	Initial disease incidence
		O ₄	Plant-virus interaction

Tab. 7.2: Plausibility ranges [p_{nmin} , p_{nmax}] and standard values \bar{p}_n of the parameters p_n in the sensitivity analysis.

p_n	\bar{p}_n	p_{nmin}	p_{nmax}	p_n	\bar{p}_n	p_{nmin}	p_{nmax}	p_n	\bar{p}_n	p_{nmin}	p_{nmax}
k_{fJ}	1	0.25	4	κ	0.345	0.173	0.69	t_{SG+}	1	0.5	2
k_{dI}	1	0.25	4	d_W	0.81	0.405	1.62	v_m	0.0025	0	0.2
s_C	40	20	80	f_{lC+}	1	0.5	2	p_i	0.01	0.002	0.05
ρ_{em}	0.05	0	0.2	a_{ll}	0.069	0.04	0.1	t_{arV+}	1	0.5	2
s_{rC+}	1	0.5	2	p_{wp}	0.111	0.03	0.3	m_t	63.2	31.5	126
$t_{t\ t}$	17	13	25	$t_{t\ b}$	0.785	0.5	0.9	s_t	26.4	13.95	55.8
$t_{t\ w}$	2.5	1	6.5	$t_{t\ w}$	0.367	0.25	0.5	m_j	233.2	144.9	579
$t_{t\ p}$	2	0	5	d_{C+}	1	0.25	4	s_j	90	36.4	146
α	0.657	0.5	1	e_{G+}	1	0.5	1.5	p_{il}	0.692	0.4	1
β	9.613	1	20	r_{IG+}	1	0.5	2				

The sensitivity analysis method was described in Chapter 2.5. The parameters were classified into three main groups according to their relationship to vector behaviour. These main groups were splitted into subgroups (Tab. 7.1). The plausibility ranges of the parameters (Eq. 2.9) are listed in Tab. 7.2. Additional scaling parameters were introduced in the model for the sensitivity analysis, to change parameters with different values for vector groups G , colonization groups C or virus strains V . They are denoted by a '+' sign. These parameters are multipliers of the respective parameters (without '+'). E.g. e_{G+} is multiplied with the vector efficiency parameters e_I to e_3 .

Behavior sequence parameters (B) were among the less sensitive (Fig. 7.18). The parameters determining model behaviour belonged mainly to the dispersal (D) and the other parameters (O).

Out of the parameters influencing the vector's behaviour sequences (B), k_{fw} and k_{dl} had the most significant effects, the former was positively, the latter negatively correlated with disease severity. These parameters had also the most significant effects on the frequencies and durations of behaviour states (Fig. 5.7 and 5.8).

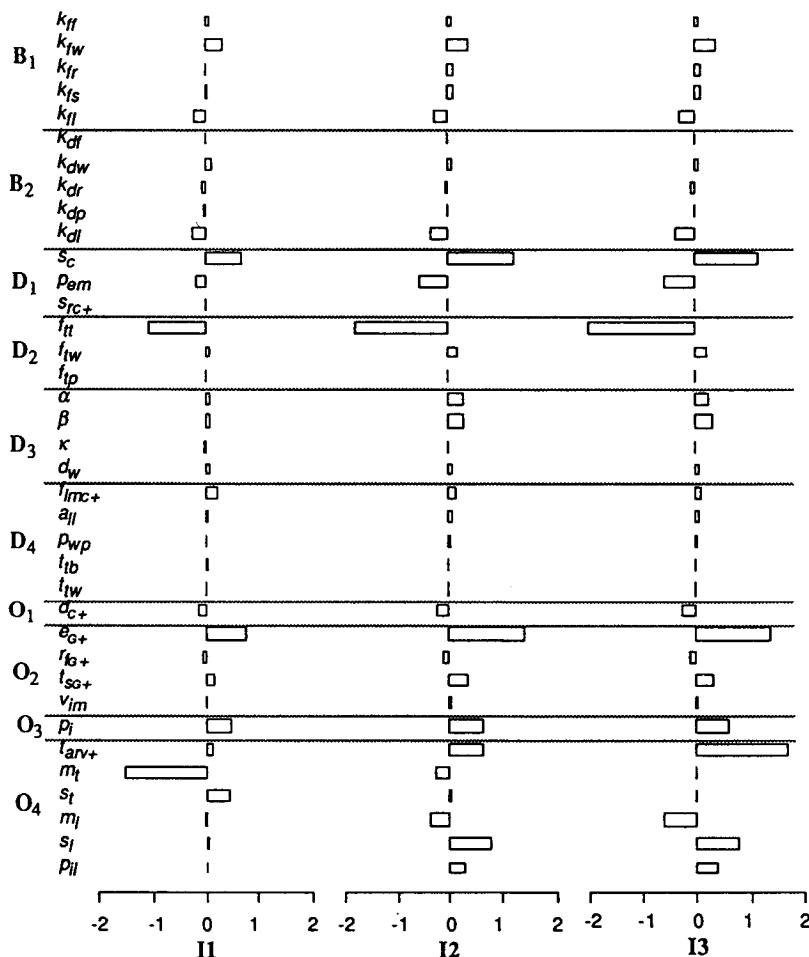


Fig. 7.18: Coefficients of the linear regression of logit-transformed output variables vs the standardized values of the parameters. $I1$ to $I3$ = logit of the proportion of infected tubers, $E1$ to $E3$ = logit of the evenness of distance class distribution at monitoring times t_1 to t_3 .

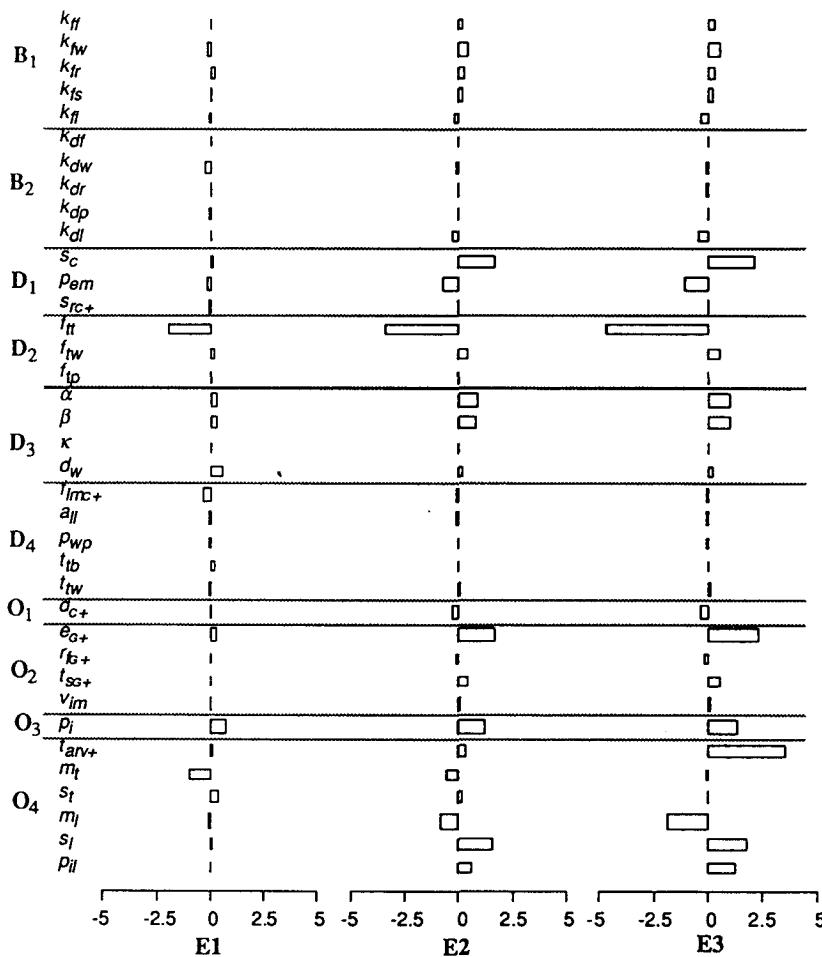


Fig. 7.18 continued.

The parameters of group D determine the dispersal and via influence on the immigration and emigration behaviour also the abundance of the vectors in the field. Some of the parameters have a direct influence on the spatial dispersal mechanisms and affect thus the spatial pattern. Parameters determining immigration and emigration, namely the concentration factor s_c and the emigration probability p_{em} were among the most sensitive model parameters (group D₁). The temperature and wind speed thresholds (t_{ff} and t_{fw}) were the most sensitive take-off thresholds (group D₂), whereas the precipitation threshold f_{fp} did not influence the epidemic progress at all. Out of parameter group D₃, only the parameters of the flight distance distribution (α and β) affected disease spread. Changes

in the distribution parameter κ of the flight directions had no influence on the model outputs. The mean walking distance d_w affected the spatial pattern during the early phases of the growing season, but not the disease severity. The rate of interleaf movement by walking f_{lmC+} was the only walking parameter (group D₄) having an effect on the model outputs.

Parameter group O is a collection of parameters with no direct relationship to vector behaviour. The mortality rate d_{C+} (group O₁) was a little sensitive parameter. In group O₂ (vector propensity parameters), the vector efficiency e_{G+} had a dominant effect, followed by the serial inoculation probability t_{sG+} . The other two parameters produced only minor changes in the model outputs. The initial disease incidence p_i (group O₃) was of moderate sensitivity. In group O₄, the plant-virus interaction parameters, the age-resistance threshold t_{arv+} and the latent period (m_l and s_l) had the largest absolute regression coefficients.

7.3.2. EFFECTS OF SINGLE PARAMETERS

In the following the responses of the output variables to changes in selected parameters are shown. To make comparisons easier, the output variables as well as the parameter values were standardized by dividing by the value in the standard run. The output values were not plotted on a logit-scale. The Y-axis crosses the X-axis at the standard value of the parameter, i.e. the value in the standard run.

7.3.2.1. Behaviour Sequence Parameters

Reducing k_{ff} (Fig. 7.19) slowed down virus spread and affected also the spatial pattern, while increasing this parameter had hardly any effect. It seems that flight frequency was near an optimum for virus dissemination. Note that the emigration rate is proportional to the flight frequency in the EPOVIR-model.

k_{fw} was positively correlated to disease severity and negatively to the evenness at t_1 . This increased clumping can be explained by favoured virus transmission by walking aphids. At t_2 and t_3 the clumping was similar to the standard run, because most of the plants in the proximity of the infection sources were already infected and thus interplant movement by walking contributed little to virus spread. Changes in k_{fw} influenced other behaviour states significantly, especially their frequency (Fig. 5.8).

Changes in k_{fr} had little effect on the disease progress, except on the evenness at t_1 , where increased k_{fr} produced a higher evenness of the distance class distribution. This could be due to a close relationship between the behaviour states resting, walking and in part flight (Fig. 5.5). Increases of k_{fr} increase the flight frequency, but reduce the proportion of time spent walking. Since the interplant movement by walking is related to the proportion of time spent walking (Eq. 6.10 and 6.11), inoculations by flight were favoured and inoculations by walking reduced.

The sensitivity of the output variables to k_{fp} was surprisingly low. Similar to flight frequency, probing frequency had apparently nearly optimal values for virus dissemination.

Contrary to the other scaling factors of behaviour frequencies, k_{fj} was negatively correlated with PVY spread.

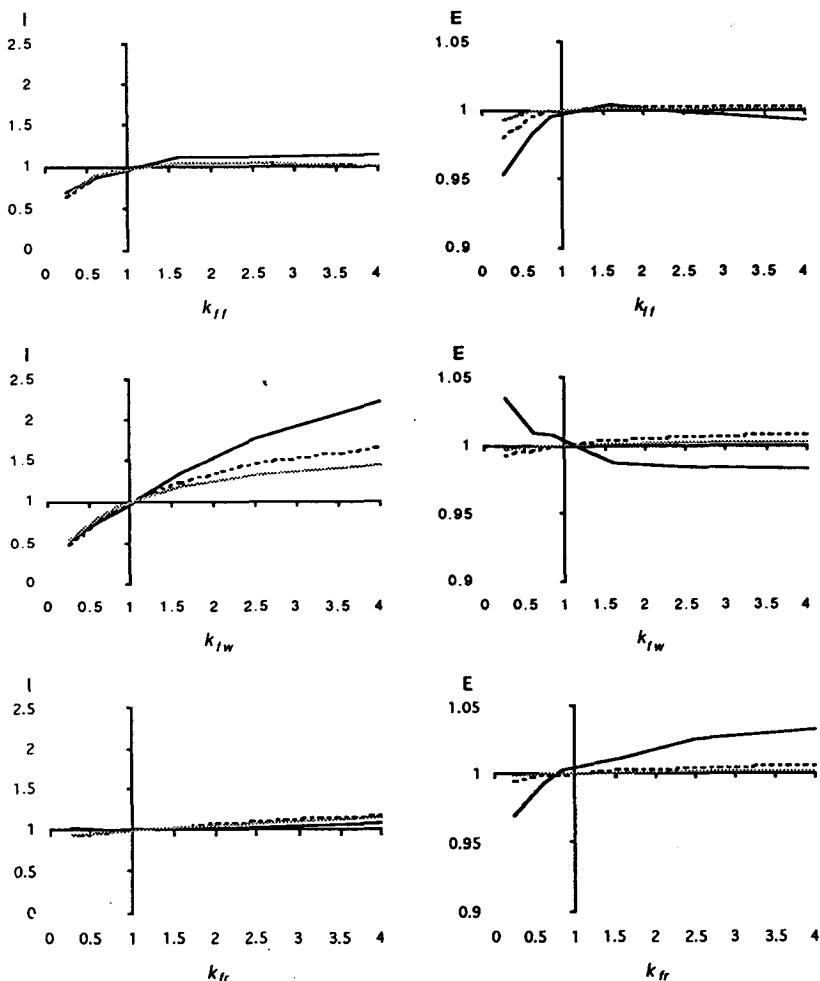


Fig. 7.19: Response of the standardized output variables I = proportion of infected tubers and E = evenness of the distance class distribution to changes in the frequency scaling factors k_{fj} (parameter group B_1) of the five behaviour states (J) at three monitoring times: t_1 : solid line, t_2 : dashed line and t_3 : dotted line.

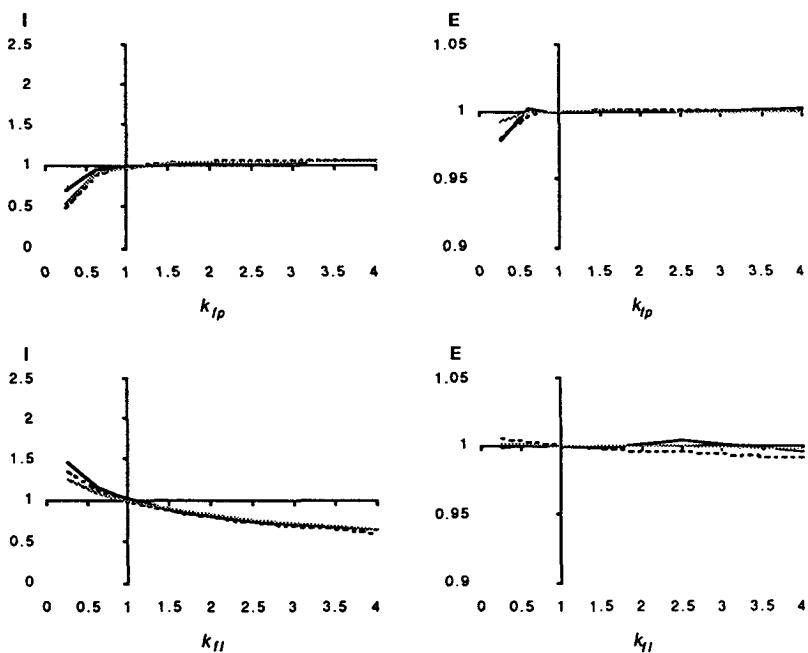


Fig. 7.19 continued.

Changing the mean residence time of walking by k_{dw} (Fig. 7.20) affected both, the disease severity and the distance class distribution at t_1 but little at t_2 or t_3 , indicating that walking aphids contributed to virus spread mainly early in the season.

Similarly to k_{ll} , k_{dl} was negatively correlated with disease severity. Of the five scaling parameters of the mean residence times this was obviously the most sensitive one. The other scaling factor of the mean residence times had only minor effects on the two output variables.

To answer the question, whether the model outputs reacted differently, if only behaviour sequence parameters of colonizing resp. noncolonizing aphids were changed, another sensitivity analysis experiment was performed, where the parameters k_{ff} and k_{df} were changed for colonizing resp. noncolonizing aphids only (according to Eq. 2.9). Since about 90 % of the infection were caused by noncolonizing aphids in the standard run (Fig. 7.28), the model was more sensitive to changes in parameters of noncolonizing than to those of colonizing species. The responses were qualitatively similar for parameters of both aphid groups, except for k_{ff} , where the results are shown in Fig. 7.21. It seems that there exist an optimal value for this parameter and that noncolonizing aphids have a parameter value close to this optimum. In the plausibility range used, which did not contain the standard value of the noncolonizing species, such an optimum was not detected for the colonizing species; tuber infection increased with increasing k_{ff} . Note that

the flight frequency of noncolonizing species was about five times higher than that of colonizing species.

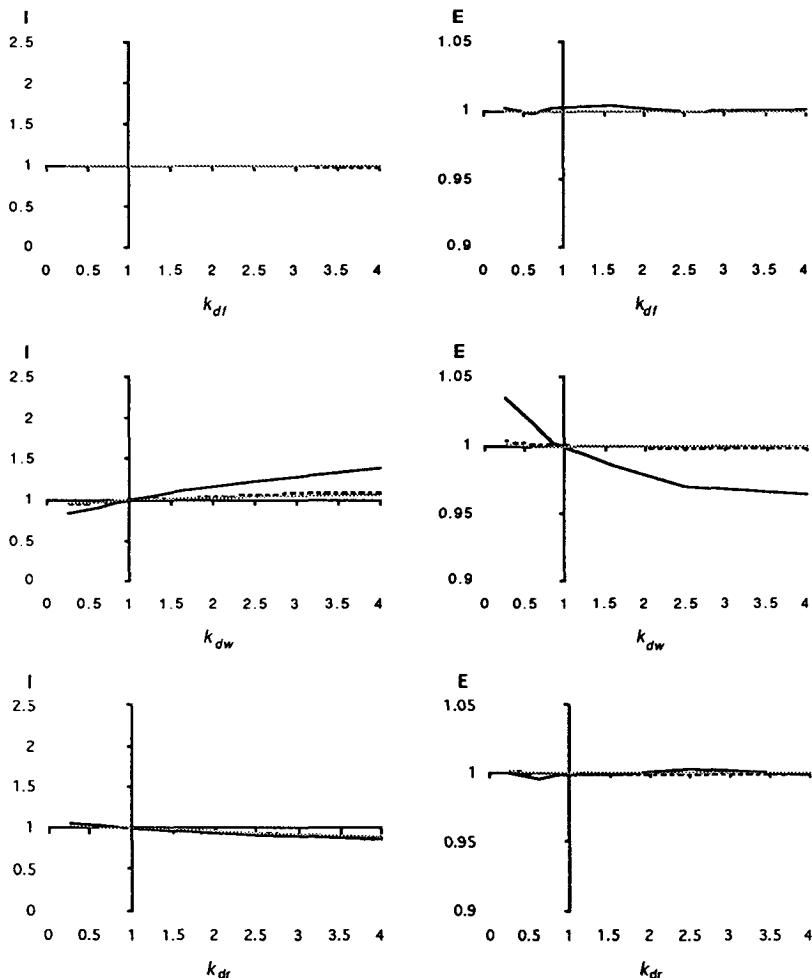


Fig. 7.20: Response of the standardized output variables I = proportion of infected tubers and E = evenness of the distance class distribution to changes in the scaling factors of the mean residence times k_{dI} (parameter group B_2) of the five behaviour states (I) at three monitoring times: t_1 : —, t_2 : - - -, and t_3 : - · - .

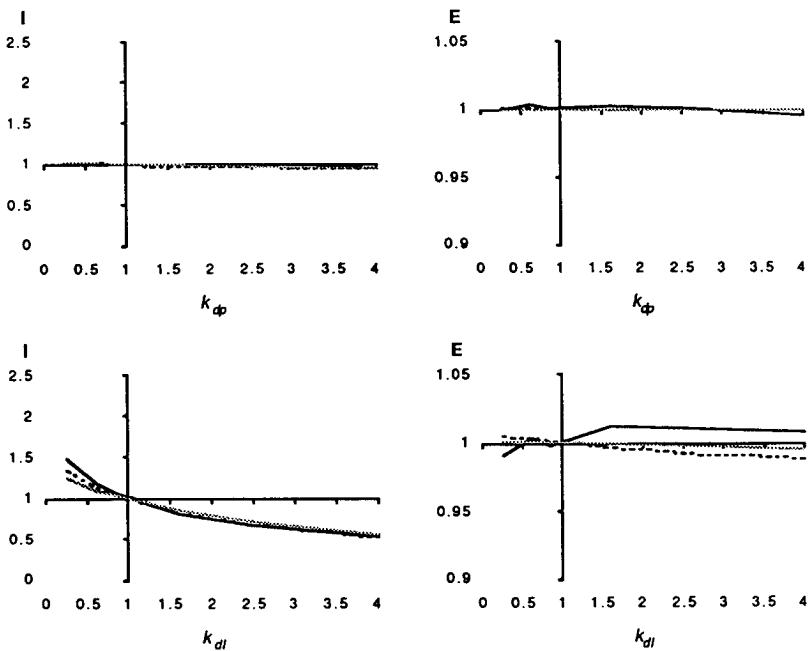
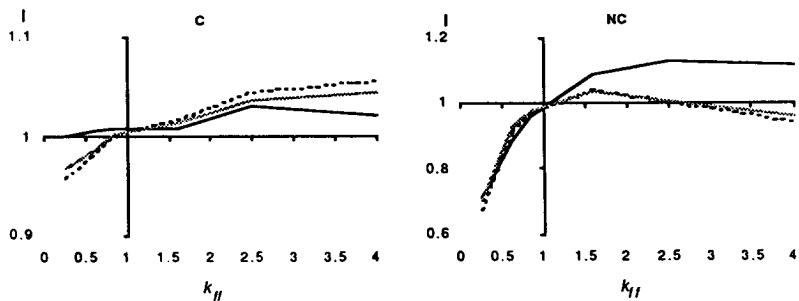


Fig. 7.20 continued.

Fig. 7.21: Response of the standardized output variable I = proportion of infected tubers to changes in the scaling factor of the flight frequency k_H of colonizing (C) and noncolonizing (NC) species at: l_1 : —, l_2 : - - -, and l_3 : ······.

7.3.2.2. Dispersal Parameters

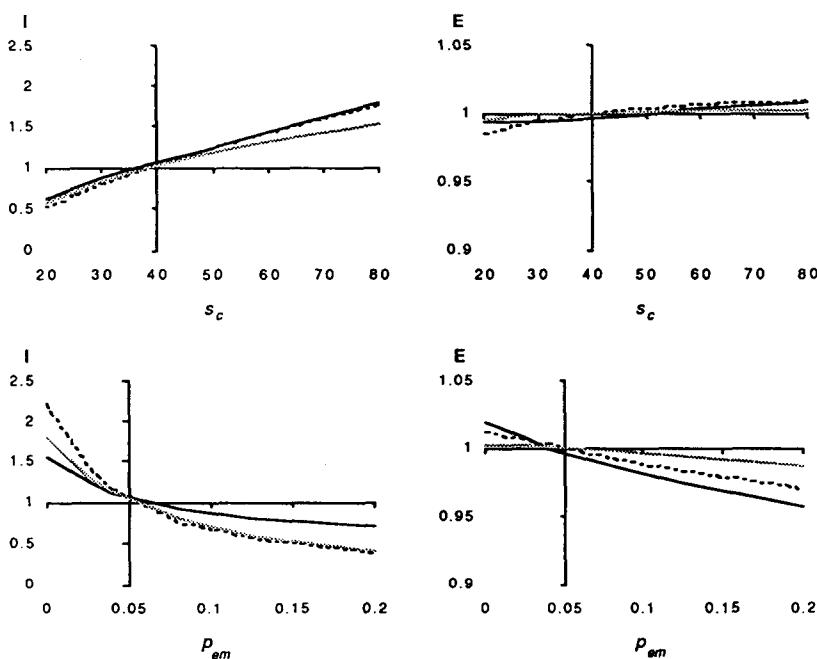


Fig. 7.22: Response of the standardized output variables I = proportion of infected tubers and E = evenness of the distance class distribution to changes in dispersal parameters affecting vector abundance (parameter group D₁): s_c = concentration factor, p_{em} = probability to emigrate at t_f : ———, t_f2 : - - -, and t_f3 :

The proportion of infected tubers was almost linearly correlated with s_c (Fig. 7.22). Increased p_{em} considerably slowed down the disease progress.

Increases of the temperature threshold and decreases of the wind speed threshold affected virus spread, whereas changes in the opposite sense had little effect (Fig. 7.23). The spatial pattern seems to be affected more than the temporal virus epidemics, probably because flight but not walking was restricted by the thresholds, leading to a more clumped pattern.

The flight distance distribution parameter α hardly affected the epidemic, except that higher α -values resulted in a slightly higher evenness of the distance class distribution (Fig. 7.24). Small values of β (< 5) reduced virus spread and lead to a stronger clumping of infections around the sources (see also Fig. 4.3).

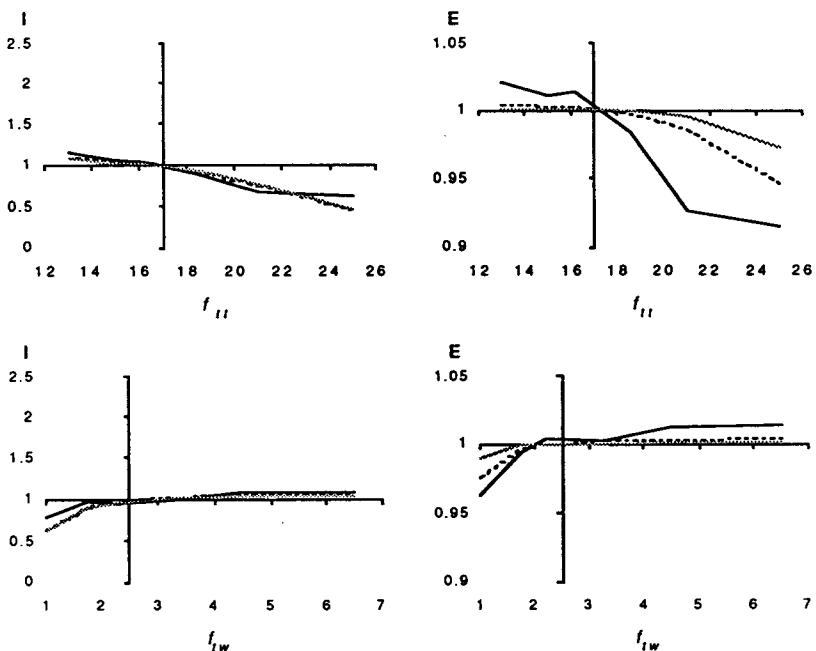


Fig. 7.23: Response of the standardized output variables I = proportion of infected tubers and E = evenness of the distance class distribution to changes in the meteorological take-off thresholds (parameter group D₂) f_H (= temperature threshold) and f_W (= wind speed threshold) at t_1 : ———, t_2 : - - - and t_3 : ······.

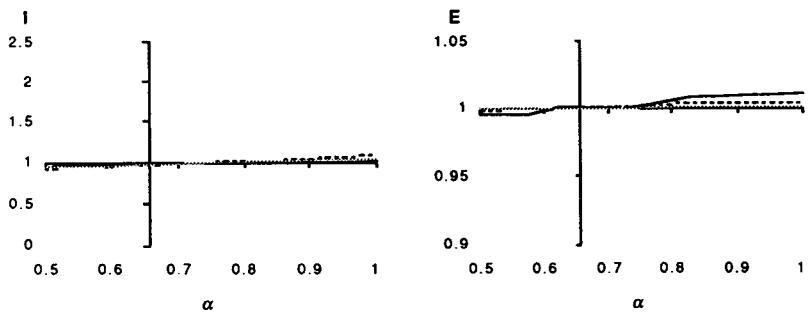


Fig. 7.24: Response of the standardized output variables I = proportion of infected tubers and E = evenness of the distance class distribution to changes in the parameters α and β of the flight distance distribution (parameter group D₃): t_1 : ———, t_2 : - - - and t_3 : ······.

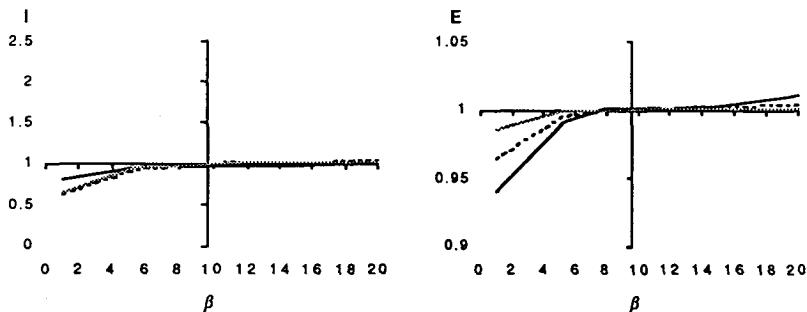


Fig. 7.24 continued.

7.3.2.4. Other Parameters

The mortality rate (Fig. 7.25) influenced virus dissemination only moderately in the parameter range used in the sensitivity analysis. Insecticide treatments could increase natural mortality by a factor 10 to 100 (LOWERY & BOITEAU, 1988), i.e. much more than it was done in the sensitivity analysis.

The vector efficiency e_{G+} and the proportion of immigrants coming from sources v_{im} were positively correlated with disease severity (Fig. 7.26). High v_{im} values tended to increase the evenness of the distance class distribution, because inoculations caused by immigrating aphids were distributed uniformly over the field.

Changes in the scaling factor of vector efficiency e_{G+} produced approximately proportional changes in the fraction of tubers infected. As suggested also by Fig. 7.18 e_G was one of the main parameters determining the epidemic progress.

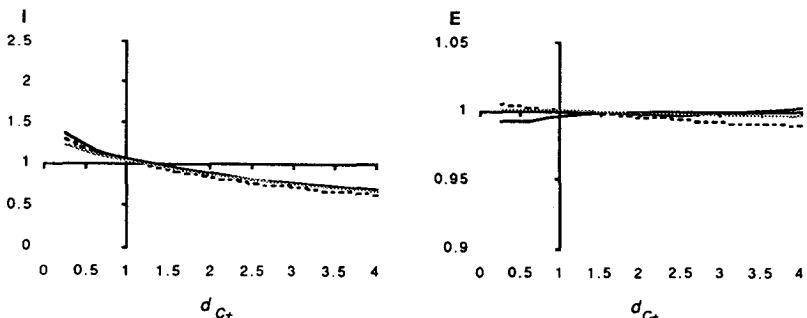


Fig. 7.25: Response of the standardized output variables I = proportion of infected tubers and E = evenness of the distance class distribution to changes in the mortality scaling factor d_{C+} (parameter group O₁) at t_1 : —, t_2 : - - - and t_3 : ······.

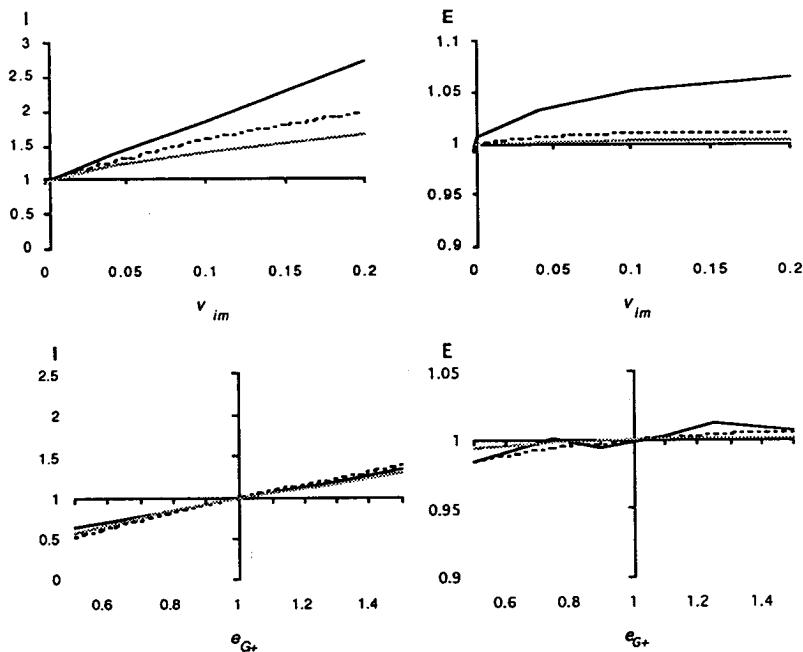


Fig. 7.26: Response of the standardized output variables I = proportion of infected tubers and E = evenness of the distance class distribution to changes in vector propensity parameters (group O_2): v_{im} = proportion of immigrating aphids coming from infection sources, e_{G+} = vector efficiency scaling factor at t_f : —, t_2 : - - - and t_3 : ·····.

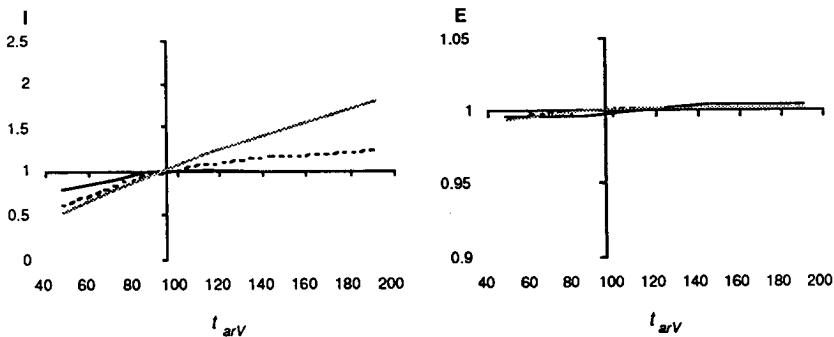


Fig. 7.27: Response of the standardized output variables I = proportion of infected tubers and E = evenness of the distance class distribution to changes in the age resistance threshold t_{arV} and the initial disease incidence p_i (parameter groups O_3 and O_4) at t_f : —, t_2 : - - - and t_3 : ·····.

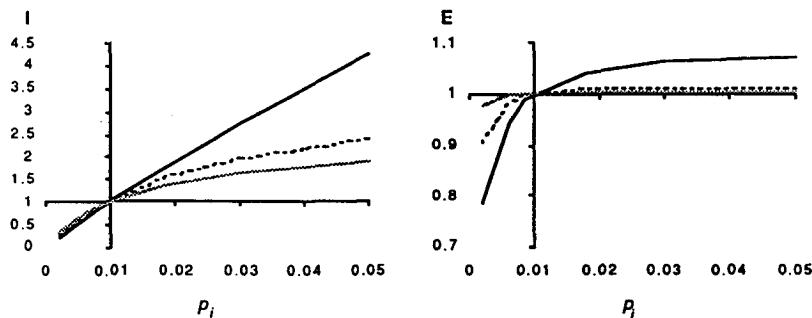


Fig. 7.27 continued.

The age resistance threshold t_{arv} played an important role in the temporal disease spread, but had only a marginal effect on the spatial pattern (Fig. 7.27).

The initial disease incidence p_i affected both the temporal and spatial PVY dynamics (see also Fig. 4.4 & 7.7). The response of the proportion of infected tubers was almost linear at t_1 , but apparently reached a saturation for high p_i -values at t_2 and t_3 , due to the fact that almost all tubers were infected at the end of the growing season.

7.4. The Role of Vector Groups and Aphid Morphs

7.4.1. COLONIZING AND NONCOLONIZING APHIDS

The contribution of the different vector groups (Tab. 6.1) to the number of infections occurring was estimated in the data frames used in Chapter 7.1.3 (data set I), keeping the initial disease incidence p_i at 0.01 in all frames. To estimate the contribution of the different groups, the number of infections per day were multiplied by the fraction of inoculations, caused by a vector group during this period:

$$p_G = \frac{\sum_{k=t_{em}}^{t_{hk}} \left(\frac{i_G(k)}{i(k)} n_{inf}(k) \right)}{\sum_{k=t_{em}}^{t_{hk}} n_{inf}(k)} \quad (7.2)$$

where p_G is the estimated contribution of group G to the infections, $i_G(k)$ is the number of inoculations caused by vector group G on day k , $i(k)$ is the total number of inoculations on day k , $n_{inf}(k)$ is the number of infections on day k , i.e. the number of cells passing from state δ_h to δ_l .

Group 2b contributed on the average 68 % to the total PVY transmissions (Fig. 7.28). This group is composed of the noncolonizing species *A. fabae*, *B. helichrysi* and *P. hunuli* (Tab. 6.1). The mean contribution of colonizing aphids (groups 1 and 2a) in the eight years was only 18 %, *M. persicae* accounted for most of it. The

contribution of the different vector groups was highly variable from year to year, mainly due to the varying composition of the vector complex.

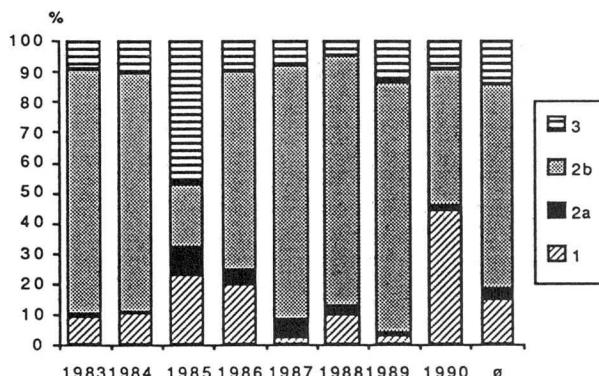


Fig. 7.28: Estimated contribution of the four vector groups to PVY infections (Y-axis). Groups 1 and 2a are groups of colonizing aphid species, group 2b and 3 are noncolonizing species.

To test the hypothesis that the restlessness of noncolonizing aphids leads to higher virus spread rates (KENNEDY, 1976; ROMANOW, 1985; CARTER & HARRINGTON, 1991) the Markov matrix parameters of colonizing, resp. noncolonizing aphids were assigned to all vectors. All other colonization group specific parameters, which were not dependent on the Markov transition matrix (i.e. mortality d_C , settling rate s_{RC} and rate of interleaf movement by walking f_{lmc}), were set to the average of the values for the two groups.

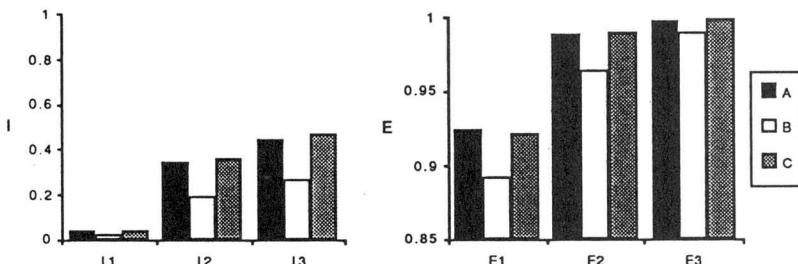


Fig. 7.29: Effect of the behaviour sequences of colonizing and noncolonizing aphids. A: standard run, B: all aphids behaving like colonizing aphids, C: all aphids behaving like noncolonizing aphids. I1 to I3: proportion of infected tubers, E1 to E3: evenness of distance class distribution at t_1 to t_3 .

The quantitative differences in behaviour sequences of colonizing and noncolonizing aphids resulted in differences in epidemic progress (Fig. 7.29). If all aphids behaved like colonizers, the proportion of infected tubers would be nearly halved, compared with a situation, where all aphids behaved like noncolonizers. This ratio seems to be similar to the ratio of the frequencies of behaviour state changes of colonizing and noncolonizing aphids (Chapter 5.3). The difference of experiment C to the standard run (A) was small, because the noncolonizing species were responsible for most of the infections (Fig. 7.28).

7.4.2. WINGED AND WINGLESS APHIDS

Wingless aphids were ignored in the EPOVIR-model, since they are considered to be of minor importance for PVY spread (BROADBENT & TINSLEY, 1951). To verify, whether this assumption was justified, the potential contribution of apterae to PVY dissemination was assessed by a simulation experiment. Only apterae of colonizing aphids appear in the potato crop. KLINGAUF (1976) found that most of the wingless aphids moving were young adult apterae. Hence, larvae were assumed to be irrelevant. It was assumed that adult apterae disperse by walking in a similar way as young unsettled alatae do, i.e. that they disperse during the first two to three days.

The contribution of apterae to PVY spread was simulated by multiplying the rate of interleaf movement by walking of colonizing aphids $f_{lm\ col}$ by an 'apterae factor' k_{apt} . This effect increased linearly from emergence to haulm-killing (Eq. 7.3). k_{apt} corresponds to the ratio of the abundances of adult apterae and alatae colonizers at haulm-killing. Since noncolonizing species do not develop colonies in the potato crop, $f_{lm\ ncol}$ was not changed. k_{apt} was varied between 0 and 100. Experimental data (DERRON, unpubl.) suggest that an apterae factor of 5 to 10 would correspond to field situations in Western Switzerland.

$$f'_{lm\ col} = f_{lm\ col} \frac{t - t_{em}}{t_{hk} - t_{em}} (1 + k_{apt}) \quad (7.3)$$

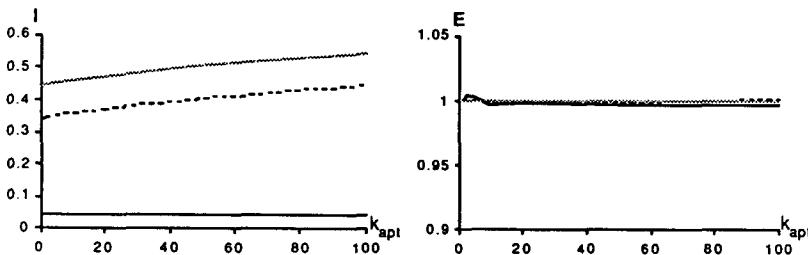


Fig. 7.30: Simulation of the effect of wingless aphids. k_{apt} = 'apterae factor' (Eq. 7.3). I = proportion of infected tubers, E = evenness of the distance class distribution at: t_1 : ———, t_2 : ----- and t_3 :

The proportion of infected tubers increased significantly with higher k_{apt} (Fig. 7.30). However, for a k_{apt} -value of 5 to 10, the relative difference in tuber infection was only 2-3 %. The spatial pattern was surprisingly little affected.

7.5. Assessment of Control Measures Affecting Winged Aphid Behaviour

Tab. 7.3: Scenarios of behavioural responses of aphids to control measures. The parameter values in the standard run are shown in bold face.

Scenario	Parameter	Values	Meaning	Sources
Reduced immigration	s_C	0.4, 0.8, 2, 4, 8, 16, 24, 32, 40	reduced immigration rate	KRING (1972), LOEBENSTEIN & RACCAH (1980), GIBSON & RICE (1989), KUROLI & ERDÉLYI (1990)
Aphicides	k_{fp}/k_{dl}	0.1/0.05, 1/1, 1.5/1	reduced probing time and unchanged probing frequency, increased probing frequency	SHANKS & CHAPMAN (1965), LEHMANN et al. (1975), SASSEN (1983), LOWERY & BOITEAU (1988)
	$k_{fw} & k_{ff}$	1, 2, 5	increased walking and flight frequency	RICE et al. (1983), LOWERY & BOITEAU (1988), GIBSON & RICE (1989)
	P_{em}	0.05, 0.1	increased emigration probability	RICE et al. (1983)
Repellents	dC_+	1, 10, 100	increased mortality rate	LOWERY & BOITEAU (1988)
	k_{dl}	0.25, 0.5, 1	decreased mean penetration time	PHELAN & MILLER (1982)
	k_{ff}	1, 1.5, 2	increased flight frequency	PHELAN & MILLER (1982)
Plant resistance	P_{em}	0.05, 0.1	increased emigration probability	PHELAN & MILLER (1982)
	k_{fp}	0.5, 1, 1.5	decreased or increased probing frequency	HENNIG (1969), TARN & ADAMS (1982), HARREWIJN (1986)
	$k_{ff} & k_{fw}$	1, 2, 5	increased walking and flight frequency	
Alarm pheromones	P_{em}	0.05, 0.1	increased emigration probability	
	k_{dl}	0.5, 1	decreased mean penetration time	PHELAN & MILLER (1982)
	k_{fw}	1, 1.5	increased walking frequency and time spent walking	PHELAN & MILLER (1982)
	P_{em}	0.05, 0.1	increased emigration probability	PHELAN & MILLER (1982)

Although many reports have been published on experimental modifications of aphid behaviour (GIBSON & RICE, 1989; see also Chapter 3.1.2), quantification of these effects usable for a simulation study is scarce. Making use of these reports, scenarios of possible changes in aphid behaviour induced by man were derived and subsequently used to assess effects of these changes in aphid behaviour on PVY epidemiology (Tab. 7.3). Many combinations of parameter values were used to cover a wide range of possible outcomes. Each value represents a level of one or two parameter, all combinations of these parameter levels were realized in the simulation experiments.

7.5.1. MEASURES REDUCING IMMIGRATION

The change of disease severity (Fig. 7.31, I) was almost proportional to the change of the concentration factor s_C , which is proportional to the immigration rate. The even-

ness of the distance class distribution increased with increasing s_C , which is probably only an effect of the higher overall disease incidence.

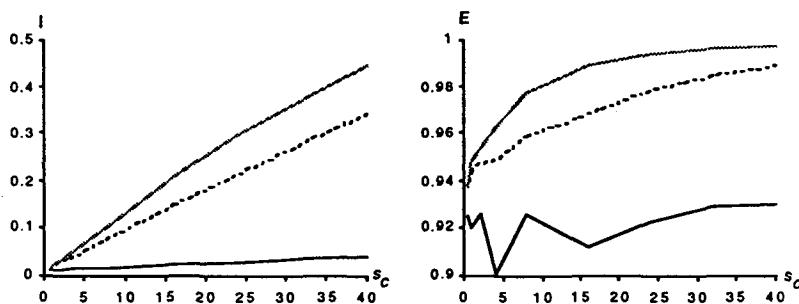


Fig. 7.31: Results of the 'reduced immigration' scenario. s_C = concentration factor, I = proportion of infected tubers, E = evenness of the distance class distribution at: I_1 : ———, I_2 : - - - and I_3 :

7.5.2. APHICIDES

Only effects on aphid behaviour and mortality were included in the scenario. Other secondary effects on the ecosystem, e.g. predator mortality, were ignored. Aphicide application was simulated in intervals of ten days ($k = 140 + i \cdot 10$), the effect lasted for 5 days. During the phases, when the aphicide was effective, altered behaviour sequence parameters were applied, during the rest of the time, the standard parameters were used.

Increases of mortality induced by aphicides approximately reduced the final disease severity logarithmically (Fig. 7.32). Increased emigration probability p_{em} slowed down, increased flight and walking frequency favoured epidemic progress. Surprisingly, reduced k_{dl} together with a reduction of k_{fp} considerably decreased disease severity. Increased flight and walking frequency increased the proportion of infected tubers except in experiment P1, where these frequencies were already very high. The differences between P2 and P3 were small, due to only marginal differences in behaviour parameters.

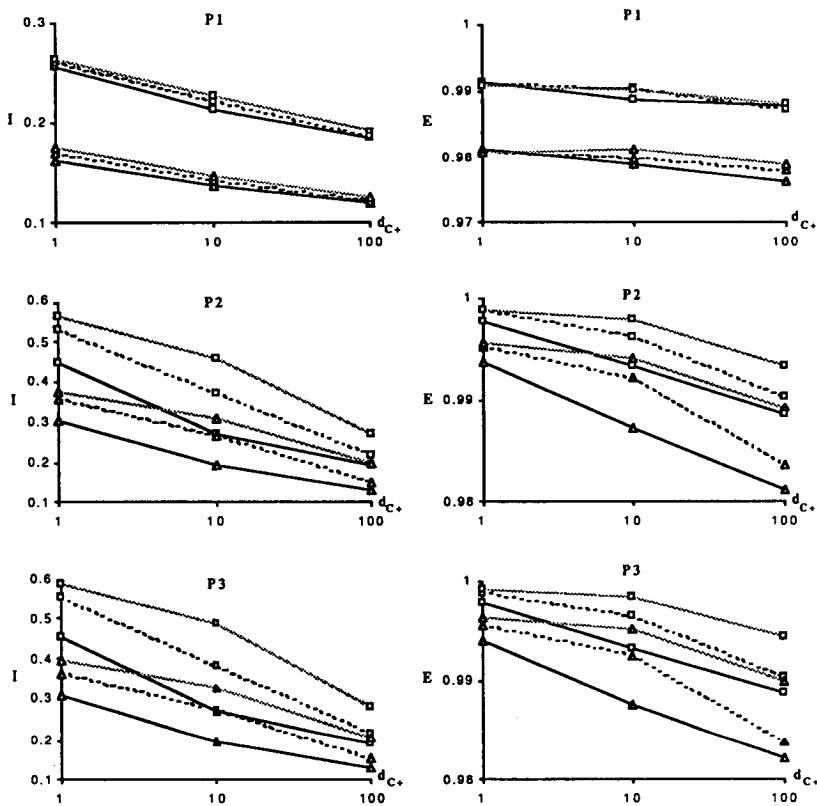


Fig. 7.32: Results of the 'aphicide' scenario. Effects of increased mortality rate d_{C+} and induced changes in behaviour on the proportion of infected tubers I and the evenness of the distance class distribution E at t_3 . Squares: $p_{em} = 0.05$, triangles: $p_{em} = 0.1$, unbroken lines: $k_{ff} = 1$ and $k_{fw} = 1$, broken lines: $k_{ff} = 2$ and $k_{fw} = 2$, spotted lines: $k_{ff} = 5$ and $k_{fw} = 5$, P1: $k_{fp} = 0.1$ and $k_{dl} = 0.05$, P2: $k_{fp} = 1$ and $k_{dl} = 1$, P3: $k_{fp} = 1.5$ and $k_{dl} = 1$.

7.5.3. REPELLENTS

Repellents could potentially increase disease spread by a reduction of the time spent in state 'long penetration' (Fig. 7.33). Reducing k_{dl} is coupled with an increase in flight, walking and probing frequencies, which is the reason why disease severity was higher in these cases. Increased k_{ff} slightly favoured virus spread, but could have also the opposite effect, if $p_{em} = 0.1$. A higher emigration probability led to significantly reduced PVY spread.

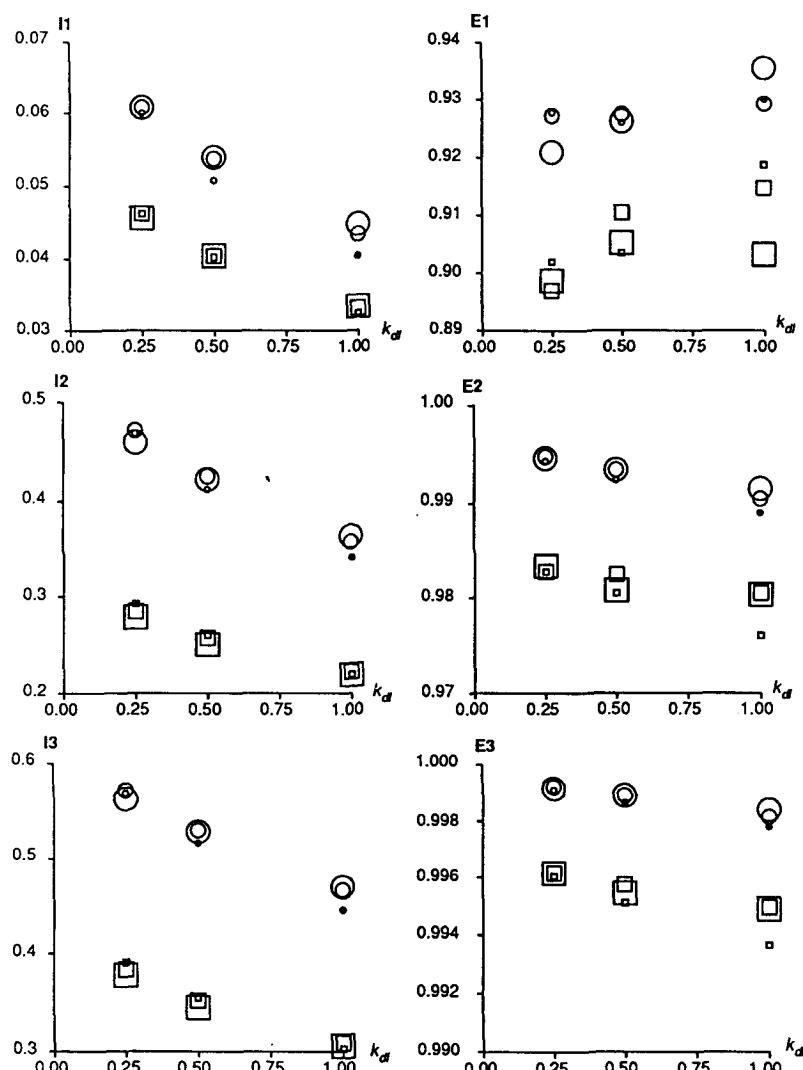


Fig. 7.33: Results of the 'repellents' scenario. I_1 to I_3 = fraction of tubers infected, E_1 to E_3 = evenness of the distance class distribution at t_1 to t_3 , k_d = scaling factor of the mean residence time of 'long penetration'. The size of the symbols shows the level of the scaling factor of flight frequency k_{ff} (small symbols: $k_{ff} = 1$, medium symbols: $k_{ff} = 1.5$, large symbols: $k_{ff} = 2$), the shape of the symbols shows the emigration probability p_{em} (circles: $p_{em} = 0.05$, squares: $p_{em} = 0.1$). The value in the standard run is shown by a small filled circle.

7.5.4. PLANT RESISTANCE TO APHIDS

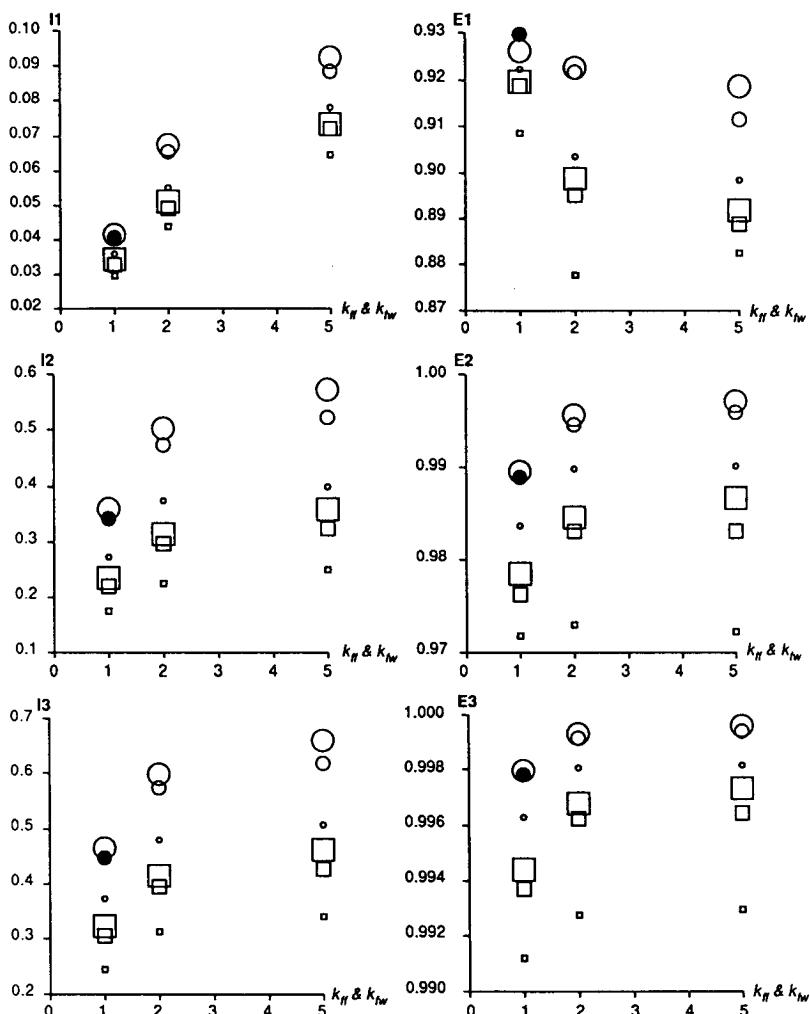


Fig. 7.34: Results of the 'plant resistance' scenario. I_1 to I_3 = proportion of infected tubers, E_1 to E_3 = evenness of the distance class distribution at I_1 to I_3 , k_{ff} and k_{fw} = scaling factors of flight resp. walking frequency. The size of the symbols shows the level of the scaling factor of probing frequency k_{fp} (small: $k_{fp} = 0.5$, medium: $k_{fp} = 1.0$, large: $k_{fp} = 1.5$), the shape of the symbols shows the emigration probability p_{em} (circles: $p_{em} = 0.05$, squares: $p_{em} = 0.1$). The value in the standard run is shown by a filled circle.

A combination of aphid resistance by antixenosis, leading to increased flight, walking, probing frequencies and emigration probability, and of vector resistance by re-

duced probing frequency is simulated in the scenario. PVY spread could be increased by increasing flight and walking frequencies and decreased by a higher emigration probability (Fig. 7.34). Changing k_{fp} and k_{fw} from 1 to 2 had a considerable effect, whereas a further increase to 5 affected the epidemic much less. Higher values of k_{fp} hardly increased virus dissemination, but smaller k_{fp} led to reduced virus spread, which seems to be consistent with the results of the sensitivity analysis.

7.5.5. ALARM PHEROMONES

Alarm pheromone applications increasing emigration from the field could effectively control virus diseases (Fig. 7.35). On the other hand virus spread could be favoured by reducing the penetration duration. The effect of the walking frequency scaling factor k_{fw} was marginal, since the values deviated little from the standard run.

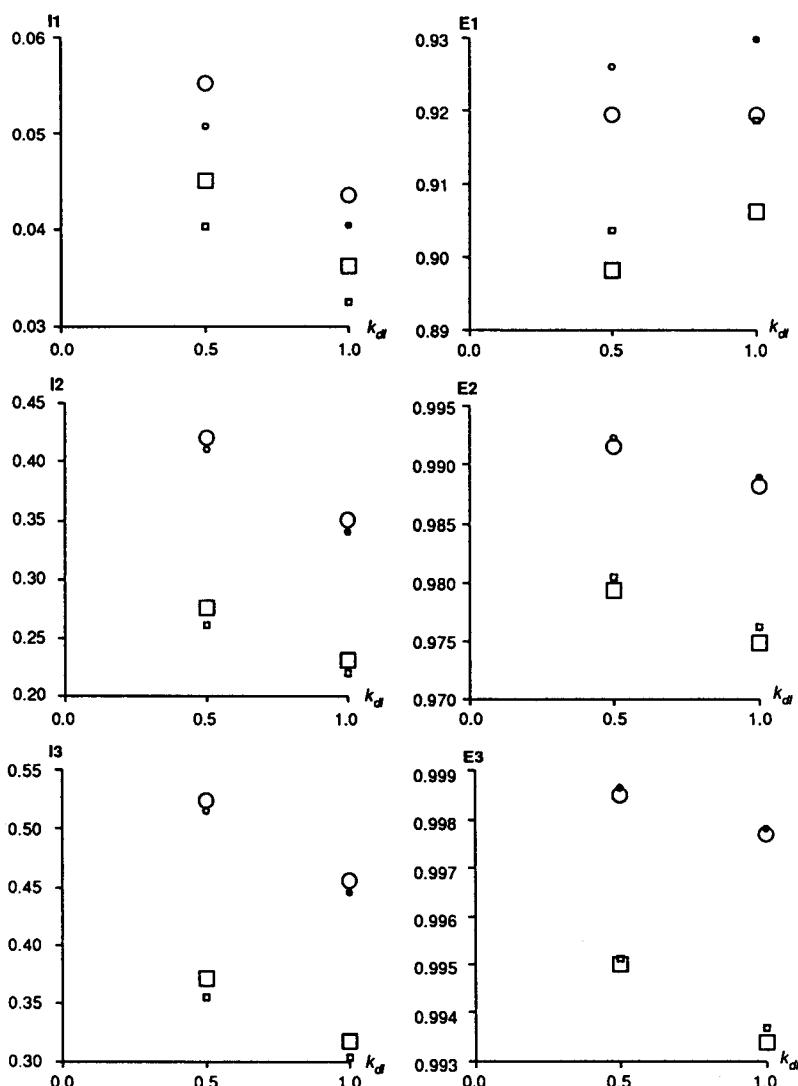


Fig. 7.35: Results of the 'alarm pheromone' scenario. I_1 to I_3 = fraction of tubers infected, E_1 to E_3 = evenness of the distance class distribution at t_1 to t_3 , k_{dl} = scaling factor of the mean residence time of 'long penetration'. The size of the symbols shows the level of the scaling factor of walking frequency k_{fw} (small: $k_{fw} = 1$, large: $k_{fw} = 1.5$), the shape of the symbols shows the level of emigration probability p_{em} (circles: $p_{em} = 0.05$, squares: $p_{em} = 0.1$). The result of the standard run is shown by a filled circle.

8. DISCUSSION

8.1. Model Structure and Behaviour

The EPOVIR-model

The EPOVIR-model is the first epidemic model of nonpersistent plant virus diseases using a model of the vector's behaviour sequences to calculate the infection rate. It has permitted to test various hypotheses on the role of vector behaviour in nonpersistent virus epidemiology. The Markov chain model used, which assumes constant transition rates, is certainly a rough simplification of vector behaviour, but it seems sufficient for the objectives of the study.

Unlike most other epidemic models, the EPOVIR-model simulates virus epidemics in time and space. This was necessary, since the spatial pattern is closely related to vector behaviour (GRAY *et al.*, 1986a).

The EPOVIR-model is further the first simulation model of plant virus diseases including a plant growth submodel. Contrary to other virus epidemic models, which calculate age resistance as a function of time (e.g. RUESINK & IRWIN, 1986; SIGVALD, 1986), this approach has allowed to define age resistance as a function of the physiological state of the plant. Hence the age resistance function in the EPOVIR-model is sensitive to the environment through influences on the plant growth submodel. Moreover, processes like migration by walking were made dependent on plant phenology.

The three features mentioned above (inclusion of vector behaviour, modelling epidemics in time and space and incorporation of plant growth and a soil water balance submodels) made the EPOVIR-model more complex than most other models. To keep its structure clear and tractable, the model was organized in a modular and hierarchical manner (Fig. 6.2). The concept of inoculation and infection submodels enhanced the clarity of the model (Fig. 6.1).

The applicability of the current model version is limited to the simulation of PVY epidemics in seed potatoes, where virus disease incidence is usually so low that the yield reduction caused is negligible. To simulate yield reduction in ware potatoes, the impact of the virus infections on plant growth should be included, especially in cases, where non-certified seed is used. The described spatial stochastic version of the EPOVIR-model is a research tool, allowing to test hypothesis about the PVY-pathosystem. It is particularly useful where the spatial component of the epidemic is of special importance or interest. For an application in practice, e.g. for forecasting of tuber infection, it is preferable to use the deterministic infection submodel of EPOVIR, which requires much less computation time and parameters. The infection rate in this submodel must be corrected for the non-random spatial distribution of the inoculations; the parameters of these spatial distributions can be estimated by the spatial stochastic infection submodel.

For a practical application in the seed potato production, the model should be capable to forecast 1) the tuber infection by PVY and PLRV and 2) the marketable tuber yield. For this purpose the model needs to be adapted in the following ways: the model should

be simplified by eliminating all parameters of minor sensitivity (the structural and sensitivity analysis provides a base, see Chapters 7.1.4, 7.3 & 8.3), the inoculation submodel should be extended to PLRV (the Markov model of vector behaviour provides a sound base for a model using current knowledge of the transmission process), a submodel for the tuber size distribution should be incorporated into the plant growth submodel and an 'input data generation model' should be added to allow forecasts. Moreover parameters for other varieties than Bintje should be estimated and the model must be validated in various conditions of the seed production regions.

Relationship Between Epidemic Progress and Predictor Variables

The disease progress curves varied considerably from year to year (Fig. 7.5), suggesting that models which assume vector intensity to be constant (e.g. VANDERPLANK, 1963) are inadequate. Total aphid abundance in the suction trap (Fig. 7.5, S) was a bad predictor, implying that models assuming the rate of virus spread to be proportional to the number of aphids (e.g. MADDEN *et al.*, 1990) cannot explain the variance between years. If only vector species (A) were considered, a larger part of the variance could be explained. Surprisingly, weighting of vector numbers by their vector efficiency (*V* and *F*) did not reduce the variance. This result seems to contradict the fact that vector efficiency was a sensitive parameter. The correlation between vector abundance and vector abundance weighted by vector efficiency was apparently so high that the differences vector efficiency did not significantly affect the results. Hence we may ask ourselves, whether quantifying vector efficiency is really so important for predictions of virus spread as claimed by most authors (e.g. PETERS, 1987; DE BOKX, 1989). It might be sufficient to test whether a species can transmit a virus or not (vector capacity according to IRWIN & RUESINK, 1986) and to use the number of vectors for virus forecasts.

Even with the best predictors, a considerable amount of variance remained unexplained by the predictor variables used. Therefore other factors or a combination of the predictor variables used must account for the remaining variation. The disease incidence, used as dependent variable in the regression model, is not directly dependent on age resistance. Hence, the latter can be excluded as an important source of variance. The sensitivity analysis of the EPOVIR-model showed that changes in vector behaviour can affect PVY epidemics. Therefore a part of this variance can be attributed to vector behaviour.

8.2. Validation

In some situations the temporal behaviour of the EPOVIR-model deviated largely from the experimental data. Since attempts to improve the fit by changing one or two parameters have failed, probably a combination of several parameters or equations is responsible for the failure of the model in certain situations. Two model parts were highly sensitive, but the parameter estimates had poor experimental bases, namely the immigration/emigration behaviour of the vectors (parameter group D₁) and the plant-virus interaction (parameter group O₄). These parts should be revised using new experimental data.

Like other aphid and virus epidemic models, the EPOVIR-model assumes the concentration factor *s_c* (used to calculate the immigration rate from the suction trap catches) to be constant for all species. CUPERUS *et al.* (1988) questioned this assumption and postu-

lated than *M. persicae* lands more frequently in the field relative to suction trap catches, compared with other species. This assumption of the model should be tested by using new data sets on the landing rate on potato plants. In years, when vectors appear late, the model tends to underestimate, in years with early appearance to overestimate disease incidence and severity. RUESINK & IRWIN (1986) made the landing rate also proportional to the fraction of soil surface covered by the canopy (canopy closure), which reduces vector immigration early in the season. Such a correction could possibly improve the agreement of the model output with the experimental data. However, IRWIN & KAMPMELIER (1989) found that aphids land more frequently in fields with a low canopy closure than in fields with a high one. It should be verified experimentally, whether concentration factor s_c changes in function of the canopy closure. Better values for the concentration factor s_c and the emigration probability p_{em} could also be found by parameter identification. The deterministic infection submodel is better suited for parameter identification than the stochastic spatial version, due to the much shorter computation time.

The plant-virus interaction part of the EPOVIR-model should also be revised. In the experiments used for estimation of the former parameter, tuber infection was determined only at harvest and not during the growing season (Chapter 2.7.3). Hence the EPOVIR-model is limited in representing the dynamical evolution of tuber infection during the growing season. For optimizations of haulm-killing dates, intending to keep tuber infections low, the knowledge of the age resistance function is essential. The age resistance threshold t_{ary} should be identified using new artificial inoculation experimental data. It should be verified, whether age resistance is more pronounced for PVY_O than for PVY_N in the variety Bintje as found by BEEMSTER (1976), or whether there is no difference, as found by GIBSON (1991) for other varieties. Preliminary analyses have shown, than changing this parameter alone is not sufficient to obtain better agreement of the model results with the experimental data. Hence the model equations have to be analyzed. Since the PVY infections become systemic and early infections lead to far more tuber infections than late ones, the EPOVIR-model assumes that the time of the first infection determines how many tubers become infected (Eq. 6.22 & 6.25). In years with an important flight later in the season (mid-June and later), the sensitivity of the crop is probably lower in the model than in reality. To solve this problem, equations could be reformulated so that age resistance would reduce the probability of plant infection and not only the proportion of tubers becoming infected after a successful infection of a plant, as it is the case in the current version. Preliminary studies have shown that such a change together with a lower concentration factor s_c resp. a higher emigration probability p_{em} would give better agreement between model outputs and experimental data. Further, it is known that drought stress increases tuber infection, presumably by partially breaking the age resistance (WISLOCKA, 1982). Since the water stress is calculated as an output variable of the soil water balance submodel, it would be possible to calculate age resistance as a function of drought.

The delay m_i of approximately one week until the tubers become infected is probably too short in the EPOVIR-model (see Fig. 7.14, 1987o1, 1989o1 and 1990o1). SIGVALD (1986) assumed 22 days; other authors suggest that a delay of two to three weeks could be appropriate (BEEMSTER, 1976; BRABER *et al.*, 1982; BEEMSTER, 1987). The parameter m_i was estimated on experimental data obtained with one-stemmed plants

grown under laboratory conditions. Possibly virus translocation into tubers of other stems takes more time, so that a longer delay should have been assumed. The delay might also be a function of plant age, like in the model of SIGVALD (1986). The underestimation of m_t together with a probably too high age resistance (see above) could explain, why tuber infection hardly increased towards the date of haulm-killing in the EPOVIR-model, contrary to experimental results, showing that changing haulm-killing dates can affect tuber infection (KELLER, 1958).

The validation of the spatial pattern (Fig. 7.15 and 7.17) showed good agreement for the experiments, where also the temporal model behaviour was close to observed disease data. In situations, where the model overestimated the disease incidence and severity, the simulated spatial pattern also deviated from the observed one. As shown below, the spatial pattern largely depends on the disease incidence. The model tended further to underestimate the infections of the two neighbouring plants of the sources. This difference might be attributed to the effect of wingless aphids, ignored in the model. On the whole, the model seems to represent satisfactorily the spatial mechanisms of virus spread.

It has to be kept in mind that the experiments were conducted in small plots in fields of about 2 ha with a variable contamination by PVY. The exact incidence of PVY infection sources in the areas surrounding the experimental plot was unknown, and was only approximated by the incidence of PVY in the seed planted (p_{ib}). Thus the contribution of the infection sources surrounding the experimental plot might be responsible for a part of the deviation between model and experiments.

The lack of agreement between model outputs and experimental results limits the applicability of the current model version in practice. Hence in addition to the model adaptations mentioned in Chapter 8.1, the model parameters and equations describing the immigration/emigration behaviour and the plant-virus interaction should be changed in order to provide a model applicable in practice.

8.3. Structural and Sensitivity Analysis

By 'switching on and off' parts of the model, the structural analysis of model behaviour revealed important epidemiological processes on the one hand, on the other hand also processes that could be ignored in epidemic models of nonpersistent virus diseases.

The sensitivity analysis allowed to find the parameters most influencing the model output variables. Parameter sensitivities were estimated with only six values of each parameter (see Eq. 2.9). Nevertheless, the choice of the values in the plausibility interval should ensure the representativity of the results for the whole interval.

The response of the model outputs to most parameters was almost linear on the logit scale. Therefore a linear regression could give a measure allowing comparisons of parameter sensitivities. However, changes in some parameters provoked nonlinear responses of the model outputs. In these cases the regression coefficient usually underestimated the sensitivity of the respective parameters in certain parts of the interval and gave only an average sensitivity for the whole interval. Therefore it is necessary to consider also the response curves of the output variables (Fig. 7.19 to 7.27).

Spatial Pattern

The aggregation of infections around the infection sources slowed down the temporal epidemic progress compared with the uniform distribution (Fig. 7.6, see also Chapter 4). The effect of clumping on the epidemics can be explained as follows: nearly all neighbouring plants of infection sources are healthy at crop emergence, therefore most inoculations lead to new infections. Later, most neighbouring plants are already infected and the number of new infections decreases, while it remains high if the inoculations follow a uniform distribution. This effect of clumping becomes progressively important during the growing season. However, if the initial disease incidence was 0.01 or higher, a saturation was reached at t_2 or t_3 , since for both distributions of the inoculations the disease incidence approached 1, and consequently tuber infection could no longer increase. Initial disease incidences as low as 0.001 are not uncommon in seed potato production. Models different from EPOVIR assuming a random distribution of the inoculations seem to be inadequate for such cases, since they could result in a considerable overestimation of disease incidence and severity.

The temporal virus spread appears to be determined mainly by flying vectors, whereas mainly walking vectors were responsible for the spatial pattern (Fig. 7.9). The flight and walking distances had little effect on the spatial pattern (Fig. 7.18), but the ratio of infections caused by walking and flying vectors was relevant (Fig. 7.9).

A comparison of sensitivities of the disease severity and of the spatial pattern shows that those parameters that determined disease severity also determined the spatial pattern to a large extent. ZADOKS & KAMPMEIJER (1977) found that disease gradients around infection sources became flatter with increasing disease incidences. The spatial pattern thus appears to be largely dependent on the disease incidence. Parameters leading to higher disease incidences tend also to produce higher evenness values.

Parameters Related to Vector Behaviour

The results in Chapter 7.4.1 suggest that the naturally occurring behavioural differences between colonizers and noncolonizers are relevant for virus epidemics. Nevertheless the sensitivity analysis of the EPOVIR-model showed that the behaviour sequence parameters were among the less sensitive ones. Contrary to the hypothesis of ROCHOW (1974) and HARREWIJN (1989), parameters determining vector abundance were more sensitive than behaviour sequence parameters. Note that the vector abundance in the model is the abundance of unsettled winged aphids. Settled aphids probably contribute little to PVY epidemics and so do wingless aphids (see below). Changing the abundance of settled or wingless aphids will therefore change PVY epidemics little.

Although vector abundance, vector propensity and inoculation sequence frequency (a function of the behaviour sequences) are mathematically equivalent factors in Eqs. 6.1 to 6.4, changes in some parameters determining vector abundance and propensity were more important for the epidemic than changes in behaviour sequences. The epidemic appears thus to be more buffered against changes in vector behaviour than against changes in vector abundance or propensity. This phenomenon is partly due to the compensation of changes in the frequency or duration of one behaviour state by another state and partly to the link between flight frequency and emigration rate (Eq. 6.6): higher flight frequency

leads to more inoculations (Eq. 6.1 & 6.9), but also to a higher emigration rate (Eq. 6.5 & 6.6) and consequently to lower abundance.

It is remarkable that those parameters, influencing the behaviour sequences most, were also the most important for virus spread, namely k_{fw} , k_{fl} and k_{dl} (Fig. 5.7, 5.8 & 7.18). The output variables responded nonlinearly to the scaling factors of flight and probing frequency (k_{ff} and k_{fp}). An increase of these parameters had no visible effect, whereas their reduction slowed down virus spread. To control PVY spread effectively, k_{ff} resp. k_{fp} must be substantially reduced. On the other hand, there is little danger of favouring virus dissemination by an increased probing frequency (Fig. 7.9). The probability to probe on a plant p_{piC} , which is a function of probing frequency, was close to 1 and thus not a limiting factor in the epidemic process.

The studies of aphid behaviour in the 50es and 60es led to the conclusion that the frequent movements from plant to plant and frequent test probes on host and nonhost plants during host selection are nearly optimal for the transmission of nonpersistent viruses (KENNEDY *et al.*, 1959). SWENSON (1968) stated that 'the nondiscriminatory alighting behaviour on hosts and nonhosts, the dominance of dispersal over host-finding, and the intensity and duration of aphid migration could hardly be improved upon to facilitate the spread of stylet-borne viruses' and cited KENNEDY (1960), who said that 'aphids now appear to be so ideally fitted behaviourally for virus spreading in the field that one is prompted to ask, not why they are so successful as vectors but why they are not more successful, why they do not prevent us from raising any healthy crops at all'. It was hypothesized that the behaviour of aphids is the main reason, why this taxonomic group is so important for plant virus spread. KENNEDY (1951) put forward the hypothesis that virus transmission and infection could have a positive feedback on the vectors. Many virus infected host plants have higher nitrogen concentrations in the phloem sap than noninfected hosts (DORSCHNER, 1990), leading to higher fecundity of the aphids. Thus colonizing aphids could profit from transmitting a virus disease to their host plants. Kennedy suggested that aphids could also profit from virus transmission to nonhost plants (KENNEDY, 1951; KENNEDY *et al.*, 1959), arguing that aphids could colonize virus infected plants of a species, which was previously a nonhost. Successively they could adapt to the new host and colonize also healthy plants. Virus diseases could therefore allow an aphid species to extend its host range. This hypothesis is supported by the fact that plant resistance can be broken by increasing amino acid concentrations in the phloem (HARREWIJN, 1986), a phenomenon often occurring in virus infected plants. The sensitivity analysis of the EPOVIR-model only partially confirmed the hypothesis that vector behaviour is optimal for nonpersistent virus transmission for the PVY-potato pathosystem. On the one hand k_{ff} and k_{fp} and therefore flight and probing frequencies seem to have nearly optimal values for virus epidemics (Fig. 7.19). An increase of these parameters did not affect virus spread, whereas a decrease reduced it. On the other hand not all behaviour sequence parameters seem to be at an optimum for PVY spread, since disease severity was enhanced by increasing the walking frequency or by reducing the duration of 'long penetrations'. Further only the flight frequency of noncolonizing species seems to be optimal for virus dissemination, whereas the much lower flight frequency of colonizing species seems to be suboptimal (Fig. 7.21).

The results indicate also that the hypothesis that restlessness accelerates virus spread (ROMANOW, 1985; CARTER & HARRINGTON, 1991), has only limited applicability to PVY epidemics. Increasing walking frequency and decreasing 'long penetration' duration led to an increase in virus dissemination; increasing the probing and flight frequencies of the noncolonizers did not affect the epidemic of PVY at all.

Among the dispersal parameters the concentration factor s_C , the emigration probability p_{em} and the temperature threshold for take-off f_{th} were of outstanding sensitivity (Fig. 7.8, 7.10 & 7.18). The high sensitivity of immigration/emigration parameters indicates that the abundance of unsettled vectors in the field is an important epidemiological factor. The estimations of s_C and p_{em} were based on literature (TAYLOR & PALMER, 1972; CARTER *et al.*, 1982) and on experiments (Chapter 5). Considering the wide range of possible deposition rates given by TAYLOR & PALMER (1972), the value chosen is to a certain extent arbitrary. Despite the research that has already been carried out on the migration and dispersal behaviour of aphids (KLINGAUF, 1987; ROBERT, 1987), many aspects need further studies to enable progress in virus epidemic research. The take-off thresholds for temperature and wind speed restricted virus spread (Fig. 7.10). The model was sensitive mainly to increases of the temperature threshold for take-off, whereas its decreases had a small effect (Fig. 7.23). This explains, why temperature restricted take-off to a small extent only (Fig. 7.10), despite its high sensitivity. The precipitation threshold, which had no visible effect on virus epidemics, could be omitted from the model. The settling rate had only a minor effect on model behaviour (Fig. 7.8 & 7.18), probably because it affects only colonizing species, and could thus be ignored.

Fig. 7.18 shows that the dispersal parameters, determining the kind and frequency of virus transmissions by walking aphids, were of minor importance for model behaviour. These model parameters were partly hypothetical, based e.g. on the geometrical relationships and distances between plants (Fig. 6.6). The low sensitivity suggests that these rough estimates were probably sufficient for this study. Although mainly walking aphids were responsible for the nonrandom spatial pattern (Fig. 7.9), the evenness of the distance class distribution responded little to changes in walking parameters.

The parameter κ of the von Mises distribution describing the flight directions had almost no effect on the temporal virus spread and was also irrelevant for the spatial pattern. For such low values of κ , the simpler circular uniform distribution could be assumed instead of the von Mises distribution, which would also make simulations more efficient. Note that the observed tuber infection patterns showed no directional tendency as well (Fig. 7.17). This seems to contradict published results, reporting marked differences between upwind and downwind virus spread in situations with a prevailing wind direction (HAMPTON, 1967; THRESH, 1976; IRWIN, 1981). Such a prevailing wind direction existed also in the experimental fields during the periods suitable for aphid flight (Fig. 5.10). The apparent contradiction could be explained by the distance at which wind direction plays an important role. The reports cited above concern experiments with a single focus of initial sources in the field and have quantified virus spread over distances of tenths and hundreds of meters. In seed potato-fields the mean distance between source plants does not exceed a few meters. It seems that over so short distances no clear directional tendency can be detected, which is confirmed by experiments of GREGORY & READ (1949) for potato viruses. It can be concluded that the influence of wind direction

is relevant over longer distances, e.g. transmission between fields, but negligible for shorter distances.

Other Parameters

Among the other parameters not related to vector behaviour, the vector efficiency and the age resistance threshold of the plant were the most sensitive. Vector efficiency parameters have already been estimated for many aphid species and for different virus strains (PETERS, 1987; DE BOEK & PIRON, 1990). Despite the high variance in the data used for the estimation of vector propensity parameters (Fig. 6.5) the differences between the vector groups were clearly visible. Fig. 7.5 suggests that the number of vectors was as good epidemic predictor as the number weighted by vector efficiency, due to the high correlation between these two variables. The data on vector propensity available by now seem thus to be sufficient for model studies and forecasting.

Natural mortality restricted PVY spread (Fig. 7.8), but the sensitivity to the mortality rate only moderate (Fig. 7.18). Since this parameter can be considerably increased by antagonists or aphicides, its effect can nevertheless be high (Fig. 7.25 & 7.32).

The proportion of immigrating aphids coming from sources v_{im} had a very low regression coefficient in the sensitivity analysis, because of its low standard value. Nevertheless, v_{im} -values exceeding 5 % could considerably increase PVY spread (Fig. 7.11 & 7.26). In most cases, v_{im} will probably be lower than 1 % and thus hardly affect the epidemics. However, if highly contaminated fields are present in the proximity of a seed potato field, v_{im} could have much higher values. The recommendation to avoid ware potato fields with a high virus contamination nearby seed potato fields (HARREWULN *et al.*, 1981) seems to be justified.

Although the model was only moderately sensitive to the initial disease incidence p_i (Fig. 7.18), this parameter is important in practice, since it is highly variable and the effect of large changes in p_i can be considerable (Fig. 7.27).

Ignoring serial inoculations led to a decrease in disease severity, whereas the retention rate hardly changed it (Fig. 7.11). The small effect of the latter parameter indicates that probing was so frequent that the loss of infectivity in-between probes could be ignored by setting the parameter to 0.

The plant-virus interaction seems to be highly relevant for PVY epidemics (Fig. 7.12 & 7.18), the age resistance thresholds and the latent period were the most sensitive parameters. By the time when alatae abundance becomes high, the susceptibility of the plants to virus infections is low. This seems to be the normal case under Swiss conditions, but in some years aphid flight can occur very early, when no age resistance could yet be developed (see Fig. 7.14, 1989 and 1990). Due to age resistance, most PVY infections occur early in the season (CADMAN & CHAMBERS, 1960; WOODFORD, 1976). The latent period m_l led also to a significant reduction of tuber infection, whereas the delay to tuber infection m_t affected disease severity only during the early phases of the epidemic, but not the final result. Note that higher variance of the latent period favoured PVY dissemination. Most epidemic models assume a constant latent period (e.g. VANDERPLANK, 1963; RUESINK & IRWIN, 1986) and only few take into account its variance (e.g. SIGVALD, 1986). The earliest models of plant disease epidemics were constructed for and applied to fungal pathogens (VANDERPLANK, 1963; ZADOKS & SCHEIN, 1979).

For non-systemic fungal pathogens, the latent period has usually a relatively small variance (ZADOKS & SCHEIN, 1979). Most plant virus diseases infect their hosts systemically and show a high variability of the latent period (BEESTER, 1979; GUGERLI, 1979). The simulation results of the EPOVIR-model suggest that the variance of the latent period considerably influences the rate of virus spread, and should therefore be included in virus epidemic models.

8.4. The Role of Vector Groups and Aphid Morphs

The simulation results suggest that the contribution of colonizing species to PVY transmissions was low (Fig. 7.28) and that *M. persicae* (vector group 1, Tab. 6.1) was the most important among these species. The most relevant vector species belonged to group 2b, composed of *A. fabae*, *B. helichrysi* and *P. humuli*. Results obtained in Sweden indicate that virus spread was little sensitive to changes in the vector efficiency of *M. persicae* (SIGVALD, 1986). HARRINGTON *et al.* (1986) estimated for Southern England that *M. persicae* was responsible for 12 % of the PVY transmissions and all colonizing species together for only 15 %. Different results were obtained by CUPERUS *et al.* (1988) in the Netherlands. CUPERUS *et al.* (1988) argue that the Taylor suction trap (providing input data for the EPOVIR-model) and the yellow water trap (providing input data for the Sigvald-model) tend to underestimate early immigration of *M. persicae* in the field. Hence it is possible that the EPOVIR-model underestimated *M. persicae* abundance in the field and its contribution to PVY epidemics. On the other hand CUPERUS *et al.* (1988) probably overestimated the importance of *M. persicae*. The authors did not take the different behaviour of colonizing and noncolonizing species into account (see Fig. 7.29). Furthermore, they did not consider the age resistance of the potato crop. The spring flight of *M. persicae* usually occurs later than that of some noncolonizing species, which limits its potential contribution. Taking these facts into account, it is likely that noncolonizing aphids were responsible for most PVY spread even if their contribution might be overestimated by the EPOVIR-model. Inclusion of apterous aphids in the model would hardly increase the colonizer's contribution (Fig. 7.30).

The hypothetical simulation experiments (Fig. 7.29) indicate that noncolonizing species transmitted PVY more frequently. This difference must be attributed to their higher activity. *A. fabae* spent a higher proportion of time walking and had a higher flight and probing frequency (Fig. 5.6). These behaviour characteristics were positively correlated with disease spread (Fig. 7.19 to 7.21), at least in certain ranges of parameter values. The inoculation sequence frequency (see Eq. 6.1 to 6.4) was therefore higher for the noncolonizing species than for colonizing ones. If only the behaviour sequence is considered, noncolonizing aphids seem to be about twice as efficient in transmitting PVY than colonizing ones. Note that the walking frequency, the frequency of behaviour state changes and the proportion of time not spent in state 'long penetration' showed differences in the same order of magnitude, i.e. the corresponding values were about twice as high for noncolonizing than for colonizing species.

The role of wingless morphs in PVY epidemics was found to be relatively irrelevant, even in cases when high populations of wingless aphids would develop on the plants. BROADBENT & TINSLEY (1951) found that PVY incidence was reduced by only

17 % in plots, where sticky barriers prevented apterae migration. The potential contribution of apterae estimated for the standard run of the EPOVIR-model was even lower, but it could be higher in some years. Wingless aphids can infect only the neighbouring plants of the infection sources, which are likely to be infected already by winged aphids, before wingless adults develop in the field. Therefore the potential contribution of the wingless aphids is limited. It seems that it was justified to ignore the effect of wingless aphids for PVY epidemics in the EPOVIR-model. However, the validation of spatial PVY spread (Fig. 7.15 and 7.17) suggest that the infection of the two plants in the row adjacent to a source was underestimated by the model. This difference could be attributed to the wingless aphids.

8.5. Assessment of Control Measures Affecting Winged Aphid Behaviour

The scenarios simulate the effects of measures applied to single fields. Certain parameters, especially the emigration probability, are likely to affect also virus the incidence in neighbouring fields. Thus increases in the emigration probability can reduce disease incidence in the field under consideration, but simultaneously increase infection in neighbouring fields, which have not been treated in the same way. Moreover, application of control measures to larger areas could lead to results, which are different from those, which were obtained in the simulation experiments for single fields.

Reduced Immigration

Reduced immigration into a field (Fig. 7.31) seems to be the most promising method of virus control of all the methods studied. Virus infections could be substantially lowered by reducing the number of vectors through reflective or white surfaces (LOEBENSTEIN & RACCAH, 1980; GIBSON & GUNENC, 1981; GIBSON & RICE, 1989; KUROLI & ERDÉLYI, 1990). Since the EPOVIR-model is little sensitive to changes in the proportion of immigrants coming from sources, the effect must be attributed mainly to a reduction of virus transmission between plants in the field and not to a reduction of infections from sources outside the field. No reports exist that reflective surfaces had led to increased virus spread. Hence, contrary to the other scenarios, this method seems to bear no risk to increase the disease incidence.

Aphicides

Aphicide treatments are reported to decrease or increase the incidence of nonpersistent viruses (SHANKS & CHAPMAN, 1965; GABRIEL *et al.*, 1981; RICE *et al.*, 1983; SASSEN, 1983; GIBSON & RICE, 1989). A reduced disease incidence is explained by increased mortality or changed vector behaviour, an increased disease incidence by changed vector behaviour or the reduction of antagonist populations, subsequently favouring the build-up of large aphid populations (GIBSON & RICE, 1989). The results of the 'aphicide' scenarios (Fig. 7.32) confirm that aphicides can increase mortality in a way substantially reducing PVY dissemination. Further it was shown that changes induced in vector behaviour can lead to lower or higher PVY transmission rates.

In the 'aphicide' scenarios, PVY spread was favoured by increased flight and walking frequencies, by a decreased emigration probability, by increased mortality rates and by reduced probing time. The last effect seems to be surprising since k_{dI} was negatively correlated with the disease severity in the sensitivity analysis (Fig. 7.20). By adjusting parameter k_{fp} , probing frequency was kept equal to the standard run, therefore the observed effect on PVY epidemics cannot be due to a changed probing frequency. Inspection of the modified behaviour sequences showed that flight frequency was increased more than walking frequency, the resulting reduction in disease severity was probably due to an increased emigration rate. In experiment P1 (short probing durations), epidemic progress was hardly affected by the increased flight and walking frequencies, probably because their values were already very high.

The results of these scenarios highlight the complexity of the potential effects of aphicide treatments. Certain insecticides could be applied in order to reduce PVY spread, provided they do not have adverse effects on vector behaviour. However, this seems not to be sensible, since a high application frequency would be needed to ensure an effective reduction of PVY dissemination, which would cause high costs, endanger the environment and reduce antagonist populations. Moreover, since our knowledge of the effects of most insecticides on aphid behaviour is very limited, their application bears the risk to increase PVY spread.

Repellents

Repellents could potentially reduce nonpersistent virus dissemination by increasing the emigration probability from the field. Carboxylic acids reduce aphid colonization of treated plants (SHERWOOD *et al.*, 1981; GIBSON *et al.*, 1982; HERRBACH, 1987) and cause earlier departure from treated surfaces (PHELAN & MILLER, 1982). Reduced settling is accompanied by a reduction in mean penetration time (PHELAN & MILLER, 1982). In the 'repellents' scenarios (Fig. 7.33), this effect caused a considerable increase in PVY spread, in accordance with results of GIBSON *et al.* (1982), showing that treatment with dodecanoid acid can increase PVY transmission in laboratory tests. The risk to favour PVY spread by carboxylic acids seems to be high. An alternative to carboxylic acids could be polygodial, which reduced PVY transmission in laboratory tests (GIBSON *et al.*, 1982). This substance was not included in the scenarios, since its effects on aphid behaviour are hardly known.

Plant Resistance to Aphids

In a review KENNEDY (1976) found that virus dissemination can be increased or decreased in aphid-resistant crops. GRAY *et al.* (1986a) found slightly lower incidence of water melon mosaic virus (WMV) in an aphid-resistant water melon variety than in a susceptible one. Simultaneously, the spatial pattern of infections showed less clumping in the resistant variety. Simulations of the 'plant resistance' scenarios showed that the effect of the resistance depends on the balance of increased emigration, increased dispersal and changes in probing frequency. Higher dispersal frequency, especially flight frequency, produced less clumped spatial patterns (Fig. 7.34), consistently with the results of GRAY *et al.* (1986a), but led to higher disease severities.

The different types of plant resistance are likely to have different effects on PVY epidemics. Antixenosis could reduce PVY spread, provided its effect on emigration is stronger than that on dispersal. Antibiosis leading to higher mortality is likely to have little effect on nonpersistent virus spread, since the mortality rate was a parameter of only moderate sensitivity. Vector resistance leading to lower probing frequencies could decrease disease severity. On the other hand, there seems to be little danger to favour virus spread by increasing probing frequency, because its value was already near an optimum.

Plant resistance can change a host plant species into a nonhost (AUCLAIR, 1989); colonizing species may become thus noncolonizers and react by increased restlessness. Certain parameter constellations in the scenario simulate this higher restlessness. It was assumed in the scenario that plant resistance acts similarly on the behaviour of colonizing and noncolonizing aphids. If the noncolonizing species would not change their behaviour simultaneously, the effect would be smaller than the one found in the scenario, since non-colonizers were responsible for most transmissions.

Alarm Pheromones

Experimental alarm pheromone applications could in certain cases reduce virus spread (DAWSON, *et al.*, 1982; GIBSON *et al.*, 1984) in other not (YANG & ZETTLER, 1975; HILLE RIS LAMBERS & SCHEPERS, 1978). The lack of effect could also be due to the low stability of the compounds (HERRBACH, 1985). Simulations of the 'alarm pheromone' scenarios (Fig. 7.35) showed that increased emigration by the action of alarm pheromones could be useful in reducing virus dissemination. On the other hand alarm pheromones led to reduced settling coupled with a reduction in penetration time (PHELAN & MILLER, 1982), which resulted in higher PVY spread. The balance of these two effects will determine, whether disease severity will be higher or lower after treatment with alarm pheromones.

8.6. Modelling and Simulation Tools

The interactive simulation features, offered by ModelWorks (FISCHLIN *et al.*, 1990) proved very useful in the first explorative phase of the modelling and simulation studies. Parameters and initial values could be easily changed, and models could be installed in and removed from the PAV simulation system interactively according to needs. After the first interactive phase, batch simulation became more important, due to high time requirements of the simulations. Since the simulation environment can also be controlled from the client's program, the simulation experiments could be programmed and the simulation experiments were executed overnight or on remote simulation servers.

The programming language Modula-2 (WIRTH, 1985) proved particularly suited for modular modelling. It allows to implement models as independent units, with a defined interface to other models. The structure of large models can be kept clear and tractable. In the PAV simulation system several models could be installed, partly composed of submodels. Some of these submodels, e.g. the plant growth submodel, are used by several models. Such a structure can hardly be implemented in a language, which does not offer possibilities of modularization.

Easy access to the DialogMachine (FISCHLIN, 1986; FISCHLIN & SCHAFELBERGER, 1987; KELLER, 1989) and to the programming language Modula-2 from ModelWorks revealed to be very useful. It allowed to use cellular automatons, a model formalism, which is not supported by ModelWorks, further to program statistical analyses and graphical outputs easily.

9. CONCLUSIONS

Vector Behaviour in the Epidemiology of Nonpersistently Transmitted Viruses

Movement and probing/feeding of the vectors are absolutely indispensable activities for the spread of vector borne viruses. The fact that aphids move and probe/feed so frequently during host selection was hypothesized to be responsible for their importance in virus transmission (SWENSON, 1968). The results of this study showed that naturally occurring differences in behaviour, changes in certain dispersal parameters and large changes in certain behaviour sequence parameters are relevant for the epidemiology of nonpersistently transmitted plant viruses, but that the system is more sensitive to proportional changes in some other parameters than to changes in behaviour sequence parameters.

Parameters determining the behaviour sequences of the vectors were among the less sensitive ones in PVY epidemics. They were particularly less sensitive than some parameters influencing the vector abundance, like the relative immigration rate and the emigration probability. The hypothesis that parameters determining movement and probing are the most relevant ones for aphid-borne virus epidemics and more sensitive than the parameters determining the vector abundance (ROCHOW, 1974; HARREWIJN, 1989) can be rejected for the PVY-potato system. The scaling factors of walking frequency and those of 'long penetration' duration influenced the characteristics of the behaviour sequences most, followed by the scaling factors of flight and probing frequencies. These parameters influenced also PVY epidemics to a certain extent. The study showed that the walking frequency and the duration of penetrations could be used to quantify and to compare the 'behavioural efficiency' of populations to transmit nonpersistent virus diseases; high walking frequency and short penetration times indicating a high potential virus transmission rate, low walking frequency and long durations of penetrations, typical for settled aphids, indicating a low potential virus transmission rate. Parameters determining alatae walking were little sensitive, whereas the temperature and wind speed flight thresholds, influencing the flight frequency, were important.

The hypothesis that 'restlessness' leads to higher virus transmission rates (KENNEDY, 1976; ROMANOW, 1985; CARTER & HARRINGTON, 1991) cannot be generally accepted for the nonpersistently transmitted PVY. Increased walking frequency and decreased 'long penetration' frequency or duration led to higher virus transmission rates. On the other hand higher flight and probing frequencies did not result in more PVY spread, whereas lower values of these parameters led to lower virus transmission rates. These behaviour characteristics were at an optimum, thus partly supporting the hypothesis that aphids have nearly optimal behaviour for nonpersistent virus spread (KENNEDY *et al.*, 1959; KENNEDY, 1960; SWENSON, 1968).

The experimental behaviour studies (Chapter 5) have confirmed that noncolonizing aphids are more 'restless' than colonizing ones (KENNEDY *et al.*, 1959; MCLEAN & KINSEY, 1968; PETERS *et al.*, 1990). Despite the low or moderate sensitivity of behaviour sequence parameters, these naturally occurring differences in behaviour were

large enough to produce marked differences in PVY epidemics. Noncolonizing aphids were about twice as efficient PVY vectors, if only their behaviour is considered. Due to these differences in behaviour, but also to the higher abundance of noncolonizing species, the latter seem to be responsible for most PVY infections. Wingless aphids seem to contribute little to PVY spread, probably because their dispersal and thus virus transmission is limited to the neighbouring plants.

Range of Validity of the Results

Although differences exist in climate, genotypes and the plant-virus interaction in different parts of the world (BERTSCHINGER, 1992), the importance of aphid vectors in PVY transmission, the composition of the vector complex (TURL & MACDONALD, 1987; VAN HARTEN, 1983; SIGVALD, 1989), the transmission process of PVY and the aphid-plant interaction are similar in PVY-potato pathosystems of all regions. Hence the role of vector behaviour in PVY epidemiology as well as the contribution of colonizers and non-colonizers to the epidemic is likely to be similar in all these systems.

Many pathosystems of nonpersistent viruses in important world crops have similar characteristics as the PVY-potato pathosystem under study. In most systems noncolonizing species play an important role. Differences in behaviour between colonizing and noncolonizing species like those observed in this study were also found in other systems (see e.g. MCLEAN & KINSEY, 1968). Other phenomena like age resistance occur in most systems, e.g. the soya-SMV system (RUESINK & IRWIN, 1986). Thus vector behaviour is likely to play a similar role in other pathosystems of nonpersistently transmitted virus diseases. This applies also to cases, where the disease is not seed transmitted.

On the other hand, a generalization of the results to semipersistently or persistently transmitted diseases seems difficult, because the sequence of behavioural events necessary for these kinds of transmission is different (SWENSON, 1968; POWER, 1990; CARTER & HARRINGTON, 1991). The transmission of these diseases requires longer penetration times than that of nonpersistent viruses and in the case of persistent transmission also a latent period of the virus in the vector. To transmit the disease in the field, an aphid must thus stay much longer in the field.

Implications for Epidemic Models

For future modelling vector borne virus epidemics, the concept of inoculation and infection submodels should be applied, since it enhances the clarity and flexibility of the models.

Naturally occurring differences in behaviour between colonizing and noncolonizing species should be included in epidemic models of nonpersistent virus diseases. As an approximation, it can be assumed that noncolonizing species transmit PVY about twice as frequently than colonizing ones. Differences between the species within the groups of colonizing and noncolonizing aphids were much smaller than those found between the groups. As a further refinement, such differences could also be taken into account. It seems not necessary to include wingless aphids in epidemic models of nonpersistent viruses, because of their marginal importance.

The dependence of flight on temperature and wind speed should also be included in virus epidemic models. The flight thresholds given by different authors vary widely

(KRING, 1972; ROBERT, 1987), and depend on the aphid species and on the season. This area would deserve further research to provide a better base for forecasting of virus epidemics. The simulation studies indicated also an urgent need for further research on the migration behaviour of aphids.

An epidemic model assuming a spatially random mechanism of virus spread would overestimate the disease severity by up to 25 %. The overestimation will be largest, when the initial disease incidence is low, which is often the case in seed potatoes. Thus the non-random spatial pattern should be taken into account in virus epidemic models. The mean flight and walking distances are extremely difficult to estimate; fortunately these parameters were insensitive.

The EPOVIR-model is not yet applicable in practice. The parts immigration/emigration behaviour and plant-virus interaction should be revised, based on new experimental data. Further it should be validated in different regions and for other varieties.

Implications for Virus Control

The sensitivity analysis confirmed that several control measures, partly already applied in practice, are useful for PVY control in potatoes. Among them are the reduction of vector efficiency by application of mineral oils, pre-sprouting and early planting, which ensures an early development of age resistance, planting seed which is little contaminated with PVY and early roguing. The high sensitivity of the temperature and wind speed thresholds emphasizes the importance to concentrate seed potato production in cool and windy areas. In addition to the meteorological effects on aphid flight, these regions tend to have low aphid abundances, presumably because they are avoided by winged migrants. Such effects will lead to substantially reduced vector abundance and flight activity, thus allowing to produce seed with a low virus contamination.

Considering the comparatively low sensitivity of the behaviour sequence parameters, it seems to be more promising to reduce the number of alatae in the field, than to manipulate vector behaviour in order to reduce the flight, walking or probing frequencies or to increase settling. The consequences of behaviour modifications are very difficult to assess. Highest reductions of virus incidence can be achieved through a reduction of immigration by reflective or white surfaces or an increase of emigration. The latter can be done by repellents, alarm pheromones or plant resistance. Some aphicides have also repellent effects. All these techniques are likely to influence also the behaviour sequence of the vectors, which can potentially increase virus spread. Therefore these control measures should not be recommended.

REFERENCES

- Anderson, R.M., 1981. *Population ecology of infectious disease agents*, In: May, R. (ed.), Theoretical ecology, Blackwell, Scientific publications, Oxford, a.o., 2nd ed., 318-355.
- Anderson, T.W. & Goodman, L.A., 1957. *Statistical inference about Markov chains*. Ann. Math. Stat., 28: 89-110.
- Apple Computer, Inc., 1988. *APPLE Numerics Manual*. Addison-Wesley, 2nd edition, 336pp.
- Auclair, J.L., 1989. *Host plant resistance*. In: Minks, A.K. & Harrewijn, P. (eds.), *Aphids - their biology, natural enemies and control*, Elsevier, Amsterdam, a.o., 2C: 225-265.
- Bagnall, R.H., 1991. *Cyclic epidemics of aphid-borne viruses in the Northern seed-potato-growing areas*. In: Harris, K.F. (ed.), Advances in disease vector research, Vol. 7, Springer, New York, a.o., 53-71.
- Bailey, N.T.J., 1975. *The mathematical theory of infectious disease*. Hafner, New York, NY.
- Batschelet, E., 1981. *Circular statistics in biology*. Academic Press, London, a.o., 371pp.
- Beemster, A.B.R., 1969. *Virus translocation in two-stemmed potato plants*. Eur. Pot. J., 12: 20-25.
- Beemster, A.B.R., 1976. *Translocation of potato viruses Y(N) and Y(O) in some potato varieties*. Pol. Res., 19: 169-172.
- Beemster, A.B.R., 1979. *Acquisition of potato virus Yn by Myzus persicae from primarily infected potato plants*. Neth. J. Pl. Path., 85: 71-81.
- Beemster, A.B.R., 1987. *Virus translocation in potato plants and mature plant resistance*. In: De Bokx, J.A. & van der Want, J.P.H. (eds.): *Viruses of potatoes and seed-potato production*. 2nd ed., 116-125.
- Beemster, A.B.R. & de Bokx, J.A., 1987. *Survey of properties and symptoms*. In: De Bokx, J.A. & van der Want, J.P.H. (eds.): *Viruses of potatoes and seed-potato production*. PUDOC, Wageningen, 2nd ed., 84-116.
- Berger, P.H. & Ferriss, R.S., 1989. *Mechanisms of arthropod transmission of plant viruses - implications for the spread of disease*. In: Jeger, M.J. (ed.), *Spatial components of plant disease epidemiology*, Prentice Hall, Englewood Cliffs, NJ, 40-84.
- Bertschinger, L., 1992. *Modelling of potato virus pathosystems by means of quantitative epidemiology: an exemplary case based on virus degeneration studies in Peru*. Diss. ETH Zürich No. 9759, 111pp.
- Best, D.J. & Fisher, N.I., 1979. *Efficient simulation of the von Mises distribution*. Appl. Stat., 28: 152-157.
- Bishop, Y.M.M., Fienberg, S.E. & Holland, P.W., 1975. *Discrete multivariate analysis*. MIT Press, Cambridge, a.o., 557pp.
- Blackman, R.L., 1990. *Specificity in aphid/plant interactions, with particular attention to the role of the alate colonizers*. In: Campbell, R.K. & Eikenbary, R.D., *Aphid-plant genotype interactions*, Elsevier Publ., Amsterdam, 251-274.
- Boiteau, G. & Singh, R.P., 1991. *Difference in the transmission of PVY \circ to potato and tobacco by the potato aphid, Macrostomum euphorbiae (THOMAS): Effects of probing duration and frequency*. Can. Ent., 123: 869-874.
- Bos, L., 1983. *Introduction to plant virology*. PUDOC, Wageningen, 160pp.
- Bouchery, Y., 1987. *Influence de la vitesse du vent sur les captures de pucerons au piège à succion de 12 m*. In: Cavalloro, R. (ed.), *Aphid migration and forecasting "Euraphid" systems in European countries*, 113-121.
- Box, G. & Muller, M., 1958. *A note on the generation of normal deviates*. Ann. Math. Stat., 28: 610.
- Brabec, J.M., Bus, C.B. & Schepers, A., 1982. *Changes in leaf components and peroxidase activity of potato plants (cv. Bintje) in relation to mature-plant resistance to PVY \circ* . Pot. Res., 25: 141-153.
- Bradley, R.H.E. & Rideout, D.W., 1953. *Comparative transmission of potato virus Y by four aphid species that infest potato*. Can. J. Zool., 31: 333-341.
- Broadbent, L., 1950. *The correlation of aphid numbers with the spread of leafroll and rugose mosaic in potato crops*. Ann. appl. Biol., 37: 58-65.
- Broadbent, L. & Tinsley, T.W., 1951. *Experiments on the colonization of potato plants by apterous and by alatae aphids in relation to the spread of virus diseases*. Ann. appl. Biol., 38: 411-424.
- Cadman, C.H. & Chambers, J., 1960. *Factors affecting the spread of aphid-borne viruses in potato in Eastern Scotland. III. Effect of planting date, rousing and age of crop on the spread of potato leaf-roll and Y viruses*. Ann. appl. Biol., 48: 729-738.

- Cambrezy, C., Augier de Montgremier, H. & Larroque, N., 1981. *Migration du virus Y. La Pomme de terre française*. 407: 333-341.
- Campbell, C.L. & Madden, L.V., 1990. *Introduction to plant disease epidemiology*. Wiley, New York, a.o., 532pp.
- Cane, V.R., 1978. *On fitting low-order markov chains to behaviour sequences*. Anim. Behav., 26: 332-338.
- Carter, N. & Harrington, R., 1991. *Factors influencing aphid population dynamics and behaviour and the consequences for virus spread*. In: Harris, K.F. (ed.), *Advances in disease vector research*, Vol. 7, Springer, New York, a.o., 19-51.
- Carter, N., Dixon, A.F.G. & Rabbinge, R., 1982. *Cereal Aphid Populations: Biology, Simulation and Prediction*. PUDOC, Wageningen, 91pp.
- Cuperus, C., Prinen, J.D., Piron, P.G.M., van Rheezen, B., Harrewijn, P., & de Bokx, J.A., 1988. *Monitoring of aphids and spread of PVYn and PLRV in a seed-potato crop in the Netherlands*. Med. Fac. Landbouw. Rijksuniv. Gent, 53: 499-505.
- Davidson, W.D., 1928. *A review of literature dealing with the degeneration of varieties of the potato*. Econ. Proc. Roy. Dublin Soc., 2: 331-389.
- Dawson, G.W., Griffiths, D.C., Pickett, J.A., Smith, M.C. & Woodcock, C.M., 1982. *Aphid alarm pheromone derivatives affecting settling and the transmission of plant viruses*. J. chem. Ecol., 8: 1377-1388.
- De Bokx, J.A., 1989. *Aphids and virus diseases*. In: Cavalloro, R. (ed.), *Euraphid network: trapping and aphid prognosis*, Proc. of a meeting of the EC-Experts' group, Catania, Italy, 7-9.11.88, ECSC-EEC-EAEC, Brussels, Luxembourg, 197-212.
- De Bokx, J.A. & Piron, P.G.M., 1990. *Relative efficiency of a number of aphid species in the transmission of potato virus Yn in the Netherlands*. Neth. J. Pl. Path., 96: 237-246.
- Derron, J.O. & Goy, G., 1987. *Utilisation du piège à aspiration pour la prévision des épidémies de virus*. Rev. suisse agric., 19: 129-132.
- Derron, J.O. & Goy, G., 1990. *Importance relative des pucerons ailés les plus fréquemment rencontrés sur la pomme de terre comme vecteur du virus Y (PVYn)*, compte tenu de leur mobilité. Rev. suisse agric., 22: 277-281.
- Dixon, A.F.G. & Mercer, D.R., 1983. *Flight behaviour in the sycamore aphid: Factors affecting take-off*. Entomol. exp. appl., 33: 43-49.
- Dixon, A.F.G., 1985. *Aphid ecology*. Blackie, Glasgow, 157pp.
- Dorschner, K.W., 1990. *Aphid induced alteration of the availability and form of nitrogenous compounds in plants*. In: Campbell, R.K. & Eikenbary, R.D., *Aphid-plant genotype interactions*, Elsevier Publ., Amsterdam, 225-235.
- Edwards, A.R., 1963. *A non-colonizing vector of potato virus diseases*. Nature, 200: 1233-34.
- Fagen, R.F. & Young, D.Y., 1978. *Temporal patterns of behaviours: durations, intervals, latencies, and sequences*. In: Colgan, P.W. (ed.), *Quantitative ethology*. Wiley, New York, a.o., 79-114.
- Ferrar, F., 1969. *Interplant movement of apterous aphids with special reference to Myzus persicae (SULZ.) (Hem., Aphididae)*. Bull. ent. Res., 58: 653-60.
- Fischlin, A., 1982. *Analyse eines Wald-Insekten-Systems: Der subalpine Lärchen-Arvenwald und der graue Lärchenwickler Zeiraphera diniana Gn. (Lep., Tortricidae)*. Diss. ETH Zürich No. 6977, 294 pp.
- Fischlin, A., 1986. *Simplifying the usage and the programming of modern working stations with Modula-2: The 'Dialog Machine'*. Internal report, Project-Centre IDA/CELTIA, Swiss Federal Institute of Technology Zürich (ETHZ), Zürich, Switzerland, 13pp.
- Fischlin, A., 1991. *Interactive modeling and simulation of environmental systems on workstations*. In: Möller, D.P.F. & Richter, O. (eds.), Proc. of the 4th Ebernburer Working Conference on the analysis of dynamic systems in medicine, biology, and ecology, April, 5-7, 1990, Ebernburg, Bad Münster, BRD, Informatik-Fachberichte, Springer, Berlin, a.o., 131-145.
- Fischlin, A. & Schaufelberger, W., 1987. *Arbeitsplatzrechner im technisch-naturwissenschaftlichen Hochschulunterricht*. Bulletin SEV/VSE, 78: 15-21.
- Fischlin, A. & Ulrich, M., 1987. *Interaktive Simulation schlecht-definierter Systeme auf modernen Arbeitsplatzrechnern: die Modula-2 Simulationssoftware ModelWorks*. Proceedings, Treffen des GI/ASIM-Arbeitskreises 4.5.2.1 "Simulation in Biologie und Medizin", February, 27-28, 1987, Vieweg, Braunschweig.
- Fischlin, A., Roth, O., Gyalistras, D., Ulrich, M. & Nemecek, T., 1990. *ModelWorks: An interactive simulation environment for work stations and personal computers*. Internal Report No. 8, Systems Ecology Group, ETH Zürich, 186pp.
- Fritsche, R., Karl, E., Lehman, W. & Pröseler, G., 1972. *Tierische Vektoren pflanzenpathogener Viren*. VEB Gustav Fischer Verlag, Jena.

- Gabriel, W., 1987. *L'influence des pucerons et des conditions climatiques sur l'infection du plant par les virus de la pomme de terre.* In: Foldø, N.E., Hansen, S.E., Nielsen, N.K. & Rasmussen, R. (eds.): EAPR, Abstracts of the 10th Triennial Conference of the European Association for Potato Research, Aalborg, Denmark, 26-31 July, 1987, 44-45.
- Gabriel, W., Kostiw, M. & Wislocka, M., 1975. *Comparaison de plusieurs méthodes d'estimation de la quantité de pucerons vecteurs de virus, pour la prévision d'infection par virus des tubercules de pomme de terre.* Pot. Res., 18: 3-15.
- Gabriel, W., Szulc, J. & Wislocka, M., 1981. *Influence de la distance des source d'infection sur l'effet de traitements à l'aide d'insecticides systémiques sur la propagation des virus Y et M de la pomme de terre.* Pot. Res., 25: 1-11.
- Gibbs, A.J. & Gower, J.C., 1960. *The use of a multiple-transfer method in plant virus transmission studies - some statistical points arising in the analysis of results.* Ann. appl. Biol., 48: 75-83.
- Gibson, R.W., 1991. *The development of mature-plant resistance in four potato cultivars against aphid-inoculated potato virus Yo and Yn in four potato cultivars.* Pot. Res., 34: 205-210.
- Gibson, R.W. & Gunenc, Y., 1981. *Effects of covering potato crops with clear polyethylene film on spread of potato virus Y.* Pl. Path., 30: 233-235.
- Gibson, R.W. & Rice, A.D., 1989. *Modifying aphid behaviour.* In: Minks, A.K. & Harrewijn, P. (eds.), *Aphids - their biology, natural enemies and control.* Elsevier, Amsterdam, a.o., 2C: 209-224.
- Gibson, R.W., Pickett, J.A., Dawson, A.D., Rice, A.D. & Stibley, M.F., 1984. *Effects of aphid alarm pheromone derivatives and related compounds on non- and semi-persistent plant virus transmission by Myzus persicae.* Ann. appl. Biol., 104: 203-209.
- Gibson, R.W., Rice, A.D., Pickett, J.A., Smith, M.C. & Sawicki, R.M., 1982. *The effects of the repellents dodecanoid acid and polygodial on the acquisition of non- semi- and persistent plant viruses by the aphid Myzus persicae.* Ann. appl. Biol., 100: 55-59.
- Govier, D.A. & Kassanis, B., 1974. *Evidence that a component other than the virus particle is needed for aphid transmission of potato virus Y.* Virology, 57: 285-286.
- Govier, D.A., Kassanis, B. & Pirone, T.P., 1977. *Partial purification and characterization of the potato virus Y helper component.* Virology, 78: 306-314.
- Gray, S.M., Moyer, J.W., Kennedy, G.G. & Campbell, C.L., 1986a. *Virus-suppression and aphid resistance effects on spatial and temporal spread of watermelon mosaic virus 2.* Phytopathology, 76: 1254-59.
- Gray, S.M., Moyer, J.W. & Bloomfield, P., 1986b. *Two-dimensional distance class model for quantitative description of virus-infected plant distribution lattices.* Phytopathology, 76: 243-248.
- Greenway, A.R., Griffiths, D.C. & Lloyd, S.L., 1978. *Response of Myzus persicae to components of aphid extracts and to carboxylic acids.* Entomol. exp. appl., 24: 169-174.
- Gregory, P.H., 1948. *The multiple infection transformation.* Ann. appl. Biol., 35: 412-417.
- Gregory, P.H. & Read, D.R., 1949. *The spatial distribution of insect-borne plant virus diseases.* Ann. appl. Biol., 36: 475-82.
- Gugerli, P., 1979. *Le test immuno-enzymatique (ELISA) et son application pour le diagnostic rapide des viroses de la pomme de terre.* Revue suisse Agric., 11: 253-260.
- Haccou, P., 1986. *Analysis of behaviour by means of continuous time markov chain models and their generalizations.* In: Colgan, P.W. & Zayan, R. (eds.): Quantitative models in ethology, Privat, Readings from the 19. International ethological conference, 3: 81-96.
- Hampton, R.O., 1967. *Natural spread of viruses infectious to beans.* Phytopathology, 57: 476-481.
- Harrewijn, P., 1986. *Vector resistance of potato aphids with respect to potato leafroll virus.* In: Beckman, A.G.B., Louwes, K.M., Dellaert, L.M.W. & Meete, A.E.F. (eds.), Potato research of tomorrow - drought tolerance, virus resistance and analytic breeding methods, Proc. of an international seminar, Wageningen, Netherlands, 30.-31.10.85, PUDOC, Wageningen, 85-95.
- Harrewijn, P., 1989. *Integrated control of potato aphids.* In: Minks, A.K. & Harrewijn, P. (eds.), *Aphids - their biology, natural enemies and control.* Elsevier, Amsterdam, a.o., 2C: 279-284.
- Harrewijn, P., Van Hoof, H.A., & Noordink, J.P.W., 1981. *Flight behaviour of the aphid Myzus persicae during its maiden flight.* Neth. J. Pl. Path., 87: 111-117.
- Harrington, R., Katis, N., & Gibson, R.W., 1986. *Field assessment of the relative importance of different aphid species in the transmission of potato virus Y.* Pot. Res., 29: 67-76.
- Hastings, N.A.J. & Peacock, J.B., 1975. *Statistical distributions - A handbook for students and practitioners.* John Wiley & Sons, New York, a.o., 130pp.
- Häni, A. & Winiger, F.A., 1987. *Wie produziert die Schweiz Pflanzkartoffeln ohne Blattlausbehandlung?* Kartoffelbau, 38: 273-276.
- Hennig, E., 1969. *Untersuchungen über die Ursachen der Resistenz von Kulturpflanzen gegenüber Blattläusen: Der Einfluss von Temperatur, Licht und Alter des Pflanzengewebes auf das Probieren der*

- Schwarzen Bohnenblattlaus (Aphis Fabae SCOPOLI)*. Tagungsberichte der Landwirtschaftswissenschaften der DDR, Berlin, 80: 529-538.
- Hertbach, E., 1985. *Rôle des sémiocimiques dans les relations pucerons-plantes. I. - Généralités et phéromones d'alarme*. Agronomie, 5: 283-286.
- Hertbach, E., 1987. *Effect of dodecanoic acid on the colonization of sugar beet by aphids and the secondary spread of virus yellows*. Ann. appl. Biol., 111: 477-482.
- Hille Ris Lambers, D. & Schepers, A., 1978. *The effect of trans- β -farnesene, used as a repellent against landing aphid alatae in seed potato growing*. Pot. Res., 21: 23-26.
- Huntingford, F., 1984. *The study of animal behaviour*. Chapman and Hall, London, a.o., 411pp.
- Irwin, M.E., 1981. *Ecology and control of soybean mosaic virus*. In: Maramorosch, K. & Harris, K. (eds.), Plant diseases and vectors, Ecology and epidemiology, Academic Press N.Y., 181-220.
- Irwin, M.E. & Kampmeijer, G.E., 1989. *Vector behaviour, environmental stimuli and the dynamics of plant virus epidemics*. In: Jeger, M.J. (ed.), Spatial components of plant disease epidemics, Prentice Hall, Englewood Cliffs, NJ, 14-39.
- Irwin, M.E. & Ruesink, W.G., 1986. *Vector intensity: a product of propensity and activity*. In: McLean, G.D., Garrett, R.G., Ruesink, W.G. (eds.), Plant virus epidemics (monitoring, modelling, and predicting outbreaks), Academic Press, Sydney, 13-33.
- Jeffers, J.N.R. (ed.), 1988. *Practitioner's Handbook on the modelling of dynamic change in ecosystems*. John Wiley & Sons, Chichester, a.o.; SCOPE 34, 181pp.
- Johnson, B., 1957. *Studies on the degeneration of the flight muscles of alatae aphids - I. A comparative study of the occurrence of muscle breakdown in relation to reproduction in several species*. J. Ins. Physiol., 1: 248-256.
- Johnson, B., 1959. *Studies on the degeneration of the flight muscles of alatae aphids - II. Histology and control of muscle breakdown*. J. Ins. Physiol., 3: 367-377.
- Johnson, C.G., 1969. *Migration and dispersal of insects by flight*. Methuen, London, 763pp.
- Johnson, K.B., Johnson, S.B. & Teng, P.S., 1986. *Development of a simple potato growth model for use in crop-pest-management*. Agricultural systems, 19: 189-209.
- Kampmeijer, P. & Zadoks, J.C., 1977. *EPIMUL, a simulator of foci and epidemics in mixtures of resistant and susceptible plants, mosaics and multilines*. Simulation Monographs, PUDOC, Wageningen, 50pp.
- Katis, N. & Gibson, R.W., 1985. *Transmission of potato virus Y by cereal aphids*. Pot. Res., 28: 65-70.
- Keller, D., 1989. *Introduction to the Dialog Machine*. Bericht Nr. 5, Projektzentrum IDA, ETH Zürich, 37pp.
- Keller, E.R., 1958. *Rückblick auf zehnjährige Bemühungen zur Erzeugung von gesundem Kartoffelpflanzgut*. Anz. Schädlingsk., 31: 56-60.
- Kennedy, G.G., 1976. *Host-plant resistance and the spread of plant viruses*. Environ. Entomol., 5: 827-832.
- Kennedy, J.S., 1951. *A biological approach to plant viruses*. Nature, 168: 890-894.
- Kennedy, J.S., 1960. *The behavioural fitness of aphids as field vector of viruses*. Rept. 7th Commonwealth Entomol. Conf., London, 165-168.
- Kennedy, J.S., 1965. *Co-ordination of successive activities in an aphid. Reciprocal effects of settling on flight*. J. Exp. Biol., 42: 489-509.
- Kennedy, J.S., 1990. *Behavioural post-inhibitory rebound in aphids taking flight after exposure to wind*. Anim. Behav., 39: 1078-1088.
- Kennedy, J.S. & Booth, C.O., 1963. *Co-ordination of successive activities in an aphid. The effect of flight on the settling responses*. J. Exp. Biol., 40: 351-369.
- Kennedy, J.S. & Thomas, A.A.G., 1974. *Behaviour of some low-flying aphids in wind*. Ann. appl. Biol., 76: 143-159.
- Kennedy, J.S., Booth, C.O. & Kershaw, W.J.S., 1959. *Host finding by aphids in the field. II. Aphis fabae SCOP. (Gynoparae) and Brevicornis brassicae L., with a re-appraisal of the role of host finding behaviour to virus spread*. Ann. appl. Biol., 47: 424-444.
- Kermack, W.O. & McKendrick, A.G., 1927. *Contributions to the mathematical theory of epidemics*. Proc. Roy. Soc., 115: 700-721.
- Klingauf, F., 1976. *Die Bedeutung der "Stimmung" im Leben phytophager Insekten am Beispiel des Wirtswahl-Verhaltens von Blattläusen*. Z. angew. Ent., 82: 200-209.
- Klingauf, F.A., 1987. *Host plant finding and acceptance*. In: Minks, A.K. & Harrewijn, P. (eds.), *Aphids - their biology, natural enemies and control*, Elsevier, Amsterdam, a.o., 2A: 209-223.
- Klinkowski, M., 1980. *Pflanzliche Virologie - Band 1: Einführung in die allgemeinen Probleme*. Akademie-Verlag, Berlin, 658pp.
- Kostiw, M., 1975. *Investigation on the retention of potato viruses M and Y in two aphids (*Myzus persicae* SULZ. and *Aphis nasturtii* KALT.)*. Pot. Res., 18: 635-640.

- Kring, B.J., 1972. *Flight behaviour of aphids*. Ann. Rev. Ent., 17: 461-492.
- Kuroli, G. & Erdélyi, K., 1990. *Repellency examinations in potato stands*. Med. Fac. Landbouww. Rijksuniv. Gent, 55: 445-451.
- Labonne, G. & Quiot, J.B., 1988. *Influence de la vitesse et de la direction du vent sur la capture des pucerons ailés par un piège à fils englués*. Agronomie, 8: 465-469.
- Lehmann, W., Klaus, S. & Karl, E., 1975. *Das Verhalten der Aphiden unter Einfluss verschiedener Pflanzenarten und ihrer chemischen Behandlung, bezogen auf den Anteil einstechender Tiere und die kurzfristigen Probesaugstiche*. Archiv für Phytopathologie und Pflanzenschutz, Berlin, 11: 273-282.
- Loebenstein, G. & Raccah, B., 1980. *Control of nonpersistently transmitted viruses*. Phytoparasitica, 8: 221-235.
- Lowery, D.T. & Boiteau, G., 1988. *Effects of five insecticides on the probing, walking, and settling behavior of the green peach aphid (Homoptera: Aphididae) on potato*. J. econ. Entomol., 81: 208-214.
- Madden, L.V., 1989. *Dynamic nature of within-field spread and pathogen distributions*. In: Jeger, M.J. (ed.), *Spatial components of plant disease epidemics*, Prentice Hall, Englewood Cliffs, NJ, 96-126.
- Madden, L., Raccah, B. & Pirone, T.P., 1990. *Modelling plant disease increase as a function of vector numbers: nonpersistent viruses*. Res. Popul. Ecol., 32: 47-65.
- Madden, L.V., Louie, R. & Knoke, J.K., 1987. *Temporal and spatial analysis of maize dwarf mosaic epidemics*. Phytopathology, 77: 148-156.
- Magurran, A.E., 1988. *Ecological diversity and its measurement*. Croom Helm, London, 179pp.
- Marcus, R., 1990. *Modeling temporal progress of plant diseases*. Parasitica, 18: 341-351.
- Marcus, R., & Raccah, B., 1986. *Model for the spread of non-persistent virus-diseases*. J. appl. Stat., 13: 167-175.
- Marcus, R., Talpaz, H. & Bar-Joseph, M., 1989. *A model for the spread and control of citrus tristeza virus disease*. J. appl. Stat., 16: 315-320.
- Matthews, R.E.F., 1991. *Plant virology*. 3rd ed., Academic Press, New York.
- Matthews, J.R. & Matthews, R.W., 1978. *Insect behavior*. John Wiley & Sons, New York, a.o., 507pp.
- McLean, D. & Kinsey, M.G., 1968. *Probing behaviour of the pea aphid, Acyrthosiphon pisum. II. Comparisons of salivation and ingestion in host and non-host leaves*. Ann. ent. Soc. Am., 61: 730-39.
- Metz, H.A.J., Dienske, H., de Jonge, G. & Putters, F.A., 1983. *Continuous-time markov chains as models for animal behaviour*. Bulletin of Mathematical Biology, 45: 643-658.
- Moericke, V., 1955. *Über die Lebensgewohnheiten der geflügelten Blattläuse (Aphidina) unter besonderer Berücksichtigung des Verhaltens beim Landen*. Z. angew. Entomol., 37: 29-91.
- Müller, H.J. & Unger, K., 1955. *Über die Vorflugzeit von Doralis fabae SCOP. und ihre Bedeutung für den Massenwechsel*. Z. angew. Ent., 38: 82-96.
- N'Guetta, Y., R., Fargette, D. & Fauquet, C., 1986. *Influence du vent sur la dispersion des maladies virales transmises par aleurodes*. In: L'agrométéorologie et la protection des cultures dans les zones semi-arides. Nyamey, Niger, 8-12 Dec. 1986, 35-54.
- Nault, L.R. & Montgomery, M.E., 1977. *Aphid pheromones*. In: Harris, K.F. & Maramorosch, K. (eds.), *Aphids as virus vectors*, Academic Press, New York, 527-545.
- Peters, D., 1987. *Spread of viruses in potato crops*. In: De Boix, J.A. & van der Want, J.P.H. (eds.): *Viruses of potatoes and seed-potato production*. PUDOC, Wageningen, 2nd ed., 126-145.
- Peters, D., Brooijmans, E. & Grothuis, P.F.M., 1990. *Mobility as a factor in the efficiency with which the aphids can spread non-persistently transmitted viruses*. Proc. Exper. & Appl. Entomol., N.E.V. Amsterdam, I: 190-194.
- Phelan, P.L. & Miller, J.R., 1982. *Post-landing-behaviour of alate Myzus persicae as altered by (E)-β-farnesene and three carboxylic acids*. Entomol. exp. appl., 32: 46-53.
- Power, A.G., 1990. *Cropping systems, insect movement and the spread of insect transmitted diseases in crops*. In: Giessman, S.R. (ed.), *Agroecology (Ecological studies 78)*, Springer, New York, 47-69.
- Pröseler, G. & Weidling, H., 1975. *Die Retentionszeit von Stämmen des Kartoffelvirus Y in verschiedenen Aphidenarten und Einfluss der Temperatur*. Archiv für Phytopathologie und Pflanzenschutz, Berlin, 11: 335-345.
- Rek, J., 1987. *Untersuchungen über die Eignung des ELISA-Verfahrens zum serienmässigen Nachweis von Viren (PLRV und PVY) bei der Kartoffel unter Berücksichtigung des Infektionszeitpunktes und des physiologischen Zustandes der Knolle*. Diss. ETH Nr. 8285, 183pp.
- Reust, W., 1991. *Liste officielle suisse des variétés de pomme de terre 1992*. Rev. Suisse Agric., 23: I-VI.

- Rice, A.D., Gibson, R.W. & Sibley, M.F., 1983. *Effects of deltamethrin on walking, flight and potato virus Y-transmission by pyrethroid resistant *Myzus persicae**. Ann. appl. Biol., 102: 229-236.
- Robert, Y., 1987. *Dispersion and migration*. In: Minks, A.K. & Harrewijn, P. (eds.), *Aphids - their biology, natural enemies and control*. Elsevier, Amsterdam, a.o., 2A: 299-313.
- Robinson, R.A., 1976. *Plant pathosystems*. Springer, Berlin, 184pp.
- Rochow, W.F., 1974. *Vectors and variations*. In: Lawson, R.H. & Corbett, M.K. (eds.): Proc. of the 3rd Int. Symposium of virus diseases of ornamental plants. Int. Soc. of Horticultural Science, the Hague, Netherlands, May 1974, 75-83b.
- Rohitha, B.H. & Penman, D.R., 1986. *Flight of the bluegreen aphid *Acyrtosiphon kondoi* SHINJI (Homoptera: Aphididae) II. The effect of weather: multiple regression and flight threshold analyses*. New Zealand Journal of Zoology, 13: 209-214.
- Romanow, L.R., 1985. *Nonpersistent virus transmission. Its alternation by plant resistance factors*. Med. Fac. Landbouw. Rijksuniv. Gent, 50: 1265-70.
- Roth, O., Derron, J., Fischlin, A., Nemecek, T. & Ulrich, M., in press. *Implementation and parameter adaptation of a potato crop simulation model combined with a soil water subsystem*. In: van de Broek, B. & Marshall, B. (eds.), Proc. of the 1st Internat. Workshop on Potato Modelling, May 29-31, 1990, Wageningen, Netherlands. Pudoc, Wageningen, 30 pp.
- Rouse, D.I., 1985. *Construction of temporal models: I. Disease progress of air-borne pathogens*. In: Gilligan, C.A. (ed.): *Advances in plant pathology Vol. 3: Mathematical modelling of crop disease*. Academic Press, London, 11-29.
- Ruesink, W.G. & Irwin, M.E., 1986. *Soybean Mosaic Virus: A model and some implications*. In: McLean, G.D., Gareth, R.G., Ruesink, W.G. (eds.), *Plant virus epidemics (monitoring, modelling, and predicting outbreaks)*. Academic Press, Sydney, 295-313.
- Sachs, L., 1984. *Angewandte Statistik: Anwendung statistischer Methoden*. Springer Verlag, Berlin, 6th ed., 552pp.
- Sands, P.J., Hackett, C. & Nix, H.A., 1979. *A model of the development and bulking of potatoes (*Solanum tuberosum* L.) I. Derivation from well-managed field crops*. Field Crops Research, 2: 309-331.
- Sassen, B., 1983. *The effect of two pyrethroids on the feeding behaviour of three aphid species and on transmission of two different viruses*. Z. Pfl.Krankh. u. Pfl.Schutz, 90: 119-126.
- Schepers, A., 1989. *Control of aphids - chemical control*. In: Minks, A.K. & Harrewijn, P. (eds.), *Aphids - their biology, natural enemies and control*. Elsevier, Amsterdam, a.o., 2C: 89-122.
- Schwärzel, R. & Gehrig, W., 1989. *La certification des plants de pommes de terre*. Revue suisse Agric., 21: 67-69.
- Scott, G.E., 1985. *Nonrandom spatial distribution of aphid-vectorated Maize Dwarf Mosaic*. Plant Disease, 69: 893-895.
- Shanks, C.H. & Chapman, R.K., 1965. *The effects of insecticides on the behaviour of the green peach aphid and its transmission of potato virus Y*. Econ. Entomol., 58: 79-83.
- Sherwood, M.H., Greenway, A.R. & Griffiths, D.C., 1981. *Responses of *Myzus persicae* (SULZER) (Hemiptera: Aphididae) to plants treated with fatty acids*. Bull. Entomol. Res., 71: 133-136.
- Sibly, R.M., Nott, H.M.R. & Fletcher, D.J., 1990. *Splitting behaviour into bows*. Anim. Behav., 39: 63-39.
- Sigvald, R., 1985. *Mature-plant resistance of potato plants against potato virus Y(0) (PVY0)*. Pot. Res., 28: 135-143.
- Sigvald, R., 1986. *Forecasting the incidence of potato virus Y0*. In: McLean, G.D., Gareth, R.G., Ruesink, W.G. (eds.), *Plant virus epidemics (monitoring, modelling, and predicting outbreaks)*. Academic Press, Sydney, 419-441.
- Sigvald, R., 1989. *Relationship between aphid occurrence and spread of potato virus Y0 (PVY0) in field experiments in Southern Sweden*. J. appl. Ent., 108: 35-43.
- Singh, M.N. & Khurana, S.M.P., 1987. *Effect of temperature on aphids acquisition/inoculation of potato virus Y*. In: Foldø, N.E., Hansen, S.E., Nielsen, N.K. & Rasmussen, R. (eds.): EAPR, Abstracts of the 10th Triennial Conference of the European Association for Potato Research, Aalborg, Denmark, 26-31 July, 1987, 405-406.
- Singh, M.N., Khurana, S.M.P., Nagach, B.B. & Agrawal, H.O., 1983. *Evidence on heredity variations in the virus transmission efficiency of aphid clones*. Z. Pfl.Krankh. u. Pfl.Schutz, 90: 345-351.
- Slater, P.J.B., 1978. *Data collection*. In: Colgan, P.W. (ed.), *Quantitative ethology*, Wiley, 7-24.
- SMA, 1980. *Projekt ANETZ 1974-1980: Automatisches Mess- und Beobachtungsnetz. Zusammenfassungen der Vorträge ANETZ-Kolloquien, 26.-27.11.1980, Zürich, Schweiz*. Meteorologische Anstalt, 25pp.
- SMA, 1985. *Charakteristiken der ANETZ-Daten*. Beiträge zum ANETZ-Daten Kolloquium, 17.4.85, Zürich, Schweiz. Meteorologische Anstalt, 56pp.

- Snell, J.L., 1989. *Introduction to probability*. McGraw-Hill International Editions, New York, a.o., 478pp.
- Storch, R.H., 1981. *Insects and diseases in the production of seed potatoes*. In: Lashomb, J.H., Casagrande, R. (eds.), Advances in Pest potato management, Hutchinson Ross, Stroudsbourg, Pennsylvania, 138-152.
- Sustare, B.D., 1978. *Systems diagrams*. In: Colgan, P.W. (ed.), Quantitative ethology, Wiley, New York, a.o., 275-311.
- Swenson, K.G., 1968. *Role of aphids in the ecology of plant viruses*. Ann. Rev. Phytopath., 6: 351-374.
- Sylvester, E.S., 1988. *Virus transmission studies*. In: Minks, A.K. & Harrewijn, P. (eds.), *Aphids - their biology, natural enemies and control*, Elsevier, Amsterdam, a.o., 2B: 69-81.
- Sylvester, E.S., 1989. *Viruses transmitted by aphids*. In: Minks, A.K. & Harrewijn, P. (eds.), *Aphids - their biology, natural enemies and control*, Elsevier, Amsterdam, a.o., 2C: 65-86.
- Taimr, L. & Kriz, J., 1978. *Post-migratory flights of Phorodon humuli SCHRANK winged migrants in a hop garden*. Z. angew. Ent., 85: 236-240.
- Tarn, T.R. & Adams, J.B., 1982. *Aphid probing and feeding, and plant breeding*. In: Harris, K.F. & Maramorosch, K. (eds.), Pathogens, vectors and plant diseases, Academic Press, New York, 221-246.
- Tatchell, G.M. & Woiwood, I.P., 1989. *Aphid migration and forecasting*. In: Cavalloro, R. (ed.), *Euraphid network: trapping and aphid prognosis*, Proc. of a meeting of the EC-Experts' group, Catania, Italy, 7.-9.11.88, ECSC-EEC-EAEC, Brussels, Luxembourg, 15-28.
- Taylor, L.R., 1960. *The distribution of insects at low levels in the air*. J. Anim. Ecol., 29: 45-63.
- Taylor, L.R. & Palmer, J.M.P., 1972. *Aerial sampling*. In: van Emden, H.F. (ed.): *Aphid Technology*, Academic Press, London, 189-234.
- Thomas, A.A.G., Ludlow, A.R. & Kennedy, J.S., 1977. *Sinking speeds of falling and flying Aphis fabae SCOPOLI*. Ecol. Ent., 2: 315-326.
- Thresh, J.M., 1976. *Gradients of plant virus diseases*. Ann. appl. Biol., 82: 381-406.
- Thresh, J.M., 1978. *The epidemiology of plant virus diseases*. In: Scott, P.R. & Bainbridge, A. (eds.), *Plant disease epidemiology*, Blackwell, Oxford, a.o., 79-91.
- Thygesen, T., 1968. *The dispersal of apterous Aphis gossypii GLOV. marked with P32*. Acta Agriculturae Scandinavica, 18: 196-198.
- Turl, L.A.D. & MacDonald, D.M., 1987. *The relationship between suction trap catches and the spread of PVY in South-East Scotland*. In: Cavalloro, R. (ed.): *Aphid migration and forecasting "Euraphid" systems in European countries*, 177-185.
- Van der Werf, W. & Riesebos, W., 1990. *Models in the epidemiology of beet yellowing viruses*. Proc. IIRB (Inst. Int. Rech. Betr.) 1990.
- Van der Werf, W., Rossing, W.A.H., Rabbinge, R., de Jong, M.D. & Mols, P.J.M., 1989. *Approaches to modelling the spatial dynamics of pests and diseases*. In: Cavalloro, R. & Delucchi, V. (eds.), Parasitism 88, Proc. of a scientific congress, Barcelona, 25-28.10.88, 89-119.
- Van der Zaag, D.E., 1987. *Yield reduction in relation to virus infection*. In: De Bokx, J.A. & van der Want, J.P.H. (eds.): *Viruses of potatoes and seed-potato production*, 2nd ed., 146-161.
- Van Harten, A., 1983. *The relation between aphid flight and the spread of potato virus Y(N) in the Netherlands*. Potato Res., 26: 1-15.
- Van Hoof, H.A., 1977. *Determination of the infection pressure of PVY(N)*. Neth. J. Pl. Path., 83: 123-127.
- Van Hoof, H.A., 1980. *Aphid vectors of PVY(N)*. Neth. J. Pl. Path., 86: 159-162.
- Vanderplank, J.E., 1963. *Plant diseases: Epidemics and Control*. Academic Press, New York a.o., 349pp.
- Vanderplank, J.E., 1982. *Host-pathogen interactions in plant disease*. Academic Press, New York a.o., 207pp.
- Venekamp, J.H., Schepers, A. & Bus, C.B., 1980. *Mature plant resistance of potato against virus diseases. III: Mature plant resistance against potato virus Yn, indicated by decrease in ribosome-content in ageing potato plants under field conditions*. Neth. J. Plant Path., 86: 301-309.
- Von Mises, R., 1918. *Über die "Ganzahligkeit" der Atomgewichte und verwandte Fragen*. Physikalische Z., 19: 490-500.
- Völk, J., 1959. *Zur Uebertragung des Y-Virus durch Insekten und Kontakt*. Z. Pfl.Krankh. u. Pfl.Schutz, 66: 563-571.
- Waggoner, P.E. & Rich, S., 1981. *Lesion distribution, multiple infection, and the logistic growth of plant diseases*. Proc. Natl. Acad. Sci. USA, 78: 3292-95.
- Walters, K.F.A. & Dixon, A.F.G., 1984. *The effect of temperature and wind on the flight activity of cereal aphids*. Ann. appl. Biol., 104: 17-26.

- Weidemann, H.L., 1981. *Die Einstichaktivität von Blattläusen, gemessen an der Anzahl von Speichel-scheiden, im Hinblick auf Virusübertragungen.* Z. angew. Ent., 92: 92-98.
- Wichmann, B. & Hill, D., 1982. *An efficient and portable pseudo-random number generator.* Appl. Stat., 31: 188-90.
- Wichmann, B. & Hill, D., 1987. *Building a random-number generator.* Byte, 12: 127-128.
- Wiktelius, S., 1982. *Flight and settling behaviour of Rhopalosiphum padi L. (Homoptera: Aphididae).* Bull. Ent. Res., 72: 157-163.
- Wirth, N., 1985. *Programming in Modula-2, Third, Corrected Edition.* Springer-Verlag, Berlin a.o., 202pp.
- Wirth, N., Gutknecht, J., Heiz, W., Schär, H., Seiler, H., Vetterli, C. & Fischlin, A., 1992. *MacMETH - A fast Modula-2 language system for the Apple Macintosh.* User Manual, Departement Informatik, ETH Zürich, 116pp.
- Wislocka, M., 1982. *Einfluss der Trockenheit vor und zu verschiedenen Zeitpunkten nach Inkulation auf den Knollenbefall der Kartoffelsorte Uran mit Kartoffelvirus Y.* Pot. Res., 25: 293-298.
- Wolfram, S., 1986. *Theory and applications of cellular automata.* Singapore: World Scientific, 560pp.
- Woodford, J.A.T., 1976. *The importance of aphid flight behaviour for the spread of potato viruses.* Bulletin of the Scottish Horticultural Research Institute, 11: 15-20.
- Woodford, J.A.T., 1979. *The control of aphid-borne viruses in seed potato crops.* In: Bulletin of the Scottish Horticultural Research Institute, 16: 30-36.
- Wymore, A.W., 1984. *Theory of systems.* In: Vick, C.R. & Ramamoorthy, C.V. (eds.): *Handbook of Software Engineering*, Van Nostrand Reinhold Company, New York, 119-133.
- Yang, S.L., & Zettler, F.W., 1975. *Effect of alarm pheromones on aphid probing behaviour and virus transmission efficiency.* Plant. Dis. Repr., 59: 902-905.
- Zadoks, J.C. & Kampmeijer, P., 1977. *The role of crop populations and their deployment, illustrated by means of a simulator, Epimul 76.* Annals of the N.Y. Acad. of Science, 287: 164-190.
- Zadoks, J.C. & Schein, R.C., 1979. *Epidemiology and plant disease management.* Oxford University Press, New York, a.o., 427pp.
- Zadoks, J.C. & Schein, R.C., 1980. *Epidemiology and plant disease management, the known and the needed.* In Palti, J. & Kranz, J. Comparative epidemiology, Proc. of the session on Comparative epidemiology, 3rd Int. Congress of plant pathology, Munich, 16-23.8.1978, 1-17.
- Zar, J.R., 1984. *Biostatistical analysis.* Prentice-Hall, Englewood Cliffs, N.J., 718pp.
- Zeigler, B.P., 1976. *Theory of Modelling and Simulation.* John Wiley & Sons, 435pp.
- Zettler, F.W., 1967. *A comparison of Aphididae with species of other aphid families regarding virus transmission and probe behaviour.* Phytopathology, 57: 398-400.
- Zitter, T.A., 1977. *Epidemiology of aphid-borne viruses.* In: Harris, K.F. & Maramorosch, K. (eds.), *Aphids as virus vectors*, Academic Press, New York, 385-412.

APPENDIX I: GLOSSARY

The following definitions are for the reader's convenience and help to clarify the sometimes conflicting usage in the literature. Arrows (\rightarrow) reference terms, which are themselves mentioned in the glossary.

Acquisition period:	Time required for an aphid to acquire virus from an infectious plant.
Antibiosis:	Kind of plant resistance leading to higher mortality or lower fecundity.
Antixenosis:	Kind of plant resistance leading to non-acceptance or non-preference of the plants.
Colonization group:	Group of aphid species with the same relationship to the potato plant, i.e. either \rightarrow colonizing or \rightarrow non-colonizing species.
Colonizing aphid species:	Aphid species, which survive and reproduce on a plant species.
Data frame:	Set of all data (parameter values, initial values, input data, etc.) required to define an executable simulation model together with the model equations.
Disease incidence:	Fraction of a plant population that is infected (CAMPBELL & MADDEN, 1990).
Disease intensity:	'Quantity of disease present' (CAMPBELL & MADDEN, 1990). Disease intensity can be divided into \rightarrow disease incidence and \rightarrow disease severity.
Disease severity:	Fraction of plant tissue or plant area that is infected (CAMPBELL & MADDEN, 1990), e.g. fraction of infected tubers, or proportion of leaf area infected.
Distance class distribution:	Distribution of the relative frequencies of infected plants at a distance from an \rightarrow secondary infection source (see Chapter 2.4.2).
Epidemic:	'Change of disease intensity in a host population over time and space' (CAMPBELL & MADDEN, 1990).
Infected tubers:	Tubers showing a positive reaction in ELISA.
Infectible plant:	Plant healthy at emergence that may become infected during a growing season.
Infection:	Successful introduction of virus particles into a healthy plant, i.e. \rightarrow inoculation of a healthy plant (RUESINK & IRWIN, 1986).

Infection cycle:	Epidemic process from infection until the infected plant material can serve as an infection source (ZADOKS & SCHEIN, 1979).
Infectious period:	Plant-virus interaction: time during which a plant can serve as infection source. Aphid-virus interaction: time during which an infectious aphid can transmit the disease.
Inoculation:	Successful introduction of virus particles into either an infected or a noninfected plant, i.e. an inoculation is a potential → infection (IRWIN & RUESINK, 1986; RUESINK & IRWIN , 1986).
Inoculation period:	Time required for an aphid to inoculate a plant after having acquired the virus.
Inoculation probability:	Probability that an aphid will transmit a virus disease to a healthy plant if it probes on this plant.
Inoculation sequence:	Sequence of behaviour events necessary for a successful → inoculation, i.e. probing or feeding on an infectious plant, movement to another plant and probing or feeding there.
Latent period:	Plant-virus interaction: time between the → infection and the instant when the plant becomes an infection source. Aphid-virus interaction: time between virus acquisition and first possible inoculation.
Monocyclic process:	Epidemic process consisting of one → infection cycle (ZADOKS & SCHEIN, 1979).
Noncolonizing aphid species:	Aphid species not surviving and reproducing on a plant species.
Nonpersistent transmission:	Virus transmission mode without latent period, requiring acquisition and infection probes of few seconds only.
Nonresident aphid species:	→ Noncolonizing aphid species.
Pathosystem:	Set of all components involved in an epidemic (ZADOKS & SCHEIN, 1980).
Penetration:	Stylet penetration activities of aphids into plant tissue, including probing and feeding on the plant (TARN & ADAMS, 1982).
Persistent transmission:	Virus transmission mode with a → latent period, requiring long acquisition and infection feedings of several hours.
Plant with infected tubers:	Plant having at least one → infected tuber in the sample.

Polycyclic process:	Epidemic process consisting of several → infection cycles (ZADOKS & SCHEIN, 1979).
Polyetic process:	Epidemic process over several years (ZADOKS & SCHEIN, 1979).
Primary infection:	→ Infection originating from an → inoculation during the growing season.
Removal:	Death or roguing of a plant.
Resident aphid species:	→ Colonizing aphid species.
Retention rate:	Rate of decline of → inoculation probability of a fasting aphid.
Retention time:	Time during which an aphid can transmit a virus disease after having acquired it.
Secondary infection:	→ Infection originating from an → inoculation during the previous growing season. Plants growing from an infected mother tuber are secondarily infected.
Serial inoculation probability:	Conditional → inoculation probability in the second, third, etc. probe, relative to the immediately preceding probe.
Spatial pattern (of disease):	Spatial arrangement of infected plants (after CAMPBELL & MADDEN, 1990).
Susceptible plant:	Plant that can be infected.
Torus:	A space without borders. By 'connecting' the borders of a finite space, a torus can be used to simulate an infinite space by a finite number of elements.
Vector efficiency:	→ Inoculation probability in the first probe immediately after virus acquisition on an infection source plant. Vector efficiency is one parameter of → vector propensity.
Vector group:	Group of aphid species with the same → vector propensity.
Vector intensity:	Number of → inoculations caused by the alatae per infection source plant and time unit, calculated as the product of vector abundance, → inoculation sequence frequency and → vector propensity (source: IRWIN & RUESINK, 1986).
Vector propensity:	Conditional probability that an aphid, having previously probed on an infection source plant, will transmit the virus (IRWIN & RUESINK, 1986). It is a function of → vector efficiency, → retention rate and → serial inoculation probability.

APPENDIX II: TABLE OF SYMBOLS

General remarks:

Some symbols used only at one place in the text are not listed. Character P resp. p is in general used to denote a proportion or probability ($P/p \in [0,1]$). In general, upper case letters are used for state variables and matrices, lower case letters for parameters, inputs, auxiliary variables and outputs. The abbreviation 'pl' stands for 'plant'.

Use of indices:

C = index variable for colonization group, $C \in \{\text{colonizing } (col), \text{ noncolonizing } (ncol)\}$. G = index variable for vector group, $G \in \{1, 2a, 2b, 3\}$. I, J = index variables for behaviour state, $I, J \in \{\text{flight } (f), \text{ walking } (w), \text{ resting } (r), \text{ probing } (p), \text{ 'long penetration' } (l)\}$. V = index variable for virus strain, $V \in \{PVYn, PVYo\}$. A '+' sign after a parameter denotes a scaling parameter introduced for the sensitivity analysis, which is multiplied with the respective parameters (without '+'). E.g. e_{G+} is multiplied with each of the vector efficiency parameters e_1 to e_3 .

Lower case indices x and y in the cellular automaton models are used for the metric coordinates in the two-dimensional plane. Upper case indices are used for the row number X and the plant number within the row Y .

Tab. A2: Table of symbols. Types of variables in dynamic models: S = state, P = parameter, I = global input, O = global output, A = auxiliary variable. Local input and output variables are classified as auxiliary variables. For variables appearing in the listings (Appendix III and IV) the identifiers are given. # = number of module, where the variable is used (see also Tab. 2.1): 0 = SpatialEpi, 1 = PAVTypes, 2 = PAVBase, 3 = PAVAux, 4 = PAVSensAna, 5 = PAVValid, 6 = Weather, 7 = PAVSucITrap, 8 = SoilWat, 9 = PotatoMod, 10 = PotModValid, 11 = PAVMetHour, 12 = PAVSteadyBeh, 13 = PAVInocs, 14 = PAVStochInfs, 15 = PAVSpatDistr, 16 = PAVMonit, 17 = PAVSetGetObj, 18 = PAVMoEDMngr, 19 = PAVExpts, 20 = PAVMaster.

Symbol	Type	Meaning	Identifier	#	Unit
A	S	dry matter of assimilate pool	assim	9	g/pl
α	P	shape parameter of Weibull distribution of flight distances	alpha	0, 14	-
a_l	A	leaf area	-		m^2/pl
a_{ll}	P	area of the lowest leaves of a plant	lowLeafArea	13	m^2/pl
a_v	A	proportionality variable used for calculation of vertical wind profile	-		m/s
β	P	scale parameter of Weibull distribution of flight distances	beta	14	-
b_n	-	slope parameter of the regression model used in the sensitivity analysis	-		-
c	-	class index for distance class distribution analysis	-	#	-
c_h	A	indicator variable of favourable flight conditions of hour h	-		-
D	S	disease incidence (proportion of latently infected, infectious and removed plants)	-		%/100
d_C	P	mortality rate of colonizing resp. noncolonizing aphids	mortality[c]	13	d^{-1}
$d_{col/ncol}$	P	see d_C	see d_C		d^{-1}
d_f	-	degrees of freedom in statistical tests	-	#	-

Symbol	Type	Meaning	Identifier	#	Unit
d_f	S	flight distance	-		m
D_h	S	proportion of healthy individuals in the population	-		%/100
$\delta_h, \delta_l, \delta_p, \delta_s$	-	disease states, see Tab. 4.1	healthy, latent, primInf, 1, 14 secinf, removed	-	-
δ_r					
D_i	S	proportion of infectious plants in the population	-		%/100
D_k	-	threshold, where the behaviour sequences become stationary	-		s
D_I	S	proportion of latently infected individuals in the population	-		%/100
d_j	A	distance of loss in flight experiments	-		m
d_m	P	width of the margin area around the experimental plot for simulation of the validation experiments	margin	14	m
D_o	P	zero plane displacement, constant for calculation of vertical wind profile	-		m
d_p	P	distance between plants in a row	plantDist	9	m
d_{pl}	A	plant density (number of mother tubers emerged per m ²)	-		pl/m ²
D_r	S	proportion of removed (immune, dead or rogued) plants in the population	-		%/100
d_r	P	distance between rows	rowDist	9	m
d_{st}	-	distance between a secondary source plant and a plant with infected tubers	-		m
Δv_f	P	difference between active flight speed and wind speed	-		m/s
d_w	P	mean walking distance of aphids	walkDist	14	m
D_{XY}	S	disease state of cell $[X, Y]$	infStateK ^a [x] ^a [y]	14	-
E	O	evenness of distance class distribution	evennessVals[i]	15	-
$e_{1/2a/2b/3}$	P	see e_G	see e_G		%/100
e_{co}	-	expected number of plants with infected tubers in a distance class c from a secondary infection source s in a simulation run o	-		#
e_G	P	vector efficiency of group G	vectorEff[g]	13	%/100
$\phi(\theta)$	-	probability density function of the von Mises distribution	-		-
$F(x)$	-	distribution function of a statistical distribution	-		-
	x				
	$(= \int f(x))$				
$f(x)$	-	probability density function of a statistical distribution	-		-
f_c	-	ratio of observed and expected number of plants with infected tubers in class c	-		# / #
f_{fC}	A	flight frequency of colonization group C	flightFreq[c]	13	d ⁻¹
f_i	A	fraction of radiation intercepted	fRadl	13, 9	%/100
f_{JC}	A	frequency of behaviour state J of colonization group C	-		d ⁻¹
$f_{im\ col/ncol}$	P	see f_{imC}	see f_{imC}		d ⁻¹
f_{imC}	P	rate of interleaf movement by walking of colonization group C	interLeafMoveRate[c]	13	d ⁻¹
f_{pc}	A	probing frequency of colonization group C	probFreq[c]	13	d ⁻¹
f_{ptC}	A	frequency of interplant movement by flight of colonization group C	moveRateF[c]	13	d ⁻¹
f_{pwIC}	A	frequency of interplant movement by walking over leaf bridges of colonization group C	moveRateWL[c]	13	d ⁻¹
f_{pwsC}	A	frequency of interplant movement by walking on soil of colonization group C	moveRateWS[c]	13	d ⁻¹
f_{tp}	P	hourly precipitation threshold for take-off	precThreshold	11	mm/h
f_{tt}	P	hourly mean temperature threshold for take-off	tempThreshold	11	°C

Symbol	Type	Meaning	Identifier	#	Unit
t_{lw}	P	hourly mean wind speed threshold for take-off (wind speed measured at 12 m height)	windThreshold	11	m/s
f_x	A	width of simulated field	fieldSizeM.x	14	m
f_y	A	length of simulated field	fieldSizeM.y	14	m
γ	P	removal rate	-	-	d ⁻¹
G^2	-	likelihood ratio test value for log-linear models	-	-	-
H	A	Shannon-Wiener index	-	-	-
h_{cso}	-	number of plants without infected tubers in a distance class c from a secondary infection source s in a simulation run o	-	-	#
h_h	A	the time of the h -th hour (e.g. 16:00 if $h=16$)	-	-	h
h_j	A	height of loss in flight experiments	-	-	m
h_m	A	the midpoint of the h -th hour (e.g. 16:30 if $h=16$)	-	-	h
H_{max}	-	maximal value of H	-	-	-
i	A	vector intensity, total number of inoculations per infection source plant, $i \in \Re$	inoculations	13	#pl ⁻¹ d ⁻¹
i'	A	infection rate or 'corrected basic infection rate'	-	-	d ⁻¹
i^*	A	effective number of inoculations per source plant, $i^* = n_{inocInt} \in \mathbb{N}$	ninocInt	14	#pl ⁻¹ d ⁻¹
I_1, I_2, I_3	-	time intervals 0-300 s, 300-600 s resp. 600-900 s after the beginning of the behaviour observation	-	-	s
I_2'	-	time interval $[D_k, D_k+300]$ after the beginning of the behaviour observation	-	-	-
i_{cso}	-	number of plants with infected tubers in a distance class c from a secondary infection source s in a simulation run o	-	-	#
i_d	A	distance of an inoculation from the source	diffPosPol.length	0	m
i_{df}	A	distance of an inoculation from the source caused by a flying vector	diffPosPol.length	14	m
i_{dws}	A	distance of an inoculation from the source caused by a vector walking on soil	diffPosPol.length	14	m
i_e	P	external inoculations (caused by vectors immigrating into the field)	-	-	d ⁻¹
i_ϕ	A	direction of an inoculation in relation to the source	diffPosPol.angle	14	rad
i_f	A	number of inoculations caused by flying aphids per source plant	inoculationsF	13, 14	#pl ⁻¹ d ⁻¹
$i_{\phi f}$	A	direction of an inoculation in relation to the source caused by a flying vector	diffPosPol.angle	14	rad
$i_{\phi ws}$	A	direction of an inoculation in relation to the source caused by a vector walking on soil	diffPosPol.angle	14	rad
i_G	A	number of inoculations caused by vector group G	inocsGrp[g]	13	#pl ⁻¹ d ⁻¹
i_i	P	internal inoculations (caused by vectors migrating within the field)	-	-	d ⁻¹
i_{im}	A	number of inoculations caused by immigrating aphids per plant	inoculationsIm	13, 14	#pl ⁻¹ d ⁻¹
i_{im}'	A	number of inoculations caused by immigrating aphids in the whole field	ninocInt	14	#d ⁻¹
$Int(x)$	-	integer part of x	-	-	-
i_{wl}	A	number of inoculations caused by aphids walking over leaf bridges per source plant	inoculationsWL	13, 14	#pl ⁻¹ d ⁻¹
i_{ws}	A	number of inoculations caused by aphids walking on soil per source plant	inoculationsWS	13, 14	#pl ⁻¹ d ⁻¹
$i_{x,y}$	A	cell coordinates of inoculated plant	inocPIno.x, inocPIno.y	0, 14	-
$i_{x,y}$	A	metric coordinates of inoculated plant	inocPos.x, inocPos.y	0, 14	m
κ	P	concentration parameter of von Mises distribution of flight directions	kappa	14	-
k_{apt}	P	'apterae factor', hypothetical factor simulating the contribution of apterae to virus spread	apteraeEff	13	%/100

Symbol	Type	Meaning	Identifier	#	Unit
k_{dl}	P	scaling factor of the mean residence time of behaviour state l	durSA[i]	13	%/100
k_{fJ}	P	scaling factor of the frequency of behaviour state J	freqSA[j]	13	%/100
k_{jXY}	A	infection time of the cell $[X, Y]$	tuberInfAgeC^*[x]^*[y]	14	-
L	S	leaf dry matter	leaf	9	g/pl
l_c	-	upper limit of distance class c	-	-	m
L_i	A	dry matter of leaf tissue produced during day i	t^*biomass	9	g/pl
l_s	A	leaf senescence	dLeafSenesc	9	gd^-1pt^-1
ly	A	proportion of physiologically young leaf tissue in total leaf tissue	youngLeafTissue	14, 9	%/100
μ	P	mean of statistical distribution	-	-	-
$m_{d\Delta}$	A	daily dry matter increase	deltaDM	9	g d^-1pt^-1
m_{eC}	A	relative emigration rate of colonization group C	emigr[c]	13	d^-1
m_{IG}	A	immigration rate of group G	immigr[g]	13	#pt^-1d^-1
$MIN(a, b)$	-	the smaller value of a and b	-	-	-
m_l	P	mean latent period	latPerMean	14	DD
m_s	P	mean infectious period	-	-	DD
m_t	P	mean delay to tuber infection	infTubDel	14	DD
$N(\bar{x}, s)$	A	normally distributed random variates with mean \bar{x} and standard deviation s	N()	14	-
N_1 to N_3	S	see N_G	see N_G	-	#/pl
N_G	S	number of unsettled alatae of group G per plant	nAph[g]	13	#/pl
n_{inf}	A	number of infections occurring	effInfections	14	#pt^-1d^-1
n_0	-	number of simulation runs in simulation experiment	-	-	#
n_p	P	number of plants in a row	nPlants	14	-
n_r	P	number of rows	nRows	14	-
n_s	-	number of secondary infection sources in the field	-	-	#
n_t	P	number of tubers sampled per plant	nTubSample	14	#/pl
o	-	index for simulation run in distance class distribution analysis	-	-	#
P	-	type I error probability in statistical tests	-	-	%/100
p_n	P	standard value of parameter p_n	-	-	*)
p_a	A	fraction of day with favourable conditions for take-off	favourCond	13	%/100
P_C	P	transition probability matrix for discrete Markov chain model for colonization group C	P[c]	13	-
p_c	-	'proportion' of l_c falling in distance class c	-	-	-
p_d	O	disease incidence, proportion of infected plants in the population	cumPPI[latent]	14	%/100
p_{ern}	P	conditional probability that an aphid leaves the field after take-off	emigrProb	13	%/100
p_G	O	fraction of infections contributed by vector group G	propInfGrp[g]	14	%/100
p_h	I	hourly precipitation	-	-	mm/h
p_i	P	proportion of secondarily infected plants at emergence time (initial disease incidence)	initInf	14	%/100
p_{ib}	P	initial disease incidence in the neighbourhood of the experimental plot	basicInf	14	%/100
P_{IC}^*, P_{JC}^*	P	proportion of aphids in steady state in behaviour state i resp. J of colonization group C	nProp[c,i], nProp[c,j]	13	%/100
P_{IC}, P_{JC}	A/S	proportion of aphids in behaviour state i resp. J of colonization group C	-	-	%/100
P_{IJC}	P	transition probability from behaviour state i to state J	P[c,i,j]	13	%/100
p_{il}	P	conditional probability of an infected plant to become an infection source	propInfLeaves	14	%/100
p_n	P	the n -th parameter	-	-	*)
p_n'	P	standardized value of parameter p_n	-	-	%/100

Symbol	Type	Meaning	Identifier	#	Unit
p_n	P	value of parameter p_n in the sensitivity analysis	-	-)
p_{min}, p_{max}	P	minimum/maximum of plausibility range of parameter p_n in the sensitivity analysis	-	-)
p_p	O	proportion of plants with at least one infected tuber in the sample	pInfPl	14	%/100
p_{piC}	A	probability that an aphid probes on the source plant ppic	ppic	13	%/100
p_r	O	proportion of removed plants in the field	propPlants[removed]	14	%/100
p_s	O	proportion of infection source plants in the field (primary and secondary sources)	inf	14	%/100
p_t	O	average proportion of infected tubers in the field	pInfTub	14	%/100
p_{wC^*}	A	proportion of aphids walking in steady state of colonization group C	nPropW[c]	13	%/100
p_{wl}	A	conditional probability of an interplant movement by moveProbL leaf bridges, given an interleaf movement occurs	moveProbL	13	%/100
p_{wp}	P	probability that an interleaf movement leads to a movement to a neighbouring plant, given these two plants touch	movePIProb	13	%/100
p_{ws}	A	conditional probability that an aphid walks off the plant, given an interleaf movement occurs	moveProbS	13	d ⁻¹
Q_C	P	transition rate matrix for continuous Markov chain model for colonization group C	Q[c]	13	-
q_{iIC}	P	same as q_{iJC} , for I=J	Q[c,i,j]	13	d ⁻¹
q_{iJC}	P	element of the Q_C -matrix, transition rate from behaviour state I to J of colonization group C for continuous Markov chain model	Q[c,i,j]	13	d ⁻¹
q_{ipC}	P	same as q_{iJC} , for J=p	Q[c,i,probng]	13	d ⁻¹
q_{plC}	P	same as q_{iJC} , for I=p and J=l	Q[c, probng, longPenetration]	13	d ⁻¹
R	S	root dry matter	root	9	g/pl
r	P	relative removal rate	-	-	
r_d	I	global daily radiation	RADG	6, 9, 8	Jcm ⁻² d ⁻¹
$r_{f1/2a/2b/3}$	P	see r_{IG}	-	-	
r_{fG}	P	retention factor of vector group G	retFact[g]	13	d ⁻¹
r_h	I	mean daily rel. humidity	RH	6, 8	%
r_{IC}	A	mean residence time of behaviour state I	-	-	d
r_o	I	removal of plants	-	-	-
S	S	stem dry matter	stem	9	g/pl
s	-	index for secondary infection source in distance class distribution analysis	-	-	#
σ	-	standard deviation of statistical distribution	-	-	-
s_C	P	'concentration factor' for aphid immigration	concFactor	13	%/100
s_d	P	deposition rate of aphids	depositRate	13	m ²
s_G	I	number of alatae of group G caught in suction trap	suctTrapGr[g]	13	# / d
s_{IXY}	A	susceptibility of a plant to virus infections	sensitivityC[x][y]	14	%/100
s_I	P	standard deviation of latent period	latPerSD	14	DD
$s_{r col/ncol}$	P	see s_{rC}	see s_{rC}	-	d ⁻¹
s_{rC}	P	relative settling rate of colonization group C	settRate[c]	13	d ⁻¹
s_I	P	standard deviation of the delay to tuber infection	infTubSD	14	DD
s_x, s_y	A	cell coordinates of an infection source plant	source.x, source.y	0, 14	-
s_x, s_y	A	metric coordinates of infection source plant	sourcePos.x, sourcePos.y	0, 14	m
T	S	tuber dry matter	tuber	9	g/pl
t_1, t_2, t_3	A	monitoring times for statistical analysis	-	-	julian day
t_a	A	actual transpiration	TA	8	mm/d
$t_{ar PVYn/0}$	P	see t_{arV}	see t_{arV}	-	DD
t_{arV}	P	age threshold for physiologically young leaves	leafAgeThresh	9	DD
t_{em}	P	date of 50 % crop emergence	emergD	5	julian day

Symbol	Type	Meaning	Identifier	#	Unit
t_{end}	P	simulation stop time	-	-	julian day
τ_h	-	plant without infected tubers	-	-	-
t_h	I	mean hourly temperature	-	-	°C
t_{hk}	P	haulm-killing date	killD	5	julian day
τ_i	-	plant with infected tubers	-	-	-
t_{iXY}	A	infection time of plant $[X, Y]$	-	-	DD
t_{IXY}	A	time when plant $[X, Y]$ becomes an infection source (outbreak time)	outbreakAgeC^x^y	14	DD
t_m	A	potential transpiration	TM	8	mm/d
t_{max}	I	daily maximum temperature	TMAX	6, 9, 8	°C
t_{min}	I	daily minimum temperature	TMIN	6, 9, 8	°C
t_0	P	simulation start time	-	-	julian day
T_p	S	physiological age of the potato crop	pA	14, 9	DD
T_{pi}	A	physiological age of leaf tissue produced on day i	t^i.pAge	9	DD
t_{pC}	A	mean time spent on a plant by an aphid of colonization group C	1/moveRate[c]	13	d
$t_{s1/2a/2b/3}$	P	see t_{SG}	see t_{SG}	-	%/100
t_{sG}	P	serial inoculation probability of vector group G	serTransm[g]	13	%/100
t_{sr}	I	sunrise time	-	-	h
t_{ss}	I	sunset time	-	-	h
$t_{t,b}$	P	threshold for touching of leaves between plant rows	touchBetween	13	%/100
$t_{t,w}$	P	threshold for touching of leaves within a plant row	touchWithin	13	%/100
t_{tXY}	A	time when tubers of plant $[X, Y]$ becomes infected	tuberInfAgeC^x^y	14	DD
t_X, t_Y	-	cell coordinates of an infectible plant	-	-	-
$t_X Y$	O	proportion of infected tubers of plant $[X, Y]$	infTubersK^x^y	14	%/100
u, u_1, u_2	A	random number sampled from uniform distribution $U[0, 1]$	U()	14	-
$U[a,b)$	A	uniformly distributed random variates between a and b	UGeneral(a,b)	14	-
v_c	P	rate of climb of alatae	-	-	m/s
v_f	A	flight speed of alatae	-	-	m/s
$v_{f,max}$	P	maximum flight speed of alatae	-	-	m/s
v_G	A	vector propensity of group G	vp	13	%/100
v_h	I	mean hourly wind speed (measured at 12 m height)	-	-	m/s
v_m	P	proportion of immigrating aphids coming directly from infection sources outside the field	immigrTransm	14	%/100
$VM(\mu_0, \kappa)$	A	von Mises distributed random variates with mean μ_0 and concentration parameter κ	VM()	14	-
v_S	P	sinking speed of alatae	-	-	m/s
v_W	I	daily mean wind velocity	VEN	6, 8	m/s
$v_W(z)$	A	wind velocity at height z	-	-	m/s
w_ϕ	I	mean wind direction during favourable flight conditions	windDir	14	rad
w_i	I	daily irrigation	IRIG	6, 8	mm/d
w_p	I	daily precipitation	PREC	6, 8	mm/d
W_r	S	water content in root zone	WR	8	mm
W_s	S	water storage on soil surface	SSTO	8	mm
w_s	A	water stress	wStress	8, 9	%/100
X	A	row index	x	14	#
Y	A	index of plant in row	y	14	#
z	S	height above ground	-	-	m
z_0	P	roughness length, constant for calculation of vertical wind profile	-	-	m

APPENDIX III: LISTING OF THE SIMULATION MODEL 'SPATIAL EPIDEMIC' [0]

The module 'SpatialEpi' contains 280 lines with 10300 characters in the source code (comments and empty lines not included). The (compiled) object code contains 5034 characters ('Compile20' used). The modules in Appendices III-V use MacMETH V 3.2, DM V 2.2 and ModelWorks V 2.2.

```

MODULE SpatialEpi;

(* **** Spatial epidemic model of virus diseases *)
(* **** Simulation of spatial dynamics of a virus epidemic in a potato field, allowing
   for different statistical distributions of the distances inoculated plant -
   infection source plant:
      * normal
      * exponential
      * 2-parametric Weibull

   The simulation step is 1 week. *)

(* other modules *)
FROM CAModBase IMPORT CAMod,CAArD,UpdateMode,StateType,nonexistentCAMod,
   Dec1CAMod,GetStatePrtsD,GetCASpecs,RemoveCAMod,UpdateCAState,CellID,
   ResetCAUpdateState,CAFrequencies,DoForAllCells,FillCAModD,CAExists,SetCellID;
FROM MathProcs IMPORT Round,PI;
FROM Vectors2D IMPORT IntVector,Vector,PolVector,TransPolToCartVector,
   AddVectors;
FROM RandGen IMPORT U;
FROM RandNormal IMPORT InstallU,SetPars,N;
FROM RandGens IMPORT RandExp,RandWeibull,UGeneral;
(* MW modules *)
FROM SimBase IMPORT DeclIM,IntegrationMethod,StashFiling,CurrentStep,
   Tabulation,Graphing,DeclMV,DeclIP,RICType,Model,SetSimTime,
   InstallStartConsistency,Parameter,AuxVar,NoTerminate;
FROM SimMaster IMPORT RunSimMaster;
(* DM modules *)
FROM DMWindowIO IMPORT WriteString,WriteLn,WindowFont,FontStyles,FontStyle,
   SetWindowFont,SetPos;
FROM DMMenus IMPORT InstallAbout;
(* PAV modules *)
FROM PAVTypes IMPORT healthy,latent,primInf,secInf,removed;
FROM PAVaux IMPORT PostToPlant,PlantToPos,Torus,RandRealToInt,
   InfectionSource,Logit;
FROM PAVBase IMPORT noT,noF,noG;

CONST
   harvTime = 9;

(* state changes: *)
noChange = 0; infect = 1; remove = 2;
notInfAge = MAX(INTEGER);
version = '*** Model SpatialEpidemic version 4 ***';

VAR
   m: Model;

(* state & aux. variables *)
   infState,infAge,infChange: CAMod;
   infStateK,infStateKl,infAgeK,infChangeK: CAArD;

(* monitorable variables *)
   numDpl: ARRAY [healthy..removed] OF LONGINT;
   propPl: ARRAY [healthy..removed] OF AuxVar;
   inf,diseasedPlants,newInfs,logitX: AuxVar;

(* parameter *)
   nRows,nPlants,initInf(*proportion of initially infected plants*),nInoc,rowDist,
   plantDist,latPer,roguingEfficiency1,roguingEfficiency2,roguingTime1,roguingTime2,
   roguingTime3,stdDevNorm,mean,alpha,distrChoice,torus:
   Parameter;

```

```

(* auxiliary variables *)
fieldSizeI:           IntVector;
fieldSizeM:           Vector;

(*****)
(* Auxiliary procedures *)
(*****)

PROCEDURE ChoosePlant(sourcePos: Vector; VAR inside: BOOLEAN): IntVector;
(* samples a plant from a 2-dimensional distribution *)
VAR diffPosPol:       PolyVector;
inocPos, diffPos:     Vector;
inocPlNo:             IntVector;

PROCEDURE InocDistribution(): REAL;
BEGIN
  IF distrChoice < 1.5 THEN
    RETURN N()                      (* normal      distribution *)
  ELSIF distrChoice < 2.5 THEN
    RETURN RandExp(mean)            (* exponential distribution *)
  ELSE
    RETURN RandWeibull(alpha, mean) (* Weibull      distribution *)
  END (*IF*);
END InocDistribution;
BEGIN
  diffPosPol.angle := UGeneral(0.0, 2.0*Pi());
  diffPosPol.length := InocDistribution();
  diffPos := TransPolToCartVector(diffPosPol);
  inocPos := AddVectors(sourcePos, diffPos);
  PosToPlant(inocPos, inocPlNo, rowDist, plantDist);
  Torus(inocPlNo, fieldSizeI, inside);
  RETURN inocPlNo;
END ChoosePlant;

PROCEDURE CalcParams(): BOOLEAN; (* start consistency procedure *)
BEGIN
  SetPars(0.0, stdDevNorm);
  RETURN TRUE;
END CalcParams;

PROCEDURE InitFieldPars;
BEGIN
  fieldSizeI.x := Round(nRows); fieldSizeI.y := Round(nPlants);
  fieldSizeM.x := nRows*rowDist; fieldSizeM.y := nPlants*plantDist;
END InitFieldPars;

PROCEDURE AboutProc;
BEGIN
  SetPos(2,1);
  SetWindowFont( Geneva, 12, FontStyle{} );
  WriteString("          Spatial Epidemic");
  SetWindowFont( Geneva, 10, FontStyle{} );
  WriteString("          Spatial Virus Epidemiology Model"); WriteLn;
  WriteString("          Version 4"); WriteLn;
  WriteString("          © 1991 by Thomas Nemecsek"); WriteLn;
  WriteString("          Simulates a hypothetical virus"); WriteLn;
  WriteString("          epidemics in time and space."); WriteLn;
END AboutProc;

(*****)
(* CA-management      *)
(*****)

PROCEDURE TermCAModel;
BEGIN
  RemoveCAMod(infAge);
  RemoveCAMod(infChange);
  RemoveCAMod(infState);
END TermCAModel;

PROCEDURE InstallCAModel(x, y: INTEGER);
  VAR dummy: CAArD;
BEGIN
  TermCAModel;
  DeclCAMod (infState, discrete,x,y,healthy, removed,withNewState);
  DeclCAMod (infChange,discrete,x,y,noChange,remove, noUpdate);
  DeclCAMod (infAge,   discrete,x,y,0,           10,      noUpdate);
  GetStatePtrsD (infState, infStateK, infStateK1);
  GetStatePtrsD (infChange,infChangeK,dummy);
  GetStatePtrsD (infAge,   infAgeK,   dummy);
END InstallCAModel;

```

```

PROCEDURE InitCAModel(x, y: INTEGER);
  (* declares resp. redefines the model (if size changed) and prepares for graphical
  monitoring. x,y = size of CA-model *)
  VAR stateType:           StateType;
      nR,nC,minState,maxState: INTEGER;
      mode:                  UpdateMode;
BEGIN
  IF NOT CAExists(infState)
  THEN InstallCAModel(x, y)
  ELSE GetCASpecs (infState, stateType, nR, nC, minState, maxState, mode);
  IF (x # nR) OR (y # nC)
  THEN InstallCAModel(x, y)
  END(*IF*);
END(*IF*);
END InitCAModel;

(*****)
(* Model procedures *)
(*****)
PROCEDURE ResetUpdate; (* reset the state change *)
BEGIN
  FillCAModD(infChange, noChange);
  ResetCAUpdateState(infState);
END ResetUpdate;

PROCEDURE SetSecInf(x,y: INTEGER);
BEGIN
  SetCellID(infState,x,y,secInf);
  SetCellID(infAge, x,y,CurrentStep());
END SetSecInf;

PROCEDURE ChooseSecInf;
  (* random choice of seondarily infected plants for initial state *)
VAR infPos: Vector; infPlNo: IntVector;
BEGIN
  infPos.x := UGeneral(0.0, fieldSizeM.x);
  infPos.y := UGeneral(0.0, fieldSizeM.y);
  PostToPlant(infPos, infPlNo, rowDist, plantDist);
  IF CellID(infState, infPlNo.x, infPlNo.y) = secInf THEN ChooseSecInf END(*IF*);
  SetSecInf(infPlNo.x, infPlNo.y);
END ChooseSecInf;

PROCEDURE NewInfections(caMod: CAMod; x, y: INTEGER);
VAR
  i,nInocInt: INTEGER; inocPlNo, source: IntVector;
  inside: BOOLEAN; sourcePos: Vector;
BEGIN
  IF InfectionSource(infStateK^x^y) THEN (* plant is an infection source *)
    source.x := x; source.y := y;
    nInocInt := RandRealToInt(ninoc);
    FOR i := 1 TO nInocInt DO
      PlantToPos(source, sourcePos, rowDist, plantDist);
      inocPlNo := ChoosePlant(sourcePos, inside);
      IF (torus > 0.5) OR inside
      THEN SetCellD(infChange,inocPlNo.x,inocPlNo.y,infect); (* checks indices *)
      END(*IF*);
    END (*FOR*);
  END (*IF*);
END NewInfections;

PROCEDURE NewState(caMod: CAMod; x,y: INTEGER);
BEGIN
  IF infChangeK^x^y = remove (* rogueing *)
  THEN infStateK1^x^y := removed;
  ELSIF (infStateK^x^y = healthy)
  THEN (infChangeK^x^y = infect) (* infection *)
  THEN IF latPer < 0.5
  THEN infStateK1^x^y := primInf;
  ELSE infStateK1^x^y := latent;
  END(*IF*);
  infAgeK^x^y := CurrentStep();
  newInfs := newInfs + 1.0;
ELSIF (infStateK^x^y = latent)
  THEN infStateK1^x^y := primInf (* outbreak *)
  ELSE infStateK1^x^y := infStateK^x^y; (* no state change *)
  END(*IF*);
END NewState;

```

```

PROCEDURE Statistics;
(* statistics of monitorable variables *)
BEGIN
  CAFrequencies(infState, numbPl, propPl, FALSE);
  inf := propPl[primInf] + propPl[secInf];
  diseasedPlants := inf + propPl[latent];
  Logit(m, logitX, diseasedPlants);
END Statistics;

PROCEDURE InitDistr; (* defines the initial disease pattern *)
  VAR i,n: INTEGER;
BEGIN
  FillCAModD(infState, healthy);
  FillCAModD(infAge, notInfAge);
  n := Round(initInf*nRows*nPlants);
  FOR i:= 1 TO n DO
    ChooseSecInf;
  END(*FOR*);
  ResetUpdate;
END InitDistr;

PROCEDURE Initial;
BEGIN
  InitFieldPars;
  InitCAModel(Round(nRows), Round(nPlants));
  InitDistr; Statistics;
END Initial;

PROCEDURE Input;
BEGIN
  ResetUpdate;
  newInfs := 0.0;
END Input;

PROCEDURE Output;
BEGIN
  Statistics;
END Output;

PROCEDURE Dynamic;
BEGIN
  (* calculation of new state *)
  DoForAllCells(infState, NewInfections, FALSE);
  DoForAllCells(infState, NewState, FALSE);

  UpdateCAState(infState);
END Dynamic;

PROCEDURE ModelObjects;
BEGIN
  DeclMV(propPl[healthy],0.0,1.0,"healthy plants",      "healthy","%/100",
  noF,noT,noG);
  DeclMV(propPl[latent], 0.0,1.0,"latent plants",       "latent","%/100",
  noF,noT,noG);
  DeclMV(propPl[primInf],0.0,1.0,"prim. inf. sources","prim.inf.", "%/100",
  noF,noT,noG);
  DeclMV(propPl[secInf], 0.0,1.0,"sec. inf. sources", "sec. inf.", "%/100",
  noF,noT,noG);
  DeclMV(propPl[removed],0.0,1.0,"dead or removed plants", "dead","%/100",
  noF,noT,noG);
  DeclMV(diseasedPlants, 0.0,1.0,"diseased plants",     "diseased","%/100",
  noF,noT,isY);

  DeclMV(inf,      0.0, 1.0,"inf. source plants", "inf","%/100",noF,noT,noG);
  DeclMV(newInfs, 0.0,100.0,"new infections", "newInfs","#",   noF,noT,noG);
  DeclMV(logitX,-12.0, 12.0,"logit(diseased)", "logit(x)","#",   noF,noT,noG);

  DeclP(nRows, 40.0,1.0,8000.0,noRtc,"% of rows",      "nRows", "#");
  DeclP(nPlants,80.0,1.0,8000.0,noRtc,"% of plants",   "nPlants", "#");
  DeclP(initInf,0.01,0.0, 1.0,noRtc,"initial infection", "initInf","%/100");
  DeclP(nInoc, 1.0, 0.0, 10.0, rtc,"% inoc. per source", "nInoc", "%/pl/d");
  DeclP(latPer, 3.0, 0.0, 20.0, rtc,"latent period",   "latPer", "weeks");

  DeclP(rowDist, 0.75, 0.0,2.0,rtc,"distance between rows", "rowDist", "m");
  DeclP(plantDist,0.375,0.0,1.0,rtc,"distance within row", "plantDist", "m");

```

```

(* distribution parameters *)
DeclP(stdDevNorm, 3.0,0.0,1000.0,  rtc,"stDev of normal distr.",
      "stdDevNorm","m");
DeclP(mean,      3.0,0.0,1000.0,  rtc,"mean of exp. and Weibull distr.",
      "mean","m");
DeclP(alpha,     1.0,0.0,1000.0,  rtc,"alpha of Weibull distr.",
      "alpha","-");

(* switch parameter *)
DeclP(distrChoice,1.0,1.0,   3.0,  rtc,"1=normal 2=exponential 3=Weibull",
      "distrChoice","-");
DeclP(torus,      1.0,0.0,   1.0,noRtc,"0=no torus,1=torus",
      "torus","-");

(* roguing parameters *)
DeclP(roguingEfficiency1,0.0,0.0,1.0,rtc,"roguing efficiency prim. inf",
      "roguing eff1","");
DeclP(roguingEfficiency2,0.0,0.0,1.0,rtc,"roguing efficiency sec. inf",
      "roguing eff2","");
DeclP(roguingTime1, 2.0,1.0,100.0,rtc,"roguing time 1","roguing1","time");
DeclP(roguingTime2, 5.0,1.0,100.0,rtc,"roguing time 2","roguing2","time");
DeclP(roguingTime3,10.0,1.0,100.0,rtc,"roguing time 3","roguing3","time");
END ModelObjects;

PROCEDURE ModelDeclaration;
BEGIN
  DeclM(m,discreteTime,Initial,Input,Output,Dynamic,NoTerminate,
        ModelObjects,"Spatial Virus Epidemic Model","SpatialEpi",AboutProc);
  SetSimTime(0.0,FLOAT(harvTime));
  InstallAbout("About SpatialEpidemic",230,120,AboutProc);
  InstallStartConsistency(CalcParams);
  InitCAModel(Round(nRows),Round(nPlants));
END ModelDeclaration;

BEGIN
  infState := nonexistentCAMod;
  infAge   := nonexistentCAMod;
  infChange := nonexistentCAMod;
  InstallU();
  RunSimMaster(ModelDeclaration);
  TermCAModel;
END SpatialEpi.

```

APPENDIX IV: LISTINGS OF THE PAV SIMULATION SYSTEM

The 19 definition modules of the PAV simulation system listed in Appendix IV contain 480 lines (25 lines/module) with 17800 characters (937 characters/module) in the source code (comments and empty lines not included). The corresponding 19 implementation modules together with the module PAVMaster (which has no definition module) contain 5730 lines (287 lines/module) with 214700 characters (10736 characters/module) in the source code, resp. 100100 bytes (5005 bytes/module) in the object code (compiled with 'Compile20'). To simplify, 5 modules defining the other models of the PAV simulation system were not listed. Their source code contains additional 80 lines (16 lines/module) with 1900 characters (384 characters/module) in the definition and 960 lines (193 lines/module) with 40000 characters (8004 characters/module) in the implementation, resp. 19400 bytes (3871 bytes/module) in the object code ('Compile20' used).

PAV Base Modules

PAVTYPES [1] D *)

*) D = definition, I = implementation module, the numbers in brackets refer to the module numbers in Tab. 2.1 and Tab. A2.

```
DEFINITION MODULE PAVTYPES;
(* ****)
  Purpose
    Definition of data types for the PAV simulation system

  Programming
    • Design:      T. Nemecek   07.09.90
    • Implementation: T. Nemecek   07.09.90

  Last revision of definition: 02.05.91 tn
(* ****)

CONST
  (* disease states: *)
  healthy = 0; latent = 1; primInf = 2; secInf = 3; removed = 4;
  firstSpec = MIN(AphSpec); lastSpec = MAX(AphSpec);
  firstBeh = MIN(Behaviour); lastBeh = MAX(Behaviour);

TYPE
  (* virus vector species: *)
  SigvSpec = (AFs, ANs, APs, BBs, BHs, MDs, MEs, MPs, PHs, RPs, SAAs, OTs);
  (* vector species used in the Sigvald-model:
     AFs = A.fabae      ANs = A.nasturtii    APs = A.pisum
     BBs = B.brassicae   BHs = B.helichrysi   MDs = M.dirrhodum
     MEs = M.euphorbiae  MPs = M.persicacei  PHs = P.humuli
     RPfs = R.padi       SAAs = S.avaneae     OTs = other species *);
  AphSpec = (AF, AN, AP, BH, ME, MP, PH, RP);
  (* vector species used in all models except Sigvald:
     AF = A.fabae      AN = A.nasturtii    AP = A.pisum
     BH = B.helichrysi   ME = M.euphorbiae  MP = M.persicacei
     PH = P.humuli      RP = R.padi *);

  Colonization = (colonizing, noncolonizing);
  SpeciesGroup = (grp1, grp2a, grp2b, grp3);

(* These groups are defines according to Derron & Goy (1990).
The species groups have similar virus transmission properties. The groups consist of following species:
  grp1:      M.persicaceae
  grp2a:     A.nasturtii, M.euphorbiae
  grp2b:     A.fabae, B.helichrysi, P.humuli
  grp3:      A.pisum, R.padi *)
```

```

AphSpecAttr = RECORD
    col: Colonization;
    grp: SpeciesGroup;
    END;

Behaviour = (flight, walking, resting, probing, longPenetration);

Virus = (PVYn, PVYo, PLRV);

BehArray = ARRAY [MIN(Behaviour)..MAX(Behaviour)] OF REAL;
BehMatrix = ARRAY [MIN(Behaviour)..MAX(Behaviour)],
    [MIN(Behaviour)..MAX(Behaviour)] OF REAL;
END PAVTypes.

```

PAVBASE [2] D

```

DEFINITION MODULE PAVBase;
{*****}
Purpose
    service module providing some procedures and variables for
    the PAV-Simulation-system

Programming
    • Design:      T.Nemecek          26.09.90
    • Implementation: T.Nemecek        26.09.90

    Last revision of definition: 12.05.92 tn
{*****}

FROM SimBase   IMPORT RTCType,StashFiling,Tabulation,Graphing,Model,
IntegrationMethod,Stain,LineStyle,StateVar,Derivative,Parameter;
FROM DMFiles   IMPORT Textfile;
FROM DMMenus   IMPORT Menu;
FROM PAVTypes  IMPORT Behaviour,SigvSpec,AphSpec,AphSpecAttr,SpeciesGroup,
Colonization,healthy,removed;

CONST
    maxModels = 10;
    maxDFs = 30;

    (* model indices *)
    validMind = 0; vdPlankMInd = 1; sigvaldMind = 2;
    ruesinkMind = 3; EPOVIRDetMInd = 4; EPOVIRSpatMInd = 5;
    johnsonMind = 6; myModell = 7; myModel2 = 8;

    nSensTub = 3;
    (* # of monitoring times between emergence and haulm-killing *)

TYPE ModelItem = RECORD
    name: ARRAY [0..63] OF CHAR;
    active: BOOLEAN;
    END;

    DFItem = RECORD
    name: ARRAY [0..63] OF CHAR;
    fileName: ARRAY [0..127] OF CHAR;
    END;

VAR
    version:           ARRAY [0..63] OF CHAR;
    versionNr:         ARRAY [0..7] OF CHAR;
    versionDate:       ARRAY [0..31] OF CHAR;
    refF:              Textfile;
    writeRes:          BOOLEAN;
    pavMenu:           Menu;
    curModelNumb,curDFNumb: INTEGER;
    model:             ARRAY [0..maxModels] OF ModelItem;
    df:                ARRAY [1..maxDFs] OF DFItem;
    curDF:             INTEGER;
    calcEvenness:      BOOLEAN;

    (* labels and attributes *)
    diseaseStateLabel: ARRAY [healthy..removed],[0..20] OF CHAR;
    behLongLabel:       ARRAY [MIN(Behaviour)..MAX(Behaviour)],[0..20] OF CHAR;
    behShortLabel:      ARRAY [MIN(Behaviour)..MAX(Behaviour)],[0..1] OF CHAR;
    sigvSpecShortLabel: ARRAY [MIN(SigvSpec)..MAX(SigvSpec)],[0..5] OF CHAR;
    sigvSpecLongLabel:  ARRAY [MIN(SigvSpec)..MAX(SigvSpec)],[0..20] OF CHAR;
    aphSpecShortLabel:  ARRAY [MIN(AphSpec)..MAX(AphSpec)],[0..5] OF CHAR;
    aphSpecLongLabel:   ARRAY [MIN(AphSpec)..MAX(AphSpec)],[0..20] OF CHAR;

    aphSpecAttr:        ARRAY [MIN(AphSpec)..MAX(AphSpec)] OF AphSpecAttr;

```

```

aphGroupAttr:      ARRAY [MIN(SpeciesGroup)..MAX(SpeciesGroup)]
                  OF Colonization;

(* monitoring attributes *)
(*****)
noT,                      (* - notInTable *)
intT: Tabulation;          (* - writeInTable *)
noG: Graphing;             (* - notInGraph *)
noF,                      (* - notOnFile *)
onF: StashFiling;          (* - writeOnFile *)

(*****)
(* Group attributes *)
(*****)
PROCEDURE InitGroupsNormal; (* normal attributes *)
PROCEDURE InitGroupsColonizing; (* all groups treated as colonizing *)
PROCEDURE InitGroupsNoncolonizing; (* all groups treated as noncolonizing *)

(*****)
(* Auxiliary procedures *)
(*****)
PROCEDURE FillArray (VAR ar: ARRAY OF REAL; val: REAL);
(* assigns 'val' to all elements of 'ar' *)
PROCEDURE CheckIntMethod (VAR m: Model);
(* ensures that the model 'm' is integrated with Euler's method *)
PROCEDURE AddIndex(VAR ident: ARRAY OF CHAR; idstr: ARRAY OF CHAR;
ind: INTEGER; inBrackets: BOOLEAN); (* assigns 'idstr' to 'ident' plus the
index 'ind' which is enclosed in brackets, if 'inBrackets' is TRUE *)
PROCEDURE InstallPAVMenu; (* installs a menu for the PAVSS *)

(*****)
(* Declaration procedures *)
(*****)
(* The following procedures declare different ModelWorks-object and add the
model identifier (mInd) at the beginning of the descriptor and the
identifier (for MVs) *)
PROCEDURE DeclIndM (VAR m: Model; defaultMethod: IntegrationMethod;
initial, input, output, dynamic, terminal: PROC; installModelObjects: PROC;
descriptor, identifier: ARRAY OF CHAR; about: PROC; mInd: INTEGER);
PROCEDURE DeclIndSV (VAR s: StateVar; VAR ds: Derivative; initial,minRange,
maxRange: REAL; descriptor,identifier,unit: ARRAY OF CHAR; mInd: INTEGER);
PROCEDURE DeclIndP (VAR p: Parameter; defaultVal,minVal,maxVal: REAL;
runTimeChange: RCType; descriptor,identifier,unit: ARRAY OF CHAR;
mInd, userLevel:INTEGER);
PROCEDURE DeclIndMV (VAR mv: REAL; defaultScaleMin,defaultScaleMax: REAL;
descriptor,identifier,unit: ARRAY OF CHAR; defaultSF: StashFiling; defaultT:
Tabulation; defaultG: Graphing; mInd,userLevel: INTEGER);

(* The following 2 procedures may be used to declare a SV or a P at the same
time also as MV *)
PROCEDURE DeclIndSvAndMv(VAR s: StateVar; VAR ds: Derivative; initial,minRange,
maxRange,defaultScaleMin,defaultScaleMax: REAL; descriptor,identifier,unit:
ARRAY OF CHAR; defaultSF: StashFiling; defaultT: Tabulation; defaultG: Graphing;
mInd, userLevel: INTEGER);
PROCEDURE DeclIndPAndMv (VAR p: Parameter; defaultVal,minVal,maxVal,
defaultScaleMin, defaultScaleMax: REAL; runTimeChange: RCType; descriptor,
identifier, unit: ARRAY OF CHAR; defaultSF: StashFiling; defaultT: Tabulation;
defaultG: Graphing; mInd,userLevel: INTEGER);

(*****)
(* Set procedures *)
(*****)
PROCEDURE SetDefltAndCurP (m: Model; VAR p: Parameter; val: REAL);
PROCEDURE SetDefltAndCurSV(m: Model; VAR sv: StateVar; val: REAL);
PROCEDURE SetDefltCurveAttrForExMV(m:Model; VAR mv:REAL; st:Stain; ls:LineStyle;
sym:CHAR); (* checks whether mv has been declared *)

(*****)
(* Identifier retrieval *)
(*****)
PROCEDURE GetPIdent (m: Model; VAR p: Parameter; VAR ident: ARRAY OF CHAR);
PROCEDURE GetSVIdent(m: Model; VAR sv: StateVar; VAR ident: ARRAY OF CHAR);

(*****)
(* File output during simulation experiments *)
(*****)
PROCEDURE WriteRes(): BOOLEAN;
(* returns TRUE if during sim.exp. and file is open *)
PROCEDURE Line;           (* writes EOL to resF *)
PROCEDURE Tab;             (* writes tab to resF *)

(*****)
(* User levels *)
(*****)
(* The user level allows to adapt the system to the current needs of the user.

```

```
The level is set once at startup and should not be changed afterwards. *)

CONST
  sDEM = 3; (* demo version for external users *)
  demo = 5; (* for demo versions *)
  ext1 = 11; (* for beginner external usage *)
  ext2 = 15; (* for advanced external usage *)
  int1 = 21; (* for basic internal usage *)
  int2 = 25; (* for 'full' internal usage *)

PROCEDURE SetUserLevel( userLevel: INTEGER );
PROCEDURE GetUserLevel(VAR userLevel: INTEGER);
PROCEDURE UserLevel(): INTEGER;

END PAVBase.
```

PAVAUX [3] D

```
DEFINITION MODULE PAVAux;
(* ****
  Purpose
    provides auxiliary procedures used in the PAV-simulation-system,
    mainly in the infection submodel of EPOVIR
  Programming
    * Design: T. Nemecek 13.2.91
    * Implementation: T. Nemecek 13.2.91
  Last revision of definition: 25.07.91 TN
*)
FROM Vectors2D IMPORT Vector, IntVector;
FROM SimBase IMPORT Model;
FROM SYSTEM IMPORT ADDRESS;

PROCEDURE InsuffMem;
  (* displays a message 'Insufficient memory and provokes a program abort *)
PROCEDURE CheckAllocation(ptr: ADDRESS);
  (* checks, whether 'ptr # NIL' and calls 'InsuffMem' otherwise *)
PROCEDURE AttentiveWait(seconds: REAL); (* the wait procedure is interrupted
  every 0.2 seconds and DialogMachineTask is called *)

  (* the following 2 procedures assign 'val' to the to the first 'n'
  elements of 'ar' *)
PROCEDURE FillArrayI(VAR ar: ARRAY OF INTEGER; n: INTEGER; val: INTEGER);
PROCEDURE FillArrayR(VAR ar: ARRAY OF REAL; n: INTEGER; val: REAL);

PROCEDURE ModAngle(angle: REAL): REAL;
  (* transforms 'angle' so that it lies in the interval [0, 2π] *)
PROCEDURE CompassToAngle(compassDir: REAL): REAL;
  (* translates the compass direction:
    N=0, E=1/2π, S=π, W=3/2π
    into a polar coordinate system which is oriented as follows:
    N=1/2π, E=0, S=3/2π, W=π *)
PROCEDURE PostoPlant(pos: Vector; VAR plNo: IntVector; rDist, pDist: REAL);
  (* Eq 4.6; converts the metric coordinates 'pos' into the cell coordinates
  'plNo'. The cell coordinate system has its origin at the left lower corner of
  the plant *)
PROCEDURE PlantToPos(plNo: IntVector; VAR pos: Vector; rDist, pDist: REAL);
  (* Eq 4.5; converts the cell coordinates 'plNo' into the metric coordinates
  'pos' *)
PROCEDURE InfectionSource(state: INTEGER): BOOLEAN;
  (* returns TRUE if the cell is an infection source *)
PROCEDURE RandRealToInt(nR: REAL): INTEGER; (*Eq 4.3*)
  (* converts a REAL number nR into an INTEGER number 'nI' for stochastic
  simulations:
    nI := TRUNC(nR), if U() > nR - TRUNC(nR)
    nI := TRUNC(nR)+1, if U() ≤ nR - TRUNC(nR) *)
PROCEDURE Torus(VAR plNo: IntVector; size: IntVector; VAR inside: BOOLEAN);
  (* returns the new position and a BOOLEAN indicating, if the original position
  was inside the field *)
PROCEDURE Logit(m: Model; VAR mv: REAL; x: REAL);
  (* returns the logit value of mv. In the case, where mv <= 0 OR mv >= 1,
  the values are set so that they fall outside the MW-Graph *)
END PAVAux.
```

PAVSSENSANA [4] D

```
DEFINITION MODULE PAVSensAna;
```

```
*****
Purpose
  execution of a stochastic sensitivity analysis with 2 options:
    - changing of single parameters          (SAMode=singlePars)
    - changing of all parameters simultaneously (SAMode=allPars)

Programming
  • Design:      T. Nemecek   01.03.91
  • Implementation: T. Nemecek   01.03.91

  Last revision of definition: 03.06.91 TN
*****)
FROM SimBase IMPORT Model;
TYPE
  RealPtr = POINTER TO REAL;
  Str31 = ARRAY [0..31] OF CHAR;
  SAProc = PROCEDURE;
  SAMode = (singlePars, allPars);
  Distribution = (uniformDistr, normDistr, triangDistr, special1, special2);
  (* uniformDistr: U[min,max] uniform
     normDistr: N(mean,stdev) normal
     triangDistr: triangular distribution with mode=mean
                   and range [min,max]*)
```

(* collecting output variable values, previously declared with 'AddSAOutVar' *)

```
PROCEDURE CollectOutValue (index: INTEGER);
PROCEDURE CollectOutValues; (* all OutVars *)
  (* reset statistics for OutVars *)
PROCEDURE ZeroOutVarSum (index: INTEGER);
PROCEDURE ZeroOutVarSums; (* all OutVars *)
  (* get statistics for OutVar *)
PROCEDURE GetOutVarSum (index: INTEGER; VAR sum: REAL);

(* management of lists of parameter, for which a sensitivity
analysis is to be carried out. *)
PROCEDURE AddParToSAParList(m: Model; VAR par: REAL; ident: Str31;
  mean,min,max,stDev: REAL);
  (* the parameter 'par' is varied using a distribution with the parameters
  mean,min,max and stDev. The actual distribution depends on the choice in
  'SetDistribution' *)
PROCEDURE RemoveParFromSAParList(VAR par: REAL);
PROCEDURE ClearSAParList;

(* management of output variables lists *)
PROCEDURE AddSAOutVar(VAR outVar: REAL; VAR index: INTEGER; ident: Str31);
PROCEDURE GetSAOutVar(index: INTEGER; VAR ov: RealPtr);
PROCEDURE RemoveSAOutVar(index: INTEGER);
PROCEDURE ClearSAOutVarList;

PROCEDURE SetSAProcs( init1, init2, init3, term3, term2, term1: SAProc);
PROCEDURE GetSAProcs(VAR init1, init2, init3, term3, term2, term1: SAProc);
(* installation of client procedures at 3 levels. An experiment with
'DoSensAna' looks as follows:
  VAR n1,n2: INTEGER;
  init1;
  FOR n2:= 1 TO nRuns2 DO
    init2;
    FOR n1:= 1 TO nRuns1 DO
      init3;
      SimMaster.SimRun;
      term3;
    END(*FOR*);
    term2;
  END(*FOR*);
  term1; *)
```

```
PROCEDURE SetDistribution( distr: Distribution); (* default = normal *)
PROCEDURE GetDistribution(VAR distr: Distribution);

PROCEDURE DoSensAna(sAMode: SAMode; nRuns1, nRuns2: INTEGER);
(* 'nRuns1' is the number of simulation runs, performed with same parameter values
in stochastic models. It should be set to 1 for deterministic models
'nRuns2' is the number of simulation experiments with different parameter
values of 1 parameter. The parameter value is determined for each run sampling
from the chosen distribution *)
END PAVSensAna.
```

General Input and Validation Modules

PAVVALID [5] D

```

DEFINITION MODULE PAVValid;
(* **** **** **** *)
Purpose
  This model is part of the PAV-Simulation-System. Stores and displays
  epidemic validation data and stores some variables used by several models
  modelID = 0

Programming
  • Design:          T. Nemecek   27.09.90
  • Implementation: T. Nemecek   27.09.90

  Last revision of definition: 22.01.92 tn
(* **** **** **** *)
FROM SimBase      IMPORT Model,AuxVar,Parameter;
FROM DFDData      IMPORT DataFrame;

VAR
  validM:           Model;
(* MV's *)
  infPlants,infLeaves,infTubers:     AuxVar;
  year,site,plotNo,variet,y,virus,
  (* pseudo parameters for documentation purposes *)
  emergD,killD:                  Parameter;
  (* dates *)

PROCEDURE DisplayValidEpiData;
  (*displays the epidemic validation time series in the MW-graph *)
PROCEDURE SetValidEpiData (df: DataFrame; ident: ARRAY OF CHAR);
  (*retrieves the epidemic validation data from the currently active data frame *)

PROCEDURE DeclValid;
PROCEDURE UndeclValid;
END PAVValid.
```

WEATHER [6] D

```

DEFINITION MODULE Weather;
(* **** **** **** *)
Purpose: reads weather data from files, stores them in memory
            and makes them accessible by other submodels.

Programming
  • Design:          O.Roth      16.08.89
  • Implementation: O.Roth, T.Nemecek 16.08.89

  Last revision of definition: 11.05.92 tn
(* **** **** **** *)
FROM SimBase IMPORT Model,AuxVar,Parameter;
FROM DFDData IMPORT DataFrame;

VAR
  wModel: Model;
  DAYNR:  (* julian date      [ d      ] *) INTEGER;
  DAY:    (* julian date      [ d      ] *) ;
  TMIN:   (* min temperature  [ °C     ] *) ;
  TMAX:   (* max temperature  [ °C     ] *) ;
  RH:     (* rel humidity     [ %     ] *) ;
  PREC:   (* precipitation    [ mm    ] *) ;
  IRRIG:  (* irrigation       [ mm    ] *) ;
  RADG:   (* radiation        [ J/cm2/d ] *) ;
  VEN:    (* wind speed       [ m/s   ] *) AuxVar;
  Z:      (* anemometer height [ m     ] *) Parameter;

PROCEDURE SetWeatherData(df: DataFrame; ident: ARRAY OF CHAR);
  (* retrieves the weather data to the selected data frame *)

PROCEDURE DeclWeatherModel;
PROCEDURE UndeclWeatherModel;
END Weather.
```

PAVSUCTTRAP [7] D

```

DEFINITION MODULE PAVSuctTrap;
(* **** **** **** **** **** **** **** **** *)
    Purpose
        reads and stores the number of aphids caught in the suction trap

    Programming
        : Design: T. Nemecek 21.03.90
        : Implementation: T. Nemecek 21.03.90

        Last revision of definition: 08.05.92 tn
(* **** **** **** **** **** **** **** *)
FROM DFData IMPORT DataFrame;
FROM SimBase IMPORT Model,AuxVar;

VAR suctTrapM:           Model;
totAphNum,cumTotAphNum: AuxVar;

PROCEDURE ReadSuctTrapData(df: DataFrame; ident: ARRAY OF CHAR);
    (* retrieves the suction trap data from data frame 'df'*)

PROCEDURE GetAphidNumber(VAR aphidNum: ARRAY OF REAL);
    (* gives an array of aphid numbers for the current time step. The species are
     given in the same order as listed in the type declaration of the type 'AphSpec'
     in the module "PAVTypes". *)
PROCEDURE DeclSuctTrap;
PROCEDURE UndeclSuctTrap;
END PAVSuctTrap.

```

EPOVIR modules

SOILWAT [8] DI (SOIL WATER BALANCE SUBMODEL OF EPOVIR)

```

DEFINITION MODULE SoilWat;
(* **** **** **** **** **** **** **** **** *)
    Purpose: soil water balance model of EPOVIR

    Remarks: most elements from PENNING DE VRIES & VAN LAAR, 1982, and
             VAN KEULEN & WOLF, 1986.

    Programming
        * 1st implementation:
            J. Derron, RAC, CH-1260 Nyon. Jul. 1987
        * implementation for ModelWorks:
            T. Nemecek & O. Roth Oct. 1987

        Last revision of definition: 10.11.92 tn
(* **** **** **** **** **** **** **** *)
FROM SimBase IMPORT Model,AuxVar,StateVar,NewState,Derivative;

VAR
    soilWatM: Model;

(* state variables of soil model *)
    WR: StateVar;
    WRNew: NewState;      (* AMOUNT OF WATER IN THE ROOT ZONE      [ mm ] *)
    dWR: Derivative;     (* rate of WR-change *)
    SSTO: StateVar;
    SSTONew: NewState;   (* ACTUAL SURFACE STORAGE          [ mm ] *)
    dSSTO: Derivative;   (* CHANGE IN SURFACE STORAGE       [ mm ] *)

(* output variable of soil model *)
    wStress: AuxVar;     (* WATERSTRESS                  [ mm/mm ] *)

(* monitoring variable of soil model *)
    kPSI: AuxVar;        (* ACTUAL HYDRAULIC CONDUCTIVITY [ cm/d ] *)

PROCEDURE SetPAI(pai: REAL);

PROCEDURE DeclSoilWaterModel;
PROCEDURE UndeclSoilWaterModel;
END SoilWat.

```

IMPLEMENTATION MODULE SoilWat;

(* This model was written by J. Derron in July 1987 in FORTRAN. It was transferred by Thomas Nemecik to Modula-2 and implemented by the means of ModelWorks 0.6 and Dialog Machine 1.0 in May, 1988. The aim of this model is to compute a water-stress factor from the daily values of precipitation.

Original comment:
 THIS SUBROUTINE COMPUTES THE WATER BALANCE OF THE SOIL AND A WATER STRESS FACTOR
 FOR THE CROP. IT ALSO COMPUTES AUXILIARY VARIABLES AS :

- ASTRONOMICAL DAYLENGTH
- DAYLENGTH EFFECTIVE FOR PHOTOSYNTHESIS (solar height 8 degrees)
- " " " PHOTOPERIODISM (" " -4 ")
- GLOBAL RADIATION ON A STANDARD CLEAR DAY
- NET RADIATION
- FRACTION OF THE DAY THE SKY IS OVERCAST

 FOLLOWING METEOROLOGICAL DATA ARE REQUIRED:

- JULIAN DATE	[-]
- MINIMUM TEMPERATURE	[°C]
- MAXIMUM TEMPERATURE	[°C]
- RELATIVE HUMIDITY	[%]
- PRECIPITATION	[mm]
- IRRIGATION	[mm]
- GLOBAL RADIATION	[J/cm ² /d] (Wh/m ² in InCli files)
- WIND SPEED	[m/s]

THE MODEL IS BASED ON:

FOR THE EVAPOTRANSPIRATION AND THE RAIN INTERCEPTION

- PENNING DE VRIES, F.W.T. AND H.H. VAN LAAR, (EDS), 1982. 'SIMULATION OF PLANT GROWTH AND CROP PRODUCTION'. PUDOC, WAGENINGEN, 308P.
 - SCHROEDER, H., 1985. 'VERDUNSTUNG'. SPRINGER VERLAG, 186P. THE SOIL WATER BALANCE AND THE WATER STRESS CONCEPT ARE ADAPTED FROM:
 - VAN KEULEN, H. AND J. WOLF (EDS), 1986. 'MODELLING OF AGRICULTURAL PRODUCTION: WEATHER, SOILS AND CROPS'. PUDOC, WAGENINGEN, 479P. *)

```

FROM PAVBase           IMPORT  int,not,noG,noF,DeclIndM,DeclIndSV,
SetDefltCurveAttrForExMV,DeclIndP,DeclIndM,johnsonMin,sdem,demo;
FROM SimBase           IMPORT  IntegrationMethod,Tabulation,Graphing,RICType,Model,
Stain,LineStyleStyle,RemoveMV,NoInput,NoOutput,NotTerminate,Parameter,AuxVar;
FROM DMWindflowIO      IMPORT  WriteString,WriteLn;
FROM Mathlib            IMPORT  Exp,Sqrt,Ln,Sin,Cos,ArcTan;
FROM MathProcs          IMPORT  ArcSin,Tan,Power,Rmin;
FROM Weather           IMPORT  DAY,TMIN,TMAX,RADG,RH,PREC,IRIG,VEN,Z;

```

```
CONST  
    PI           = 3.14159;  
    RADIANT     = PI/180.0;  
    mind        = johnsonMind;
```

VAR (* input variable (from crop model): *)
PAI: (* PLANT AREA INDEX [m2/m2] *)

```

(* field specific model parameter: *)
LAT,      (* LATITUDE                                [ °N ] *)
ALB,      (* ALBEDO                                     [ - ] *)
ALT,      (* ALTITUDE                                    [ M° ] *)
DELT,     (* CLOUD/FURROW ANGLE                          [ ° ] *)
hypETP,   (* hypothetic evapotranspiration mult. factor [ %/100 ] *)
PHI,      (* SLOPE OF THE FIELD                           [ rad ] *)
PRS,      (* PROPORTION OF STONES                         [ * ] *)
RD,       (* ROOTING DEPTH                               [ cm ] *)
SL,       (* SLOPE OF THE FIELD                           [ * ] *)
SROU,    (* SURFACE ROUGHNESS                          [ - ] *)
ZT,       (* GROUND WATER TABLE                         [ cm ] *)

InitWR,  (* initial soil water status                  [ - ] *)
soilNr,   (* soil type (selector variable)             [ - ] *)
Crop,     (* crop type (selector var: 1=potato, 2=cereals) [ - ] *)

(* soil specific parameters *)
ALP,      (* TEXTURE SPECIFIC CONSTANT                  [ cm2.4/d ] *)
ALPH,     (* "          "                                [ 1/cm ] *)
GAM,      (* TEXTURE SPECIFIC GEOMETRY FACTOR          [ 1/cm2 ] *)
PSImax,   (* TEXTURE SPECIFIC SUCTION LIMIT            [ cm ] *)
SMO,      (* TOTAL PORE SPACE (SATURATION)              [ cm3/cm3 ] *)
SO,       (* STANDARD SORPTIVITY                      [ cm/d0.5 ] *)
TZP,      (* TRANSMISSION ZONE PERMEABILITY           [ cm/d ] *)
ZK,       (* SATURATED HYDRAULIC CONDUCTIVITY         [ cm/d ] *)

Parameter;

```

```
(* auxiliary variables: *)
Cfao,
COSLD, (* AUX. VARIABLE FOR COMPUTING DAYLENGTH [ - ] *)
CR, (* CAPILLARY RAISE [ mm/d ] *)
DEC, (* DECLINATION OF SUN [ ° ] *)
DELTA, (* SLOPE OF VAPOUR PRESSURE-TEMPERATURE CURVE [ mb/oC ] *)
DLA, (* ASTRONOMIC DAYLENGTH [ h ] *)
DRC, (* GLOBAL RADIATION ON STANDARD CLEAR DAY [ J/cm2/d ] *)
EA, (* ACTUAL EVAPORATION [ mm/d ] *)
eAlpha,
eAlpha2, (* both auxillary for CR *)
EAMAX, (* SATURATION VAPOUR PRESSURE AT MAX DAILY
        * AIR TEMPERATURE [ mb ] *)
EAMIN, (* SATURATION VAPOUR PRESSURE AT MIN DAILY
        * AIR TEMPERATURE [ mb ] *)
EAS, (* SATURATION VAPOUR PRESSURE [ mb ] *)
ED, (* actual vapour pressure [ mb ] *)
EM, (* MAXIMUM EVAPORATION [ mm/d ] *)
EWSUP, (* ACTUAL WATER SUPPLY [ mm/d ] *)
EQRAD, (* EVAPORABLE WATER EQUIVALENT(1J/cm2/245:=1 mm) [ mm ] *)
EORDG,
ERD, (* ACTUAL ROOTING DEPTH [ mm ] *)
ETP, (* POTENTIAL EVAPOTRANSPIRATION (PENMAN) [ mm/d ] *)
ETR, (* REAL EVAPOTRANSPIRATION (PENMAN) [ mm/d ] *)
EVAPD, (* ETP DUE TO DRYING POWER OF THE AIR [ mm/d ] *)
EVAPR, (* ETP DUE TO RADIATION [ mm/d ] *)
FOV, (* FRACTION OF THE DAY THAT SKY IS OVERCAST [ - ] *)
FU, (* turbulence resistance [ - ] *)
GAMMA, (* PSYCHROMETER CONSTANT [ mb/oC ] *)
HL, (* LATENT HEAT OF VAPORIZATION [ cal/g ] *)
P, (* BAROMETRIC PRESSURE [ mb ] *)
PERC, (* PERCOLATION [ mm/d ] *)
PSI, (* MATRIC HEAD IN ROOT ZONE [ cm ] *)
RA, (* REDUCTION FACTOR (RADIATION ABSORBED BY PL.) [ - ] *)
RADL, (* LONG WAVE RADIATION [ J/cm2/d ] *)
RADN, (* NET RADIATION [ J/cm2/d ] *)
RDN, (* AMOUNT OF RADIATION THAT WOULD REACH THE
      * EARTH IN THE ABSENCE OF ATMOSPHERE [ J/cm2/d ] *)
RIN, (* ACTUAL INFILTRATION [ mm/d ] *)
RINmax, (* MAXIMUM INFILTRATION [ mm/d ] *)
SINLD, (* AUX. VARIABLE FOR COMPUTING DAYLENGTH [ - ] *)
SOR, (* ACTUAL SORPTIVITY [ cm/d0.5 ] *)
SM, (* MOISTURE CONTENT IN THE ROOT ZONE [ cm3/cm3 ] *)
SMA, (* MOISTURE CONTENT OF AIR-DRY SOIL [ cm3/cm3 ] *)
SMCR, (* CRITICAL SOIL MOISTURE CONTENT [ cm3/cm3 ] *)
SMF, (* MOISTURE CONTENT AT FIELD CAPACITY [ cm3/cm3 ] *)
SMW, (* MOISTURE CONTENT AT WILTING POINT [ cm3/cm3 ] *)
SM005, (* MOISTURE CONTENT NEAR SATURATION [ cm3/cm3 ] *)
SSMAX, (* SURFACE STORAGE CAPACITY [ mm ] *)
SRO, (* SURFACE RUN OFF [ mm/d ] *)
TA, (* ACTUAL TRANSPIRATION [ mm/d ] *)
TM, (* MAXIMUM TRANSPIRATION [ mm/d ] *)
TMOY, (* AVERAGE TEMPERATURE [ °C ] *)
VEN2, (* wind speed at 2m [ m/s ] *)
VJ2,
VHCorr, (* wind height correction factor *)
W,
WA, (* MAXIMUM QUANTITY OF AVAILABLE WATER [ mm ] *)
WDF, (* SOIL WATER DEPLETION FRACTION [ - ] *)
WF, (* AMOUNT OF WATER AT FIELD CAPACITY [ mm ] *)
WINT, (* WATER INTERCEPTION [ mm/d ] *)
WO, (* AMOUNT OF WATER AT SATURATION [ mm ] *)
WSUP, (* WATER SUPPLY (PRECIPITATION + IRRIGATION) [ mm ] *)
WW, (* AMOUNT OF WATER AT WILTING POINT [ mm ] *)
withVWM:
AuxVar;
```

```
PROCEDURE InterP( x, x1, x2, y1, y2: REAL ): REAL;
(* interpolates linearly inbetween x1 and x2, limits extrapolation to boundary
   values y1 or y2 respectively. *)
BEGIN
```

```
  IF (x <= x1) THEN RETURN y1;
  ELSEIF (x >= x2) THEN RETURN y2;
  ELSE (*x1 < x < x2*) RETURN (x-x1)/(x2-x1)*(y2-y1) + y1;
END(*IF*);
END InterP;
```

```
PROCEDURE RLIMIT( xin, xdown, xup : REAL ) : REAL;
BEGIN
  IF xin < xdown THEN RETURN xdown;
  ELSEIF xin > xup THEN RETURN xup;
  ELSE RETURN xin;
END(*IF*);
END RLIMMIT;
```

```

PROCEDURE ShowModelInfo;
BEGIN
  WriteLn;
  WriteString(' Water Balance in Soils V0.5'); WriteLn;
  WriteLn;
  WriteString(' The parameter "soilNr" may take one'); WriteLn;
  WriteString(' of the following codes:'); WriteLn;
  WriteString(' 0 = loamy sand'); WriteLn;
  WriteString(' 1 = sandy loam'); WriteLn;
  WriteString(' 2 = silt loam'); WriteLn;
  WriteString(' 3 = loam'); WriteLn;
  WriteString(' 4 = sandy clay loam'); WriteLn;
  WriteString(' 5 = clay loam'); WriteLn;
  WriteString(' 6 = silty clay loam'); WriteLn;
  WriteString(' 7 = light clay'); WriteLn;
  WriteString(' 8 = silty clay'); WriteLn;
  WriteString(' 9 = heavy clay'); WriteLn;
END ShowModelInfo;

PROCEDURE Initial;
  VAR ws,AA,B,C: REAL; gr: INTEGER;
BEGIN
  (* soil specific constants *)
  IF (soilNr < 0.5) THEN (* loamy sand: *)
    SMO := 0.439; GAM := 0.0330; PSImax:= 200.0; ZK := 26.5;
    ALP := 16.4; ALPH := 0.0398; SO := 19.2; TZP := 17.8;
  ELSIF (soilNr < 1.5) THEN (* sandy loam: *)
    SMO := 0.504; GAM := 0.0207; PSImax:= 290.0; ZK := 12.0;
    ALP := 26.5; ALPH := 0.0248; SO := 17.57; TZP := 9.36;
  ELSIF (soilNr < 2.5) THEN (* silt loam: *)
    SMO := 0.509; GAM := 0.0185; PSImax:= 300.0; ZK := 6.5;
    ALP := 47.3; ALPH := 0.02; SO := 14.46; TZP := 5.32;
  ELSIF (soilNr < 3.5) THEN (* loam: *)
    SMO := 0.503; GAM := 0.018; PSImax:= 300.0; ZK := 5.0;
    ALP := 14.4; ALPH := 0.0231; SO := 11.73; TZP := 3.97;
  ELSIF (soilNr < 4.5) THEN (* sandy clay loam: *)
    SMO := 0.432; GAM := 0.0096; PSImax:= 200.0; ZK := 23.5;
    ALP := 33.6; ALPH := 0.0353; SO := 19.05; TZP := 16.51;
  ELSIF (soilNr < 5.5) THEN (* clay loam: *)
    SMO := 0.445; GAM := 0.0058; PSImax:= 300.0; ZK := 0.98;
    ALP := 1.69; ALPH := 0.0248; SO := 4.; TZP := 0.76;
  ELSIF (soilNr < 6.5) THEN (* silty clay loam: *)
    SMO := 0.475; GAM := 0.0105; PSImax:= 170.0; ZK := 1.5;
    ALP := 36.0; ALPH := 0.0237; SO := 6.15; TZP := 1.18;
  ELSIF (soilNr < 7.5) THEN (* light clay: *)
    SMO := 0.453; GAM := 0.0085; PSImax:= 300.0; ZK := 3.5;
    ALP := 55.6; ALPH := 0.0174; SO := 10.74; TZP := 2.94;
  ELSIF (soilNr < 8.5) THEN (* silty clay: *)
    SMO := 0.507; GAM := 0.0065; PSImax:= 50.0; ZK := 1.3;
    ALP := 28.2; ALPH := 0.048; SO := 3.98; TZP := 0.8;
  ELSE (*soilNr = 9.0*) (* heavy clay: *)
    SMO := 0.54; GAM := 0.0042; PSImax:= 80.0; ZK := 0.22;
    ALP := 4.86; ALPH := 0.038; SO := 1.93; TZP := 0.15;
  END (*IF*);

  (* SSMAX (surface storage capacity): *)
  PHI := ArcTan(SL/100.0);
  AA := 0.5*SRQOU*Sin(DELT-PHI)*Sin(DELT-PHI)/Sin(DELT);
  B := 1.0/Tan(DELT+PHI)+1.0*Tan(DELT-PHI);
  C := 2.0*Cos(DELT)*Cos(PHI);
  SSMAX := AA*B/C;

  SMO05:= SMO-0.05;
  SMF := SMO*Exp(-GAM*Ln(100.0)*Ln(100.0));
  SMW := SMO*Exp(-GAM*Ln(16000.0)*Ln(16000.0));
  SMA := 0.33*SMW;
  ERD := (RD-(RD/100.0*PRS))*10.0;
  WO := ERD*SMO;
  WF := ERD*SMF;
  WW := ERD*SMW;
  WA := WF-WW;

  (* Initialization of state variables: *)
  IF InitWR < 1.5 THEN (* dry soil *)
    WR := WW+0.4*WA;
    SSTO := 0.0;
  ELSIF InitWR > 2.5 THEN (* medium wet soil *)
    WR := WW+0.8*WA;
    SSTO := 0.0;
  ELSE (* InitWR > 2.5 *) (* wet soil *)
    WR := WW+WA+0.5*(WO-WF);
    SSTO := 0.0;
  END;

```

```

SM := WR/ERD;
PSI := Exp(Sqrt((-Ln(SM/SM0))/GAM));
P := 1013.0-0.1055*ALT;
vHCorr:= Exp(0.2*Ln(2.0/Z));
wStress := 1.0;
END Initial;

PROCEDURE Dynamic;
  VAR sico: REAL;
BEGIN
(* calculate the fraction of the day, when the sky is overcast (FOV) *)
  DEC := -23.4*Cos(2.0*PI*(DAY+10.0)/365.0);
  SINLD:= Sin(DEC*RADIAN)*Sin(LAT*RADIAN);
  COSLD:= Cos(DEC*RADIAN)*Cos(LAT*RADIAN);
  sico := SINLD/COSLD;
  DLA := 12.0*(PI*2.0*Arcsin(sico))/PI;
  RDN := 500.0*(SINLD*DLA/24./PI*COSLD*Sqrt(1.0-sico*sico));
  DRC := 0.936*RDN*Exp(-0.1/(RDN/(DLA*500.0)));
  FOV := RLlimit((DRC-RADG)/(0.8*DRC), 0.0, 1.0);

(* ETP (potential evapotranspiration after PENMAN): *)
  TMOY := (TMIN+TMAX)/2.0;
  HL := 595.0-0.51*TMOY;
  GAMMA:= 0.386*P/HL;
  DELTA:= 2.0*Power(0.00738*TMOY+0.8072, 7.0) -0.00116;
  W := DELTA/(DELTA+GAMMA);
  EAMIN:= Exp(0.07159*TMIN -0.000238*TMIN*TMIN +1.81528);
  EAMAX:= Exp(0.07159*TMAX -0.000238*TMAX*TMAX +1.81528);
  EAS := (EAMIN+EAMAX)/2.0;
  ED := EAS*RH/100.0;

  RADL := 4.2*1.17E-7*Power(TMOY+273.0, 4.0)
    *(0.38-0.035*Sqrt(ED))*(1.0-0.9*FOV); (* eqn 9 *)
  RADN := RADG*(1.0-ALB)-RADL; (* eqn 10 *)
  EQRDG:= RADG/245.0;
  EQRAD:= RADN/245.0;
  VEN2 := VEN*vHCorr;
  VJ2 := VEN2*1.2;
  Cfao := 0.867+0.023*EQRDG-0.036*VJ2;
  FU := 0.27*(1.0+VEN2*86.4/100.0);

  EVAPR:= W*EQRAD;
  EVAPD:= (1.0-W)*FU*(EAS-ED);

  ETP := (EVAPR + EVAPD)*Cfao *hypETP;

(* EM (max. evaporation) and TM (max. transpiration): *)
  RA := 1.0-Exp(-0.6*PAI); (* eqn 37ff *)
  EM := ETP *(1.0-RA); (* eqn 69 *)
  TM := ETP * RA; (* eqn 70b *)

(* EA (actual evaporation): *)
  EA := EM*(SM-SMA)/(SM0-SMA); (* eqn 39 *)

(* WDF is crop specific, see VanKeulen & Wolf, table 20, *)
(* (polynomials fitted from that table). *)
  IF (Crop > 1.5) THEN (* wheat type: *)
    WDF := 1.0 -0.050348*TM -0.009305*TM*TM +0.000886*TM*TM*TM;
  ELSE (* potato type: *)
    WDF := 1.0 -0.141523*TM +0.007160*TM*TM;
  END(*IF*);
  WDF := RLlimit(WDF, 0.0, 1.0);

(* TA (actual transpiration): *)
  SMCR := (1.0-WDF)*(SMF-SMW)+SMW; (* eqn 52, 71 *)
  (* SM005 > SMF > SMCR > SMW *) (* eqn 55 *)
  IF SM > SM005 THEN TA := 0.0;
  ELSIF SM > SMF THEN TA := TM*(SM005-SM)/(SM005-SMF);
  ELSIF SM > SMCR THEN TA := TM;
  ELSIF SM > SMW THEN TA := TM*(SM-SMW)/(SMCR-SMW);
  ELSE (*SM < SMW *) TA := 0.0;
  END;

  ETR := TA+EA;

(* EWSUP (effective water supply): *)
  WSUP := PREC + IRIG ;
  WINT := -0.42 +0.245*WSUP +0.2*PAI +0.0271*WSUP*PAI- 0.0111*WSUP*WSUP
    -0.0109*PAI*PAI;
  WINT := RLlimit(WINT, 0.0, WSUP);

  EWSUP := WSUP-WINT;

```

```

(* PERC (percolation) and capillary raise: *)
  IF PSI > PSImax
    THEN kPSI := ALP*Exp(Ln(PSI)*(-1.4)); (* eqn 31 *)
  ELSE eAlpha:= Exp(-ALPH*PSI); kPSI := ZK * eAlpha END;
  CR:= 0.0;

  PERC := kPSI*10.0;
  IF (WR > WF)      THEN PERC := Rmin(PERC, WR-WF) END(*IF*);
  IF withVWM <= 0.1 THEN PERC := 0.0          END(*IF*);

(* RINmax (max. infiltration): *)
  SOR := SO*(1.0-SM/SM0); (* eqn 33 *)
  RINmax:= (SOR+T2P)*10.0; (* eqn 34 *)
  RINmax:= Rmin(RINmax, WO-WR);

  IF EWSUP>RINmax THEN
    RIN := RINmax;
    IF SSTO>0.0 THEN
      IF EWSUP+SSTO-RINmax>=SSMAX
        THEN SSTONew := SSMAX; SRO := EWSUP+SSTO-RINmax-SSMAX;
        ELSE SSTONew := EWSUP+SSTO-RINmax; SRO := 0.0;
      END(*IF*);
    ELSE
      IF EWSUP-RINmax>=SSMAX
        THEN SSTONew := SSMAX; SRO := EWSUP-RINmax-SSMAX;
        ELSE SSTONew := EWSUP-RINmax; SRO := 0.0;
      END(*IF*);
    END(*IF*);
    ELSE (* EWSUP<=RINmax *)
      SRO := 0.0;
      IF SSTO>0.0 THEN
        IF EWSUP+SSTO<=RINmax
          THEN RIN := EWSUP+SSTO; SSTONew := 0.0;
          ELSE RIN := RINmax; SSTONew := EWSUP+SSTO-RINmax;
        END(*IF*);
        ELSE RIN := EWSUP; SSTONew := 0.0;
      END(*IF*);
    END(*IF*);
  END(*IF*);

(* wStress (Water supply/demand ratio): *)
  IF (PAI > 0.0) THEN wStress := TA/TM;
  ELSE wStress := 0.0;
END(*IF*);

(* update new states: *)
  dWR := RIN - EA - TA - PERC + CR;
  WRNew := WR + dWR;
  SM := WRNew/ERD;
  IF (SM > SM0) OR (SM <= 0.0) THEN HALT; END(*IF*);
  PSI := Exp(Sqrt(-Ln(SM/SM0)/GAM)); (* inverse of eqn. 27 *)
END Dynamic;

PROCEDURE ModelObjects;
BEGIN
(* State variables: *)
  DeclIndSV(WR,WRNew,0.0,0.0,1000.0,'water in the root zone','WR','mm',mInd);
  DeclIndMV(WR,0.0,200.0,'water in the root zone','WR','mm',
  noF,noT,isty,mInd,sDem);
  DeclIndMV(dWR,-10.0,10.0,'waterchange in the root zone','dWR','mm/d',
  noF,noT,noG,mInd,demo);
  SetDefltCurveAttrForExMV(soilWatM, WR, sapphire, spotted, 'o');
  DeclIndSV(SSTO,SSTONew,0.0,0.0,10000.0,'surface storage','SSTO','mm',mInd);
  DeclIndMV(SSTO,0.0,100.0,'surface storage','SSTO','mm,noF,noG,mInd,demo');
  DeclIndMV(dsSSTO,-5.0,5.0,'surface storage change','dsSSTO','mm/d',
  noF,noT,noG,mInd,demo);

(* Monitoring variables: *)
  DeclIndMV(wStress,0.0,1.0,'water stress', 'wStress','-',
  noF,noT,isty,mInd,sDem);
  DeclIndMV(RIN,-50.0,10.0,'actual water influx', 'RIN','mm/d',
  noF,noT,noG,mInd,demo);
  DeclIndMV(WSUP,0.0,50.0,'water supply', 'WSUP','mm/d',
  noF,noT,noG,mInd,demo);
  DeclIndMV(EA,0.0,1.0,'actual evaporation', 'EA','mm/d',
  noF,noT,noG,mInd,demo);
  DeclIndMV(EM,0.0,1.0,'maximal evaporation', 'EM','mm/d',
  noF,noT,noG,mInd,demo);
  DeclIndMV(TA,0.0,1.0,'actual transpiration', 'TA','mm/d',
  noF,noT,noG,mInd,demo);
  DeclIndMV(TM,0.0,1.0,'maximal transpiration', 'TM','mm/d',
  noF,noT,noG,mInd,demo);
  DeclIndMV(SRO,0.0,10.0,'surface run-off', 'SRO','mm/d',
  noF,noT,noG,mInd,demo);

```

```

DeclIndMV(ETP, 0.0,10.0,'pot. evapo-transpiration', 'ETP','mm/d',
  noF,noG,mInd,demo);
DeclIndMV(EVAPR, 0.0,10.0,'ETP due to radiation', 'EVAPR','mm/d',
  noF,noT,noG,mInd,demo);
DeclIndMV(EVAPD, 0.0,10.0,'ETP due to dry air', 'EVAPD','mm/d',
  noF,noT,noG,mInd,demo);
DeclIndMV(SM, 0.0,10.0,'soil moisture in root zone', 'SM','cm/cm',
  noF,noT,noG,mInd,demo);
DeclIndMV(RINmax, 0.0,60.0,'maximum infiltration', 'RINmax','mm/d',
  noF,noT,noG,mInd,demo);
DeclIndMV(PERC, 0.0,60.0,'percolation', 'PERC','mm/d',
  noF,noT,noG,mInd,demo);
DeclIndMV(CR, 0.0,60.0,'capillary raise', 'CR','mm/d',
  noF,noT,noG,mInd,demo);
DeclIndMV(WINT, 0.0,50.0,'water intercepted', 'WINT','mm/d',
  noF,noT,noG,mInd,demo);
DeclIndMV(kPSI, 0.0,30.0,'actual hydraulic conductivity', 'kPSI','cm/d',
  noF,noT,isy,mInd,demo);
DeclIndMV(PSI, 0.0,2000.0,'matric suction', 'PSI','cm',
  noF,noT,noG,mInd,demo);

(* Parameters of soil, field situation: *)
DeclIndP(soilNr,3.0,0.0,9.0,noRtc,'l=loamy sand,9=heavy clay', 'NOR','-',
  mInd,sDem);
DeclIndP(InitWR,2.0,1.0,3.0,noRtc,'1=dry; 2=medium; 3=wet', 'InitWR','-',
  mInd,sDem);
DeclIndP(hypETP,1.0,0.0,2.0, rtc,'hypoth. ETP mult.F.', hypETP,'%/100',
  mInd,sDem);
DeclIndP(Crop, 1.0,1.0,2.0,noRtc,'1=potato,grape; 2=cereals','Crop','-',
  mInd,sDem);
DeclIndP(SROU,20.0, 0.0,100.0, rtc,'surface roughness','SROU','-',mInd,sDem);
DeclIndP(ZT, 70.0, 0.0,1000.0, rtc,'water table','ZT','cm',mInd,sDem);
DeclIndP(RD, 45.0, 0.0,200.0, rtc,'rooting depth','RD','cm',mInd,sDem);
DeclIndP(PRS, 7.0, 0.0, 50.0,noRtc,'proportion of stones','PRS','%',mInd,sDem);
DeclIndP(ALB, 0.2, 0.0, 1.0, rtc,'albedo','alb','-',mInd,sDem);
DeclIndP(LAT, 46.3, 0.0, 90.0,noRtc,'latitude','LAT','N',mInd,sDem);
DeclIndP(ALT,450.0, 0.0,2.0E3,noRtc,'altitude','ALT','m',mInd,sDem);
DeclIndP(SL, 5.0, 0.0,100.0,noRtc,'slope of the field','SL','%',mInd,sDem);
DeclIndP(DELT,0.524,0.0, 30.0,noRtc,'clod/furrow angle','DELT','-',mInd,sDem);
DeclIndP(withVWM,1.0,0.0, 1.0, rtc,'vert. Water movement flag','withVWM','-',
  mInd,sDem);
END ModelObjects;

PROCEDURE SetPAI(pai : REAL);
BEGIN
  PAI:= pai;
END SetPAI;

PROCEDURE DeclSoilWaterModel;
BEGIN
  DeclIndM(soilWatM,discreteTime,Initial,NoInput,NoOutput,Dynamic,NoTerminate,
    ModelObjects,'Soil-Water-Model','Soil-Water',ShowModelInfo,mInd);
END DeclSoilWaterModel;

PROCEDURE UndeclSoilWaterModel;
BEGIN
  RemoveM(soilWatM);
  END UndeclSoilWaterModel;

BEGIN
  PAI := 0.0;
  wStress := 1.0;
END SoilWat.

```

POTATOMOD [9] DI (PLANT GROWTH SUBMODEL OF EPOVIR)

```

DEFINITION MODULE PotatoMod;
*****
  Purpose
    plant growth submodel of EPOVIR

  Programming
    • Design:
      Markus Ulrich          Aug. 86
    • Implementation:
      Markus Ulrich          Aug. 86
      Andreas Fischlin        May 87
      Olivier Roth            Nov. 89

```

Thomas Nemecek Dec. 90

```

Last revision of definition: 28.12.90 tn
*****)
FROM SimBase    IMPORT Model,Parameter,StateVar,NewState,Derivative,AuxVar;

CONST
  minVariety = 1.0;       maxVariety = 5.0;

VAR
  potatoModel: Model;

  (* outputs *)
  lai;                      (*leaf area index*)
  fRadI;                   (*fract. intercepted radiation*)
  youngLeafTissue:        (*proportion of leaf tissue below a specified threshold*)
  AuxVar;

  (* parameter *)
  leafDwtoArea,            (*leaf dry weight to leaf area conv.factor*)
  rowDist,                 (*distance of rows*)
  plantDist:              (*distance of plants within rows*)
  Parameter;

  (* state variables *)
  pA,                      (*physiological age of plant*)
  leaf,                    (*dry weight of leaves of one plant*)
  stem,                    (*dry weight of stems of one plant*)
  root,                    (*dry weight of roots of one plant*)
  tuber,                   (*dry weight of tubers of one plant*)
  assim:                   (*assimilated dry matter per day*)

  StateVar;
  newPA,newLeaf,newStem,newRoot,newTuber,newAssim;
  NewState;
  dPA,dLeaf,dStem,dRoot,dTuber,dAssim;
  Derivative;

PROCEDURE DeclPotatoModel;
PROCEDURE UndeclPotatoModel;
END PotatoMod.
```

```

IMPLEMENTATION MODULE PotatoMod;
(* The first implementation was realized as a practical work of a post-graduate
   study in automatics under the guidance of Dr. A. Fischlin by Markus M. Ulrich. *)

FROM DMStorage    IMPORT Allocate,Deallocate;
FROM DMWindowIO   IMPORT WriteString,SetPos;
FROM MathLib       IMPORT Exp;
FROM MathProcs     IMPORT Rmax;
FROM Weather       IMPORT TMIN,TMAX,RADG;
FROM SoilWat       IMPORT wStress,SetPAI;
FROM PAValid       IMPORT killID;
FROM PAAux          IMPORT InsuffMem;
FROM PAVBase        IMPORT nof,int,noG,noF,DecIndM,DecIndSV,
SetDefltCurveAttrForExMV,DecIndP,DecIndMV,johnsonMind,sDem,demo;
FROM SimBase        IMPORT StashFilling,Tabulation,Graphing,RICType,SetP,
Stain,LineStyle,RemoveM,CurrentTime,Parameter,AuxVar,IntegrationMethod;

CONST
  mind = johnsonMind;

TYPE
  LeafTrainPointer = POINTER TO LeafTrainElement;
  LeafTrainElement = RECORD
    biomass: REAL; (*leaf biomass*)
    pAge:    REAL; (*physiological age*)
    next:    LeafTrainPointer; (*to next element *)
  END;
  VAR
  *****)
  (* Parameters *)
  ****)
  (* General abbreviations:
  PA/pA:            physiological age
  A/a:              assimilation/ assimilated matter
  L/l:              leaf
  S/s:              stem
  R/r:              root
  T/t:              tuber
  Tmp:              temperature
  d:                "delta" *)
```

```

(** Parameters which depend on the variety: **)
(** Distribution equations: **)
kGrowth,          (*general growth parameter*)
kLeaf,            (*leaf distribution parameter*)
kStem,            (*stem distribution parameter*)
kRoot,            (*root distribution parameter*)
kTuber,           (*tuber distribution parameter*)

(** Proper physiology of the plant: **)
variety,          (* potato variety:
                    * 1.0 = Bintje
                    * 2.0 = Russet Burbank
                    * 3.0 = Norland
                    * 4.0 = Marie Piper
                    * 5.0 = any other .*)

oldVar,
maxPlantPAge,    (*maximal physiological age of plant*)
maxLeafPAge,     (*maximal physiological age of leaves*)
tuberInitPAge,   (*physiological age at tuber initiation*)
tuberInit,        (*minimal value for tuber after tuber init.*)
propInstantUse,  (*proportion of assimilated matter used on the same day*)
leafRecycle,     (*proportion of senesced biomass recycled*)
minGrowthTmp,    (*minimal temperature for plant growth*)
optGrowthTmp,    (*optimal temperature for plant growth*)
maxGrowthTmp,    (*maximal temperature for plant growth*)
averageDTP,      (*mean daily increase of physiol. age*)
stemArea,         (*contribution of stem area to leaf area index*)
maxPotNetGR,     (*maximum potential growth rate of leaves*)
lpAge1,           (*begin of assimilation decline of leaves*)
lpAge2,           (*begin of assimilation optimum of leaves*)

(** Transfer of biomass from seed tubers to plant: **)
pASeedCont,      (*interval, during which seed contribution to plant occurs*)
propBefE,         (*proportion incorporated before tuber init.*)
propafte,         (*proportion incorporated after tuber init.*)
effUtil,          (*proportion incorporated in biomass*)
propInLeaf,        (*proportion incorporated in leaf*)
propInStem,        (*proportion incorporated in stem*)
propInRoot,        (*proportion incorporated in root*)

hypWStrF,          (*hypothetic WStressF (-1=noWStr)*)
leafAgeThresh,    (*threshold used for determination of youngLeafTissue*)
(** conversion factors: **)
tuberDWtoFW,      (*tuber dry weight to fresh weight*)

(** Parameters depending on culture/seed tubers/year: **)
seedWeight,        (*average fresh weight of seed potatoes*)
Parameter;

(* **** Output Variables *)
(* **** Seed tuber contributions: **)
dSeedLeaf,         (*dayly contribution of tuber to leaf*)
dSeedStem,         (*dayly contribution of tuber to stem*)
dSeedRoot,         (*dayly contribution of tuber to root*)

(** Various variables: **)
potNetGrowthRate, (*potential net growth rate*)
dleafSenesc,       (*dayly leaf senescence*)
tuberFW,           (*tuber fresh weight*)
pai,               (*plant area index*)
harvInd,           (*harvest index*)
usedWStress,       (*used water stress after corr. hypWStrF*)
minMoptGrt2,      (*minMoptGrt2, maxMoptGrt2, al, bl, a2, b2, leafMassBelowThresh, deltaDM:*
AuxVar;*
trainEntry, trainEnd;
LeafTrainPointer;

(* **** Initializing procedures:
(* ****)
(* ****)

PROCEDURE ModelObjects;
BEGIN
(* Declarations of the state variables: *)
DeclIndsV(pA ,newPA ,0.0,0.0,1.0,"Phys. age", "pA", "DD", mInd);
DeclIndsV(leaf ,newLeaf ,0.0,0.0,1.0,"Leaf", "Leaf", "g dw/pl",mInd);
DeclIndsV(stem ,newStem ,0.0,0.0,1.0,"Stem", "Stem", "g dw/pl",mInd);
DeclIndsV(root ,newRoot ,0.0,0.0,1.0,"Root", "Root", "g dw/pl",mInd);
DeclIndsV(tuber,newTuber,0.0,0.0,1.0,"Tuber", "Tuber", "g dw/pl",mInd);
DeclIndsV(assim,newAssim,0.0,0.0,1.0,"Assimilates", "Ass.", "g dw/pl",mInd);

```

```

(* Declarations of the output variables: *)
DeclIndMV(pA,0.0,700.0,"Physiological age","pA","DD",onF,inT,noG,mInd,demo);
DeclIndMV(dLeaf, 0.0, 80.0, "Leaf", "leaf", "g dw/pl",onF,inT,isy,mInd,sDem);
DeclIndMV(stem, 0.0, 80.0, "Stem", "stem", "g dw/pl",onF,inT,noG,mInd,sDem);
DeclIndMV(root, 0.0, 10.0, "Root", "root", "g dw/pl",onF,inT,noG,mInd,sDem);
DeclIndMV(tuber, 0.0, 500.0, "Tuber", "tuber", "g dw/pl",onF,inT,isy,mInd,sDem);
DeclIndMV(assim, 0.0, 10.0, "Assimilates", "assim", "g dw/pl",noF,inT,noG,mInd,demo);

DeclIndMV(dPA, 0.0, 10.0, "A phys. age", "dPA", "DD/day", noF, noT, noG, mInd, demo);
DeclIndMV(dLeaf, 0.0, 2.0, "A leaf", "dLeaf", "g dw/pl/day", noF, noT, noG, mInd, demo);
DeclIndMV(dStem, 0.0, 2.0, "A stem", "dStem", "g dw/pl/day", noF, noT, noG, mInd, demo);
DeclIndMV(dRoot, 0.0, 2.0, "A root", "dRoot", "g dw/pl/day", noF, noT, noG, mInd, demo);
DeclIndMV(dTuber, 0.0, 2.0, "A tuber", "dTuber", "g dw/pl/day",
           noF, noT, noG, mInd, demo);
DeclIndMV(dAssim, 0.0, 2.0, "A assim", "dAssim", "g dw/pl/day",
           noF, noT, noG, mInd, demo);

DeclIndMV(tuberFW,0.0,500.0,"Tuber fresh weight","tuberFW","kg fw/a",
           noF,noT,noG,mInd,sDem);

DeclIndMV(dSeedleaf, 0.0, 1.0, "A SeedLeaf", "dSeedLeaf", "g dw/pl/day",
           noF,noT,noG,mInd,demo);
DeclIndMV(dSeedStem, 0.0, 1.0, "A SeedStem", "dSeedStem", "g dw/pl/day",
           noF,noT,noG,mInd,demo);
DeclIndMV(dSeedRoot, 0.0, 1.0, "A SeedRoot", "dSeedRoot", "g dw/pl/day",
           noF,noT,noG,mInd,demo);

DeclIndMV(potNetGrowthRate,0.0,1.0,"Pot. Net Growth Rate","potNetGrowthRate",
           "g dw/NJ/DD", noF, noT, noG, mInd, demo);
DeclIndMV(dLeafSenesc, 0.0, 1.0, "LeafSenesc", "LeafSenence", "g dw/pl",
           noF,noT,noG,mInd,demo);

DeclIndMV(pai, 0.0, 4.0, "PlantAreaIndex", "PAI", "--", noF, noT, noG, mInd, demo);
DeclIndMV(lai, 0.0, 0.7, "LeafAreaIndex", "LAI", "--", noF, noT, noG, mInd, sDem);
DeclIndMV(fRadic, 0.0, 1.0, "Fract InterceptedRad", "fRadic", "%/100",
           noF, noT, noG, mInd, demo);
DeclIndMV(harvInd, 0.0, 1.0, "Harvest index", "harvInd", "g/g",
           noF, noT, noG, mInd, demo);
DeclIndMV(youngLeafTissue, 0.0, 1.0, "Leaf tissue younger than threshold", "ly",
           "%/100", noF, noT, noG, mInd, demo);

(* Curve attributes of the most important monitorable variables: *)
SetDefltCurveAttrForExMV(potatoModel, leaf ,emerald ,broken, 'OC');
SetDefltCurveAttrForExMV(potatoModel, stem ,coal ,dashSpotted, 'OC');
SetDefltCurveAttrForExMV(potatoModel, root ,turquoise,spotted, 'OC');
SetDefltCurveAttrForExMV(potatoModel,tuber,ruby ,unbroken, '+');
SetDefltCurveAttrForExMV(potatoModel,lai ,sapphire ,invisible, 'x');

(* Declarations of the parameters: *)
(* model switches: *)
DeclIndP(variety,oldVar,minVariety,maxVariety,noRtc,
         "Potato variety: 1-Bintje; 2=Russet Burbank; 3-Norland; 4-Maris Piper",
         "variety", "--",mInd,sDem);
DeclIndP(hypWStrF,1.0,0.0,1.0,rtc,"hyp.WStressF (0=noWStr,1=norm)", "hypWStrF",
         "--",mInd,demo);

(* general potato model parameters (for Bintje): *)
DeclIndP(kGrowth,60.0,0.0,1000.0,rtc,"kGrowth", "g dw/pl",mInd,sDem);
DeclIndP(kLeaf, 1.0,0.0, 500.0,rtc, "kLeaf", "kLeaf", "--", mInd,sDem);
DeclIndP(kStem, 15.0,0.0, 500.0,rtc, "kStem", "kStem", "--", mInd,sDem);
DeclIndP(kRoot, 1.0,0.0, 500.0,rtc, "kRoot", "kRoot", "--", mInd,sDem);
DeclIndP(kTuber, 12.0,0.0, 500.0,rtc, "kTuber", "kTuber", "--", mInd,sDem);

DeclIndP(maxPlantPAge,1000.0,0.0,2000.0,noRtc, "maxPlantPAge", "maxPlantPAge",
         "DD",mInd,sDem);
DeclIndP(maxLeafPAge, 600.0,0.0,1000.0,noRtc, "maxLeafPAge", "maxLeafPAge",
         "DD",mInd,sDem);
DeclIndP(tuberInitPAge,120.0,0.0,1000.0,noRtc, "tuberInitPAge", "tuberInitPAge",
         "DD",mInd,sDem);
DeclIndP(tuberInit, 0.5,0.0,10.0,noRtc, "tuberInit", "tuberInit",
         "g dw/pl",mInd,sDem);
DeclIndP(propAInstantUse,0.75,0.0,1.0,rtc,"propAInstantUse", "propAInstantUse",
         "--",mInd,sDem);
DeclIndP(leafRecycle,0.5,0.0,1.0,rtc, "leafRecycle", "leafRecycle", "--",mInd,sDem);

DeclIndP(minGrowthTmp, 4.0,0.0, 50.0,rtc, "minGrowthTmp", "minGrowthTmp",
         "C",mInd,sDem);
DeclIndP(optGrowthTmp,20.0,0.0, 50.0,rtc, "optGrowthTmp", "optGrowthTmp",
         "C",mInd,sDem);
DeclIndP(maxGrowthTmp,30.0,0.0, 50.0,rtc, "maxGrowthTmp", "maxGrowthTmp",
         "C",mInd,sDem);
DeclIndP(averageDtP, 8.0,0.0, 50.0,rtc, "averageDtP", "averageDtP",
         "DD",mInd,sDem);

```

```

DeclIndP(maxPotNetGR, 2.2,0.0, 20.0,rtc,"maxPotNetGR","maxPotNetGR",
  "g/MJ",mInd,sDem);
DeclIndP(lpAge1, 150.0,0.0,300.0,rtc,"lpAge1",      "lpAge1",
  "DD", mInd,sDem);
DeclIndP(lpAge2, 75.0,0.0,200.0,rtc,"lpAge2",      "lpAge2",
  "DD", mInd,sDem);

DeclIndP(paSeedCont,120.0, 0.0,400.0,noRtc,"seed contrib.pA", "paSeedCont",
  "DD", mInd,sDem);
DeclIndP(propBefE, 0.2, 0.0, 1.0, rtc,"prop. before emerg.", "propBefE",
  "DD", mInd,sDem);
DeclIndP(propAftE, 0.65,0.0, 1.0, rtc,"prop. after emerg.", "propAftE",
  "DD", mInd,sDem);
DeclIndP(effUtil, 0.4, 0.0, 1.0, rtc,"effUtil",      "effUtil",
  "DD", mInd,sDem);
DeclIndP(propInLeaf, 0.4, 0.0, 1.0, rtc,"propInLeaf", "propInLeaf",
  "DD", mInd,sDem);
DeclIndP(propInStem, 0.4, 0.0, 1.0, rtc,"propInStem", "propInStem",
  "DD", mInd,sDem);
DeclIndP(propInRoot, 0.2, 0.0, 1.0, rtc,"propInRoot", "propInRoot",
  "DD", mInd,sDem);

DeclIndP(leafDWtoArea,0.023, 0.0, 1.0,rtc,"leafDWtoArea","leafDWtoArea",
  "m^2/g dw", mInd,sDem);
DeclIndP(stemArea, 0.0869,0.0, 1.0,rtc,"stemArea",    "stemArea",
  "DD", mInd,sDem);
DeclIndP(tuberDWtoFW, 5.0, 1.0,10.0,rtc,"tuberDWtoFW", "tuberDWtoFW",
  "g fw/g dw",mInd,sDem);

DeclIndP(leafAgeThresh,100.0,0.0,3000.0,rtc,"age threshold for young leaves",
  "leafAgeThresh","DD",mInd,sDem);

(* situation specific model parameters: *)
DeclIndP(seedWeight,60.0, 0.0,200.0,noRtc,"seedWeight","seedWeight",
  "g fw",mInd,sDem);
DeclIndP(rowDist, 0.75,0.0, 5.0,noRtc,"rowDist",    "rowDist",
  "m", mInd,sDem);
DeclIndP(plantDist, 0.33,0.0, 5.0,noRtc,"plantDist", "plantDist",
  "m", mInd,sDem);
END ModelObjects;

```

```

PROCEDURE SetPForBintje;
BEGIN
  SetP(potatoModel,variety,           1.0); SetP(potatoModel,kGrowth,          60.0);
  SetP(potatoModel,kLeaf,             1.0); SetP(potatoModel,kStem,            15.0);
  SetP(potatoModel,kRoot,             1.0); SetP(potatoModel,kTuber,           12.0);
  SetP(potatoModel,maxPlantPAge,1000.0); SetP(potatoModel,maxLeafPAge,   600.0);
  SetP(potatoModel,tuberInitPAge,120.0); SetP(potatoModel,minGrowthTmp,  4.0);
  SetP(potatoModel,optGrowthTmp, 20.0); SetP(potatoModel,maxGrowthTmp, 30.0);
  SetP(potatoModel,lpAge1,           150.0); SetP(potatoModel,lpAge2,          75.0);
  SetP(potatoModel,paSeedCont, 120.0); SetP(potatoModel,propBefE,         0.2);
  SetP(potatoModel,propAftE,          0.65);SetP(potatoModel,propInLeaf,     0.4);
  SetP(potatoModel,propInStem,        0.4); SetP(potatoModel,propInRoot,     0.2);
  SetP(potatoModel,maxPotNetGR,     2.2);
END SetPForBintje;

```

```

PROCEDURE SetPForRussetBurbank;
BEGIN
  SetP(potatoModel,variety,           2.0); SetP(potatoModel,kGrowth,          50.0);
  SetP(potatoModel,kLeaf,             1.0); SetP(potatoModel,kStem,            6.0);
  SetP(potatoModel,kRoot,             1.0); SetP(potatoModel,kTuber,           6.0);
  SetP(potatoModel,maxPlantPAge,675.0); SetP(potatoModel,maxLeafPAge, 400.0);
  SetP(potatoModel,tuberInitPAge,215.0); SetP(potatoModel,minGrowthTmp, 7.0);
  SetP(potatoModel,optGrowthTmp, 21.0); SetP(potatoModel,maxGrowthTmp, 30.0);
  SetP(potatoModel,lpAge1,           160.0); SetP(potatoModel,lpAge2,          80.0);
  SetP(potatoModel,paSeedCont, 200.0); SetP(potatoModel,propBefE,         0.2);
  SetP(potatoModel,propAftE,          0.6); SetP(potatoModel,propInLeaf,     0.2);
  SetP(potatoModel,propInStem,        0.4); SetP(potatoModel,propInRoot,     0.4);
  SetP(potatoModel,maxPotNetGR,     1.5);
END SetPForRussetBurbank;

```

```

PROCEDURE SetPForNorland;
BEGIN
  SetP(potatoModel,variety,           3.0); SetP(potatoModel,kGrowth,          40.0);
  SetP(potatoModel,kLeaf,             1.0); SetP(potatoModel,kStem,            6.0);
  SetP(potatoModel,kRoot,             1.0); SetP(potatoModel,kTuber,           8.5);
  SetP(potatoModel,maxPlantPAge,630.0); SetP(potatoModel,maxLeafPAge, 350.0);
  SetP(potatoModel,tuberInitPAge,200.0); SetP(potatoModel,minGrowthTmp, 7.0);
  SetP(potatoModel,optGrowthTmp, 21.0); SetP(potatoModel,maxGrowthTmp, 30.0);
  SetP(potatoModel,lpAge1,           160.0); SetP(potatoModel,lpAge2,          80.0);
  SetP(potatoModel,paSeedCont, 200.0); SetP(potatoModel,propBefE,         0.2);
  SetP(potatoModel,propAftE,          0.6); SetP(potatoModel,propInLeaf,     0.2);

```

```

SetP(potatoModel,propInStem,      0.4); SetP(potatoModel,propInRoot,      0.4);
SetP(potatoModel,maxPotNetGR,    1.5);
END SetPForNorland;

PROCEDURE SetPForMarisPiper;
BEGIN
  SetP(potatoModel,variety,        4.0); SetP(potatoModel,kGrowth,       56.0);
  SetP(potatoModel,kLeaf,          1.2); SetP(potatoModel,kStem,         20.0);
  SetP(potatoModel,kRoot,          5.5); SetP(potatoModel,kTuber,        9.0);
  SetP(potatoModel,maxLeafAge,    600.0); SetP(potatoModel,maxPlantPAge, 1000.0);
  SetP(potatoModel,tuberInitAge, 120.0); SetP(potatoModel,minGrowthImp,  4.0);
  SetP(potatoModel,optGrowthImp, 20.0); SetP(potatoModel,maxGrowthImp,  30.0);
  SetP(potatoModel,lpAgeL,        200.0); SetP(potatoModel,lpAge2,       75.0);
  SetP(potatoModel,pASeedCont,   120.0); SetP(potatoModel,propBEI,       0.2);
  SetP(potatoModel,propAftE,     0.6); SetP(potatoModel,propInLeaf,    0.2);
  SetP(potatoModel,propInStem,   0.4); SetP(potatoModel,propInRoot,    0.4);
  SetP(potatoModel,maxPotNetGR,  2.0);
END SetPForMarisPiper;

(*****)
(* Model procedures: *)
(*****)

PROCEDURE CalcSeedTuberContribution;
VAR x: REAL;
BEGIN
  IF pA<=pASeedCont THEN
    (* contributions after emergence up to a page of pAEnd *)
    x:= effUtil * seedWeight / tuberDWtoFW * propAftE * (dPA/pASeedCont);

    dSeedLeaf:= propInLeaf*x; dSeedStem:= propInStem*x; dSeedRoot:= propInRoot*x;
  ELSE
    dSeedLeaf:= 0.0;           dSeedStem:= 0.0;           dSeedRoot:= 0.0;
  END; (*IF*)
END CalcSeedTuberContribution;

PROCEDURE InitializeTuber;
BEGIN
  IF (tuber<tuberInit) AND (pA>=tuberInitPAge) THEN (*initialize tuber:*)
    tuber:= tuberInit;
  END; (*IF*)
END InitializeTuber;

PROCEDURE CalcStateVariableIncrease;
VAR dl,ds,dt,dz, x: REAL;
BEGIN
  (* auxiliary calculations:*)
  lai:= leaf*leafDWtoArea / (rowDist*plantDist);
  pai:= (leaf+stemArea*stem)*leafDWtoArea/(rowDist*plantDist);
  (*contribution of stem to lai.*)

  fRadI:= (1.0-Exp(-pai/2.0));
  deltaDM:= fRadI*potNetGrowthRate*rowDist*plantDist*RADG/100.0*dPA/averageDtP
  *usedWStress;

  (* compute sink strengths *)
  dl:= kLeaf *(kGrowth*(kGrowth+stem+tuber+root));
  dt:= kTuber *(tuber / (kGrowth+tuber));
  ds:= kStem *(stem / (stem+kGrowth*(1.0+tuber)));
  dr:= kRoot *(root / (root+kGrowth*(1.0+tuber)));

  (* compute rate of change based on proportional sink strengths *)
  IF CurrentTime() < killD
  THEN x:= propAInstantUse*usedWStress*(assim+deltaDM)/(dl+dt+ds+dr);
  ELSE x:= 0.0;
  END(*IF*);
  dtuber := dt*x; dLeaf := dl*x; dStem := ds*x; dRoot := dr*x;
  dAssim:= deltaDM+leafRecycle*dLeafSenesc (* production *)
  -dTuber-dLeaf-dRoot-dStem; (* consumption *)
END CalcStateVariableIncrease;

PROCEDURE AddFirstLeafIncreaseToTrain;
BEGIN
  Allocate(trainEntry, SIZE(LeafTrainElement));
  IF trainEntry<>NIL THEN
    WITH trainEntry DO
      biomass:= leaf;
      pAge:= 0.0;
      next:= NIL;
    END; (*WITH*)
    trainEnd:= trainEntry;
  END;

```

```

ELSE
  InsuffMem;
END; (*IF*)
END AddFirstLeafIncreaseToTrain;

PROCEDURE AddLeafIncreaseToTrain;
  VAR t: LeafTrainPointer;
BEGIN
  Allocate(trainEnd^.next, SIZE(LeafTrainElement));
  IF trainEnd^.next <> NIL THEN
    trainEnd:= trainEnd^.next;

    (*add new biomass at the end of the train*)
    WITH trainEnd^ DO
      biomass := dLeaf + dSeedLeaf; pAge := 0.0; next := NIL;
    END; (*WITH*)

    (*add phy. age increase to train age elements and calculate
     leafMassBelowThresh*)
    leafMassBelowThresh := 0.0; t:= trainEntry;
    WHILE t <> NIL DO
      t^.pAge:= t^.pAge + dPA;
      IF t^.pAge <= leafAgeThresh THEN
        leafMassBelowThresh := leafMassBelowThresh + t^.biomass;
      END(*IF*);
      t:= t^.next;
    END (*WHILE*);
  ELSE
    InsuffMem;
  END(*IF*);
END AddLeafIncreaseToTrain;

PROCEDURE CalcLeafSenence;
  VAR t: LeafTrainPointer;
BEGIN
  dleafSenesc:= 0.0; t:= trainEntry;
  WHILE (t <> NIL) AND (t^.pAge>maxLeafPAge) DO
    dleafSenesc:= dleafSenesc + t^.biomass;
    t:= t^.next; Deallocate(trainEntry); trainEntry:= t;
    (*release of old element*)
  END; (*WHILE*)
END CalcLeafSenence;

PROCEDURE CalcPotNetGrowthRate;
  VAR t: LeafTrainPointer;
BEGIN
  potNetGrowthRate:= 0.0; t:= trainEntry;
  WHILE (t <> NIL) AND (t^.pAge>lpAge1) DO
    potNetGrowthRate:= maxPotNetGR*(a1+b1*t^.pAge)*t^.biomass/leaf
      +potNetGrowthRate;
    t:= t^.next;
  END (*WHILE*);

  WHILE (t <> NIL) AND (t^.pAge>lpAge2) DO
    potNetGrowthRate:= maxPotNetGR*t^.biomass/leaf + potNetGrowthRate;
    t:= t^.next;
  END (*WHILE*);
  WHILE (t <> NIL) DO (*rest of list*)
    potNetGrowthRate:= maxPotNetGR*(a2 +b2*t^.pAge) *t^.biomass /leaf
      +potNetGrowthRate;
    t:= t^.next;
  END; (*WHILE*)
END CalcPotNetGrowthRate;

PROCEDURE UpdateStateVariables;
BEGIN
  newAssim:= Rmax(0.0, assim+dAssim);

  newPA := pA + dPA;
  newLeaf := leaf + dLeaf; (*dSeedLeaf added before for agetrain*)
  newTuber:= tuber+ dTuber;
  newStem := stem + dSeedStem + dStem;
  newRoot := root + dSeedRoot + dRoot;
END UpdateStateVariables;

PROCEDURE CalcDeltaPhysiologicalAge;
  CONST a = 0.6667; b = 0.3333;
  VAR i : CARDINAL;
  temp,dpA: ARRAY [1..4] OF REAL;

```

```

BEGIN
(* compute temperature intermediates (see Sands et al., 1979 *) 
temp[1]:= TMIN;
temp[2]:= a*TMIN + b*TMAX;
temp[3]:= b*TMIN + a*TMAX;
temp[4]:= TMAX;

(* Sands et al.'s algorithm *)
FOR i:= 1 TO 4 DO
  IF (temp[i] < minGrowthTmp) OR (temp[i] >= maxGrowthTmp) THEN
    dpA[i]:= 0.0;
  ELSIF (temp[i] >= minGrowthTmp) AND (temp[i] < optGrowthTmp) THEN
    dpA[i]:= 10.*((1.-(temp[i]-optGrowthTmp)*(temp[i]-optGrowthTmp))
    /minMoptGrT2);
  ELSIF (temp[i] >= optGrowthTmp) AND (temp[i] < maxGrowthTmp) THEN
    dpA[i]:= 10.*((1.-(temp[i]-optGrowthTmp)*(temp[i]-optGrowthTmp))
    /maxMoptGrT2);
  END; (*IF*)
END; (*FOR*)

(* compute change in physiological age *)
dpA:= (5.0*dpA[1] + 8.0*dpA[2] + 8.0*dpA[3] + 3.0*dpA[4])/24.0;
END CalcDeltaPhysiologicalAge;

PROCEDURE CalcFinalDLeaf;
BEGIN
(* subtract killed leaves from new leaf growth to compute net change *)
dleaf:= dLeaf + dSeedLeaf - dLeafSenesc;
END CalcFinalDLeaf;

PROCEDURE UpdateVariety;
BEGIN
  IF (oldVar # variety) THEN
    IF (variety < 1.5) THEN SetPForBintje;
    ELSIF (variety < 2.5) THEN SetPForRussetBurbank;
    ELSIF (variety < 3.5) THEN SetPForNorland;
    ELSIF (variety > 3.5) THEN SetPForMarisPiper;
  END(*IF*);
  oldVar:= variety;
END(*IF*);
END UpdateVariety;

PROCEDURE Initialize;
(*initializes potato model for a simulation run, used in SimMaster*)

PROCEDURE InitSeedContributionBeforeEmergence;
  VAR x: REAL;
BEGIN
  x:= effUtil * seedWeight / tuberDWtoFW * propBeFF;
  dSeedLeaf:= x*xpropInLeaf; dSeedStem:= x*xpropInStem; dSeedRoot:= x*xpropInRoot;
  leaf := dSeedLeaf; stem := dSeedStem; root := dSeedRoot;
END InitSeedContributionBeforeEmergence;

BEGIN (*Initialize*)
UpdateVariety;

minMoptGrT2:= (minGrowthTmp-optGrowthTmp) * (minGrowthTmp-optGrowthTmp);
maxMoptGrT2:= (maxGrowthTmp-optGrowthTmp) * (maxGrowthTmp-optGrowthTmp);

(*compute slopes and intercepts of the "NCGRV vs pAge" relationship
  (see Ng & Loomis (1984), p27): *)
b1:= (0.0-1.0) / (maxLeafPAge-lpAge1);
a1:= 0.0 - b1*maxLeafPAge;
a2:= 0.4;
b2:= (1.0-a2) / lpAge2;

(*initialization of dynamic train list:*)
trainEntry:= NIL; trainEnd:= NIL;

(*initial calculations: *)
InitSeedContributionBeforeEmergence;
AddFirstLeafIncreaseToTrain;
pai:= (leaf+stemArea)*leafDWtoArea/(rowDist*plantDist);
  (*contribution of stem to lai.*)
tuberFW:= tuber*tuberDWtoFW/10.0/plantDist/rowDist;
youngLeafTissue := 1.0; leafMassBelowThresh := leaf;
END Initialize;

PROCEDURE Output;
BEGIN
  SetPAI(pai);

```

```

harvInd := tuber/(leaf+stem+root+tuber);
youngLeafTissue := leafMassBelowThresh/leaf;
tuberFW := tuber*tuberDWTofW/10.0/plantDist/rowDist;
END Output;

PROCEDURE Input; (* Calculates all variables depending on inputs (weather) *)
BEGIN
  CalcPotNetGrowthRate;
  CalcDeltaPhysiologicalAge;
  CalcSeedTuberContribution;
  usedWStress:= 1.0 - (1.0-wStress)*hypWStrF;
END Input;

PROCEDURE Dynamic;
BEGIN
  InitializeTuber;
  CalcLeafSenescence;
  CalcStateVariableIncrease;
  AddLeafIncreaseToTrain;
  CalcFinalLeaf;
  UpdateStateVariables;
END Dynamic;

PROCEDURE Terminate;

PROCEDURE DeallocateAgeTrainList;
  VAR t, next: LeafTrainPointer;
BEGIN
  t:= trainEntry;
  WHILE t<>NIL DO
    next:= t^.next; Deallocate(t); t:= next;
  END; (*WHILE*)
  trainEntry:= NIL; trainEnd:= NIL;
END DeallocateAgeTrainList;

BEGIN (*Terminate*)
  DeallocateAgeTrainList;
END Terminate;

PROCEDURE PotModInfo;
BEGIN
  SetPos( 2,2);
  WriteString('POTATO GROWTH MODEL');
  SetPos( 4,2);
  WriteString('after the publication of:');
  SetPos( 5,2);
  WriteString('Johnson, K.B., Johnson, S.B. & Teng, P.S., 1986.');
  SetPos( 6,2);
  WriteString('Development of a Simple Potato Growth Model for Use');
  SetPos( 7,2);
  WriteString('in Crop-Pest Management', Agricultural Systems 19:189-209.');
  SetPos( 9,2);
  WriteString('and:;');
  SetPos(10,2);
  WriteString('Johnson, K.B., Teng, P.S. & Radcliffe, E.B., 1987.');
  SetPos(11,2);
  WriteString('Coupling Feeding Effects of Potato Leafhopper, Emoasca fabae');
  SetPos(12,2);
  WriteString('(Homoptera: Cicadellidae), Nymphs to a Model of Potato Growth', );
  SetPos(13,2);
  WriteString('Environmental Entomology 16:250-258');
  SetPos(15,2);
  WriteString('implemented for ModelWorks by:');
  SetPos(16,2);
  WriteString('Markus Ulrich, Andreas Fischlin, Olivier Roth & Thomas Nemecek');
  SetPos(17,2);
  WriteString('Systems Ecology, ETH /VOD, CH-8092 Zurich, Switzerland.');
END PotModInfo;

PROCEDURE DeclPotatoModel;
BEGIN
  oldVar:= 1.0;
  DeclIndM(potatoModel,discreteTime,Initialize,Input,Output,Dynamic,Terminate,
  ModelObjects,"Potato growth model (Johnson)","PotatoMod",PotModInfo,mind);
END DeclPotatoModel;

PROCEDURE UndeclPotatoModel;
BEGIN
  RemoveM(potatoModel);

```

```
END UndeclPotatoModel;
END PotatoMod.
```

POTMODVALID [10] D

```
DEFINITION MODULE PotModValid;
(* **** *)
  Purpose: reading and display of plant growth validation data

  Programming
    • Design:          O. Roth, T. Nemecek   4.9.89
    • Implementation: T. Nemecek            6.9.89

    Last revision of definition: 11.04.92 tn
(* **** *)
FROM SimBase IMPORT Model,AuxVar;
FROM DFDData IMPORT DataFrame;

VAR validMod:           Model;
  (* MV potato for validation *)
  leafVDay, stemVDay, rootVDay, tuberVDay, laiVDay,
  leafVDD, stemVDD, rootVDD, tuberVDD, laiVDD,
  soilWatVDay:          AuxVar;

PROCEDURE UsePotValidData(df: DataFrame; ident: ARRAY OF CHAR);
  (* retrieves the potato validation data from the selected data frame *)

PROCEDURE DeclValidModel;
PROCEDURE UndeclValidModel;
END PotModValid.
```

PAVMETHOUR [11] D

```
DEFINITION MODULE PAVMetHour;
(* **** *)
  Purpose
    management of hourly meteorological data for the EPOVIR-model

  Programming
    • Design:          T. Nemecek      20.3.90
    • Implementation: T. Nemecek      20.3.90

    Last revision of definition: 01.05.92 tn
(* **** *)
FROM Vectors2D IMPORT PolVector;
FROM SimBase IMPORT Model,Parameter;

CONST
  firstDay = 121; (* 1st May of normal year *)
  lastDay = 213; (* 31st July of leap year *)

VAR metHourMod:
  Model;
  windThreshold,tempThreshold,precThreshold: (* take-off thresholds *)
  Parameter;

PROCEDURE GetActivityPeriod(): REAL;
  (* returns the fraction of day during which take-off of aphids is possible [d/d] *)
PROCEDURE GetWind(): PolVector;
  (* returns the wind direction of winds during the hours,
  when aphid take-off is possible [rad]*)
PROCEDURE DeclPAVMetHour;
PROCEDURE UndeclPAVMetHour;
END PAVMetHour.
```

PAVSTEADYBEH [12] D

```
DEFINITION MODULE PAVSteadyBeh;
(* **** *)
  Purpose
    estimation of the steady state of the Markov model of vector behaviour

  Programming
    • Design:          T. Nemecek   16.02.91
```

```

    * Implementation: T. Nemecek 16.02.91
    Last revision of definition: 24.01.92 TN
*****
FROM Simbase IMPORT Model;
FROM PAVTypes IMPORT BehArray, BehMatrix;

VAR steadyStateMod: Model;

PROCEDURE CalcSteadyState(cQ, ncQ: BehMatrix; VAR cProp, ncProp: BehArray);
(* calculates the steady state proportions for colonizing 'cProp' and
noncolonizing species 'ncProp', given the transition matrices for a continuous
Markov chain 'cQ' resp. 'ncQ'. The estimate is performed by simulation using
the module Simintegrate. The simulation is stopped if the convergence criterion
is met: |(d Nbc/dt)/Nbc| < 0.0005 for all values, where 'Nbc' is the proportion
of individuals in behaviour state 'b' of colonization group 'c'. This value was
found to be sufficiently small to give a reliable estimate. To avoid an
overflow, the integration step must be sufficiently small. *)
PROCEDURE DeclSteadyBeh;
PROCEDURE UndeclSteadyBeh;
END PAVSteadyBeh.
```

PAVINOCs [13] DI (INOCULATION SUBMODEL OF EPOVIR)

```

DEFINITION MODULE PAVInocs;
*****
Model
  PAVInocs is the inoculation submodel of the EPOVIR-model. It calculates
  the vector intensity, which is used in the infection submodel

Programming
  • Design: T. Nemecek 14.12.90
  • Implementation: T. Nemecek 14.12.90

  Last revision of definition: 06.05.92 TN
*****
FROM SimBase IMPORT Model,Parameter,AuxVar;
FROM PAVTypes IMPORT SpeciesGroup,Colonization,BehArray,BehMatrix;

CONST
  nTreatments = 10;

VAR
  inocsM: Model;

(* parameter *)
(* *****)
  emigrProb,depositRate,concFactor,touchWithin,touchBetween,
  lowLeafArea,movePlProb: Parameter;
  aphicideTime: ARRAY [1..nTreatments] OF Parameter;
  aphicideFact,
  aphicideDur,
  aphicideTreatment, (* effect of aphicide treatment on behaviour *)
  apteraeEff: Parameter; (* "apterae factor" *)
  behChanged: ARRAY [MIN(Colonization)..MAX(Colonization)]
  OF Parameter; (* determines, whether changed
  parameters are applied to a colonization group *)

(* colonization group specific parameter *)
  settlRate,mortality,interLeafMoveRate:
    ARRAY [MIN(Colonization)..MAX(Colonization)] OF Parameter;
  P, Q: ARRAY [MIN(Colonization)..MAX(Colonization)] OF BehMatrix;
  (* Markov matrices P = discrete markov chain, Q = continuous markov chain *)
  nProp: ARRAY [MIN(Colonization)..MAX(Colonization)] OF BehArray;
  (* proportion of the population in a certain behaviour state *)

(* group specific parameter *)
  vectorEff,retFact,serTransm:
    ARRAY [MIN(SpeciesGroup)..MAX(SpeciesGroup)] OF Parameter;

(* parameters for sensitivity analysis *)
  probEff,flightEff,settLRateSA,interLeafMoveRateSA,vectorEffSA,retFactSA,
  serTransmSA,mortalitySA: Parameter;
  freqSA,durSA: BehArray;

(* output *)
  inoculationsF, (*inoculations by flying vectors*)
  inoculationsWL, (*inoculations by vectors walking over leaf bridges*)
  inoculationsWS, (*inoculations by vectors walking over the soil*)
  inoculationsIm, (*inoculations by immigrating vectors vectors*)
```

```

walkProbWithin: (*probability that an interplant movement takes place *)
    AuxVar; (* within the row *)
contribGrp: (*contribution of the different vector groups to the number
            of inoculations *)
    ARRAY [MIN(SpeciesGroup)..MAX(SpeciesGroup)] OF AuxVar;
(* MVs *)
nAphTot: AuxVar;

PROCEDURE CalcNewQ;
(* calculates new transition rates after changes in behaviour parameters *)

PROCEDURE DeclInocs; (* installs the model *)
PROCEDURE UndeclInocs; (* removes the model *)
END PAVInocs.
-----
```

```

IMPLEMENTATION MODULE PAVInocs;
(* DM modules *)
FROM MathProcs IMPORT Rmin,Pi;
FROM DMStrings IMPORT Append,AssignString;
FROM DMWindowIO IMPORT WriteString,WriteLn;
(* MW modules *)
FROM ReadData IMPORT OpenDataFile,CloseDataFile,GetReal;
FROM SimBase IMPORT Model,IntegrationMethod,RCTType,NoTerminate,StashFiling,
    Tabulation,Graphing,RemoveM,CurrentTime,Parameter,StateVar,Derivative,AuxVar;
(* other modules *)
FROM MathLib IMPORT Ln,Exp;
(* Johnson-soil-model modules *)
FROM PotatoMod IMPORT fRadI,rowDist,plantDist,leafDWtoArea,leaf;
(* PAV modules *)
FROM PAVTypes IMPORT Behaviour,Colonization,SpeciesGroup,AphSpec;
FROM PAVBase IMPORT aphSpecAttr,aphGroupAttr,FillArray,noT,noG,noF,DeclIndM,
    AddIndex,DeclIndSV,DeclIndP,DeclIndMV,EPOVIRDetMind,sDem,demo,intl,int2;
FROM PAVValid IMPORT emergD,killD;
FROM PAVMetHour IMPORT GetActivityPeriod;
FROM PAVSuctTrap IMPORT GetAphidNumber;

CONST f = flight; w = walking; r = resting; p = probing; l = longPenetration;
col = colonizing; ncol = noncolonizing;
minC = MIN(Colonization); maxC = MAX(Colonization);
minB = MIN(Behaviour); maxB = MAX(Behaviour);
minG = MIN(SpeciesGroup); maxG = MAX(SpeciesGroup);
minS = MIN(AphSpec); maxS = MAX(AphSpec);
mind = EPOVIRDetMind;

VAR
(* states *)
    nAph: ARRAY [minG..maxG] OF StateVar;
    nAphDot: ARRAY [minG..maxG] OF Derivative;

(* parameter *)
    mortalityFact: (* increase of mortality by aphicide application *)
        Parameter;

(* output *)
    vectPressST,vectPressF,inoculations:
        AuxVar;

(* auxiliary *)
(* calculation of vector intensity: *)
    probFreq, flightFreq, nPropW, (*used in the 'vector intensity' equations*)
    probFreqSD, flightFreqSD, nPropWSD, (* parameters of standard sequences *)
    probFreqMod, flightFreqMod, nPropWMod, (* parameters of modified sequences *)
(* other auxiliaries *)
    emigr, (* emigration rate *) {#/d} *
    nAphCol, (* # of aphids in the field per colonization group *) {#/p} *
    moveRateWL, (* rate of interplant movement over leaf bridges *) {d^-1} *
    moveRateNS, (* rate of interplant movement on soil *) {d^-1} *
    moveRateF, (* rate of interplant movement (within field) by flight *) {d^-1} *
    moveRate, (* plant leaving rate *) {d^-1} *
    aptWalkIncr: (* increase of interplant movement by walking,
                  contributed by apterae *) {#/100} *
    ARRAY [minC..maxC] OF AuxVar;
    suctTrapGr, (* # of aphids caught in the suction trap per group *) {#/d} *
    immigr: (* immigration per group *) {#/d} *
    ARRAY [minG..maxG] OF AuxVar;
    moveProbL, (* conditional probability of interplant movement
                by leaf bridges in case of interleaf movement *) {#/100} *
    moveProbS, (* conditional probability of interplant movement
                on soil in case of interleaf movement *) {#/100} *
    favourCond: (* fraction of day with favourable take-off conditions *) {#/100} *
    AuxVar;
```

```

PROCEDURE About;
BEGIN
  WriteString("PAVINocs");
  WriteLn;
  WriteString(' EPOVIR inoculation model.');
  WriteLn;
  WriteString(' Calculates the number of inoculations per');
  WriteLn;
  WriteString(' source plant per day.');
  WriteLn;
  WriteString(' This model is used as input model by');
  WriteLn;
  WriteString(' PAVDetInfs and PAVStochInfs.');
  WriteLn;
  WriteString(' Uses: PotatoMod, PAVSuctTrap & PAVSteadyBeh.');
  WriteLn;
END About;

{*****}
{* Handling of transition probabilities and rates *}
{*****}

PROCEDURE CalcQ(c: Colonization);
(* calculates the Q-matrix for a continuous markov chain from a
 p-matrix for a discrete markov chain (step = 1 second). *)
CONST secsPerDay = 86400.0;
VAR i,j: Behaviour; sic,Mic: REAL;
BEGIN
  FOR i:= minB TO maxB DO
    Mic := -Ln(P[i,j])*secsPerDay; (* Eq. 5.8 *)
    (* different units of P and Q make multiplication with secsPerDay necessary *)
    Q[i,j] := -Mic;

    sic := 0.0;
    FOR j:= minB TO maxB DO
      IF i#j THEN sic := sic + P[i,j] END (*IF*); (* Eq. 5.7 *)
    END (*FOR*);

    FOR j:= minB TO maxB DO
      IF i#j THEN Q[i,j] := P[i,j]/sic*Mic END (*IF*); (* Eq. 5.6 *)
    END (*FOR*);
  END CalcQ;

PROCEDURE ReadMatrix(c: Colonization);
(* reads the P-matrix for colonization group c from the currently
 open data file *)
VAR i,j: Behaviour;
BEGIN
  (* transition probabilities (P-matrix) *)
  FOR i:= minB TO maxB DO
    FOR j:= minB TO maxB DO
      GetReal("Transition matrix, line ", ORD(i), P[i,j], 0.0, 1.0);
    END (*FOR*);
  END (*FOR*);

  (* precalculated limiting probabilities *)
  FOR i:= minB TO maxB DO
    GetReal("Transition matrix, line ", ORD(i), nProp[c, i], 0.0, 1.0);
  END (*FOR*);
END ReadMatrix;

PROCEDURE ReadMatrices;
VAR fn: ARRAY [0..255] OF CHAR; ok: BOOLEAN;
BEGIN
  fn := 'MPSstation'; OpenDataFile(fn, ok);
  IF ok THEN ReadMatrix(col); (*for M.persicae*) CalcQ(col); CloseDataFile;
  ELSE HALT
  END (*IF*);

  fn := 'AFStation'; OpenDataFile(fn, ok);
  IF ok THEN ReadMatrix(ncol); (*for A.fabae*) CalcQ(ncol); CloseDataFile;
  ELSE HALT
  END (*IF*);
END ReadMatrices;

PROCEDURE ChangeDur(c: Colonization; i: Behaviour; kdi: REAL);
(* changes the mean residence time of behaviour 'i' by factor 'kdi' *)
VAR j: Behaviour;
BEGIN
  FOR j:= minB TO maxB DO
    Q[i,j] := Q[i,j]/kdi; (* Eq. 5.11 *)
  END (*FOR*);
END ChangeDur;

```

```

PROCEDURE ChangeFreq(c: Colonization; j: Behaviour; kfj: REAL);
  (* changes the frequency of behaviour 'j' by factor 'kfj' *)
  VAR i: Behaviour; diffQ: REAL;
BEGIN
  FOR i:= minB TO maxB DO
    IF i # j THEN
      diffQ := Q[c,i,j]-Q[c,i,j]*kfj;
      Q[c,i,j] := Q[c,i,j]*kfj; (*Eq 5.12*)
      Q[c,i,i] := Q[c,i,i]+diffQ; (* adjusting diagonal elements *)
    END(*IF*);
  END(*FOR*);
  IF (j = p) THEN (*probing is treated differently, by using a correction term
                    (Eq 5.13)*)
    diffQ := Q[c,p,l]-Q[c,p,l]/kfj;
    Q[c,p,l] := Q[c,p,l]/kfj;
    Q[c,p,p] := Q[c,p,p]+diffQ; (* adjusting diagonal elements *)
  END(*IF*);
END ChangeFreq;

PROCEDURE CalcNewQ;
  VAR c: Colonization;
BEGIN
  FOR c:= minC TO maxC DO
    CalcQ(c);

    ChangeFreq(c, f, freqSA[f]);
    ChangeFreq(c, w, freqSA[w]);
    ChangeFreq(c, r, freqSA[r]);
    ChangeFreq(c, p, freqSA[p]);
    ChangeFreq(c, l, freqSA[l]);

    ChangeDur (c, f, durSA [f]);
    ChangeDur (c, w, durSA [w]);
    ChangeDur (c, r, durSA [r]);
    ChangeDur (c, p, durSA [p]);
    ChangeDur (c, l, durSA [l]);

  END(*FOR*);
END CalcNewQ;

(*****)
(* 'Dynamic' procedures *)
(*****)

PROCEDURE StandardBehAuxs;
  (* stores the values of the behaviour variables for the standard sequences *)
  VAR c: Colonization;
BEGIN
  FOR c:= minC TO maxC DO
    probFreqSd [c] := nProp[c,f]*Q[c,f,p]+nProp[c,w]*Q[c,w,p]
                  +nProp[c,r]*Q[c,r,p]-nProp[c,p]*Q[c,p,l];
    flightFreqSd[c] := nProp[c,w]*Q[c,w,f]+nProp[c,r]*Q[c,r,f]
                  +nProp[c,p]*Q[c,p,f]+nProp[c,l]*Q[c,l,f];
    nPropWSd [c] := nProp[c,w];
  END(*FOR*);
END StandardBehAuxs;

PROCEDURE CalcMovementRate; (* calculates the rate of interplant movement *)
  VAR c: Colonization;
BEGIN
  FOR c:= minC TO maxC DO
    moveRateF [c]:=flightFreq[c]*favourCond           *(1.0-emigrProb)*flightEff;
    moveRateWL[c]:=nPropW[c] *interLeafMoveRateSA*interLeafMoveRate[c]*moveProbL;
    moveRateWS[c]:=nPropW[c] *interLeafMoveRateSA*interLeafMoveRate[c]*moveProbS;
    moveRate [c]:=flightFreq[c]*favourCond*flightEff+ moveRateWL[c]+moveRateWS[c];
  END(*FOR*);
END CalcMovementRate;

PROCEDURE AphicideTreatment(): BOOLEAN;
  (* chooses the mortality factor and the behaviour sequence parameters according to
     whether an aphicide was applied and to whether the substance is still active *)
  VAR i: INTEGER;

PROCEDURE BehPars(modified: BOOLEAN);
  (* sets the behaviour sequence parameters according to the aphicide treatment *)
  VAR c: Colonization;
BEGIN
  FOR c:= minC TO maxC DO
    IF modified & (behChanged[c]>0.5)
    THEN probfreq [c] := probFreqMod [c];

```

```

flightFreq[c] := flightFreqMod[c];
nPropW [c] := nPropWMod [c];
ELSE probFreq [c] := probfreqSd [c];
flightFreq[c] := flightFreqSd [c];
nPropW [c] := nPropWSd [c];
END(*IF*);
END(*FOR*);
END BehPars;

BEGIN
IF aphicideTreatment > 0.5 THEN
  FOR i:= 1 TO nTreatments DO
    IF (CurrentTime() >= aphicideTime[i])
    AND (CurrentTime() < aphicideTime[i]+aphicideDur)
    THEN (* "aphicide effect")
      mortalityFact := aphicideFact;
      BehPars(TRUE);
      RETURN TRUE
    END(*IF*);
  END(*FOR*);
  (* no aphicide effect *)
  mortalityFact := 1.0;
  BehPars(FALSE);
  RETURN FALSE
ELSE (* no aphicide treatment. Use modified sequences for
      the simulation of other measures *)
  mortalityFact := 1.0;
  BehPars(TRUE);
  RETURN FALSE
END(*IF*);
END AphicideTreatment;

PROCEDURE Apterae;
(*simulates the effect of apterae by increasing the rate of interplant movement by
 walking (Eq 7.3)*)
BEGIN
aptWalkIncr[col] := (CurrentTime()-emergD)/(killD-emergD)*apteraeEff;
(* linear increase assumed *)
aptWalkIncr[ncol] := 0.0;
END Apterae;

PROCEDURE Initialize;
VAR c: Colonization; aphicide: BOOLEAN;
BEGIN
FOR c:= minC TO maxC DO (* calculate variables for modified behaviour sequences *)
  probFreqMod [c] := nProp[c,f]*Q[c,f,p] + nProp[c,w]*Q[c,w,p]
  + nProp[c,r]*Q[c,r,p] - nProp[c,p]*Q[c,p,l];
  flightFreqMod[c] := nProp[c,w]*Q[c,w,f] + nProp[c,r]*Q[c,r,f]
  + nProp[c,p]*Q[c,p,f] + nProp[c,l]*Q[c,l,f];
  nPropWMod [c] := nProp[c,w];
END(*FOR*);

favourCond := GetActivityPeriod();
FillArray(contribGrp, 0.0); FillArray(suctTrapGr, 0.0); FillArray(immigr, 0.0);
moveProbL := 0.0; moveProbs := 0.0;
Apterae; aphicide := AphicideTreatment();
CalcMovementRate;
END Initialize;

PROCEDURE Output;
VAR g: SpeciesGroup; c: Colonization;
  inoccsGr,inocsWLGr,inocsWSGr,inocsImGr,
  vp; (*vector propensity*)
  ppic: AuxVar; (*probability to probe on a plant*)
  inoccsGrp: ARRAY [minG..maxG] OF AuxVar;
PROCEDURE InitOutVars;
BEGIN
  inoculationsF := 0.0; inoculationsWL := 0.0; inoculationsWS := 0.0;
  inoculationsIm := 0.0; vectPressF := 0.0; vectPressST := 0.0;
  FillArray(nAphCol,0.0); FillArray(inocsGrp,0.0); FillArray(contribGrp,0.0);

  END InitOutVars;
BEGIN
  InitOutVars;
  FOR g:= minG TO maxG DO
    c := aphGroupAttr[g];

    (* calculate the number of colonizing resp. noncolonizing aphids *)
    (-----)
    nAphCol[c] := nAphCol[c] + nAph[g];
  END;
END;

```

```

(* calculate the number of inoculations by flight, walking and immigration *)
(*-----*)
vp := Rmin(1.0, vectorEff[g] * vectorEffSA)
  /(1.0 - Rmin(1.0, serTransm[g]*serTransmSA)
   *Exp(-retFact[g]*retFactSA/probFreq[c])); (*Eq 6.16*)
IF (moveRate[c] > 0.0) AND (probEff > 0.5)
THEN ppic := 1.0-Exp(-probFreq[c]/moveRate[c]); (*Eq 6.14*)
ELSE ppic := 1.0;
END(*IF*);

inoccsFGr := nAph [g]*vp*moveRateF [c]*ppic; (*Eq 6.1*)
inoccsWLGr := nAph [g]*vp*moveRateWL[c]*ppic*(1.0+aptWalkIncr[c]); (*Eq 6.2*)
inoccsWSGr := nAph [g]*vp*moveRateWS[c]*ppic*(1.0+aptWalkIncr[c]); (*Eq 6.3*)
inoccsImGr := immigr[g]*vp*ppic; (*Eq 6.4*)

inoculationsF := inoculationsF + inoccsFGr;
inoculationsWL := inoculationsWL + inoccsWLGr;
inoculationsWS := inoculationsWS + inoccsWSGr;
inoculationsIm := inoculationsIm + inoccsImGr;

(* sum up the number of inoculations contributed by group g *)
(*-----*)
inoccsGrp[g] := inoccsFGr + inoccsWLGr + inoccsWSGr + inoccsImGr;

(* vector pressure in the field (F) and in suction trap (ST) *)
(*-----*)
vectPressST := vectPressST+suctTrapGr[g]*Rmin(1.0, vectorEff[g]*vectorEffSA);
vectPressF := vectPressF +nAph [g]*Rmin(1.0, vectorEff[g]*vectorEffSA);
END(*FOR*); (* Species Group loop *)
nAphTot := nAphCol[ncoll] + nAphCol[ncol]; (*total number of aphids*)

(* calculate the contribution of different vector groups to the inoculations *)
(*-----*)
inoculations := inoculationsF + inoculationsWL + inoculationsWS + inoculationsIm;
FOR g:= minG TO maxG DO
  IF inoculations > 0.0
  THEN contribGrp[g] := inoccsGrp[g]/inoculations;
  ELSE contribGrp[g] := 0.0;
  END(*IF*);
END(*FOR*);
END Output;

PROCEDURE Input;
  VAR sp: AphSpec; g: SpeciesGroup; c: Colonization; aphicide: BOOLEAN;
    suctTrapCatches: ARRAY [minS..maxS] OF REAL;
    (* # of aphids caught in suction trap (#/d) *)
BEGIN
  favourCond := GetActivityPeriod();
  FillArray(suctTrapGr, 0.0);
  IF CurrentTime() < killD THEN (*get input from PAVSuctTrap:
    aphid abundance in suction trap*)
    GetAphidNumber(suctTrapCatches);
    FOR sp:= minS TO maxS DO (* calculate # of aphids per vector group*)
      g := aphSpecAttr[sp].grp;
      suctTrapGr[g] := suctTrapGr[g] + suctTrapCatches[sp];
    END(*FOR*);
    FOR g:= minG TO maxG DO (*Eq 6.8, immigration rate per vector group*)
      immigr[g] := depositRate * concFactor * suctTrapGr[g] * (plantDist*rowDist);
    END(*FOR*);
  ELSE
    FillArray(immigr, 0.0); (*no immigration after haulm-killing*)
  END(*IF*);

  (*calculate the interplant movement prob. within/between rows in function of the
  leaf area (Eq. 6.13) *)
  IF fRadI < touchWithin THEN (*plants do not touch*)
    walkProbWithin := 0.0; moveProbL := 0.0;
  ELSIF fRadI < touchBetween THEN (*plants touch within row*)
    walkProbWithin := 1.0; moveProbL := 2.0*movePlProb;
  ELSE
    (*plants touch within & between rows*)
    walkProbWithin := 0.5; moveProbL := 4.0*movePlProb;
  END(*IF*);
  moveProbS := Rmin(1.0, lowLeafArea/(leafDWtoArea*leaf));
  (*interplant movement prob. on bare soil (Eq 6.12) *)

  aphicide := AphicideTreatment();
  CalcMovementRate;

  FOR c:= minC TO maxC DO (*emigration rates, Eq 6.6*)
    emigr[c] := flightFreq[c]*flightEff*emigrProb*favourCond;
  END(*FOR*);

  Apterae;
END Input;

```

```

PROCEDURE Dynamic;
  VAR g: SpeciesGroup; c: Colonization;
BEGIN
  FOR g:= minG TO maxG DO
    c := aphGroupAttr[g];
    (*vector abundance (Eq 6.5): *)
    nAphDot[g] := -(mortality[c]*mortalitySA*mortalityFact
      + emigr[c]
      + settlRateSA*settlRate[c]) * nAph[g]
      + immigr[g];
  END (*FOR*);
END Dynamic;

{*****}
(* Declarations *)
{*****}

PROCEDURE InitPars;
  (*initialization of vector group specific parameters*)
BEGIN
  vectorEff[grp1] := 0.5086; retFact[grp1] := 10.57; serTransm[grp1] := 0.5517;
  vectorEff[grp2a]:= 0.1006; retFact[grp2a]:= 26.21; serTransm[grp2a]:= 0.2803;
  vectorEff[grp2b]:= 0.1006; retFact[grp2b]:= 26.21; serTransm[grp2b]:= 0.2803;
  vectorEff[grp3] := 0.01701;retFact[grp3] := 4.5106;serTransm[grp3] := 0.2803;
END InitPars;

PROCEDURE DeclModelObjects;
  CONST numbAph = 30.0; (*for scaling of MVs*)
  upLimSA = 10.0; (*for scaling of parameters in sensitivity analysis*)
  VAR g: SpeciesGroup; i: INTEGER;
  newDescr, newIdent: ARRAY [0..63] OF CHAR;
  PROCEDURE CreateLabels(grp: SpeciesGroup; descr, ident: ARRAY OF CHAR;
    VAR newDescr, newIdent: ARRAY OF CHAR);
    (*prepares idents for vector group specific objects*)
    VAR label1, label2: ARRAY [0..15] OF CHAR;
  BEGIN
    AssignString(descr, newDescr); AssignString(ident, newIdent);
    CASE grp OF
      grp1 : label1 := 'of group 1'; label2 := 'g1';
      | grp2a : label1 := 'of group 2a'; label2 := 'g2a';
      | grp2b : label1 := 'of group 2b'; label2 := 'g2b';
      | grp3 : label1 := 'of group 3'; label2 := 'g3';
    END(*CASE*);
    Append(newDescr, label1); Append(newIdent, label2);
  END CreateLabels;
BEGIN
  InitPars;
  FOR g:= minG TO maxG DO (*vector group specific objects*)
    CreateLabels(g, 'aphids', 'nAph', newDescr, newIdent);
    DeclIndSV(nAph[g],nAphDot[g],0.0,0.0,1000.0,newDescr,newIdent,'#/pl',mInd);
    DeclIndMV(nAph[g], 0.0,numbAph,newDescr,newIdent,'#/pl',
      noF,noT,noG,mInd,int2);
    DeclIndMV(nAphDot[g], 0.0,numbAph,newDescr,newIdent,'#/pl*d'),
      noF,noT,noG,mInd,int2);
    CreateLabels('immigration', 'immigr', newDescr, newIdent);
    DeclIndMV(immigr[g],0.0,numbAph,newDescr,newIdent,'#/pl*d'),
      noF,noT,noG,mInd,int2);

    CreateLabels(g, 'vector efficiency', 'vectEff', newDescr, newIdent);
    DeclIndP(vectorEff[g],vectorEff[g],0.0,1.0,rtc,newDescr,newIdent,'%/100',
      mInd,sDem);

    CreateLabels(g, 'retention factor', 'retFact', newDescr, newIdent);
    DeclIndP(retFact[g],retFact[g],0.0,300.0,rtc,newDescr,newIdent,'d^-1',
      mInd,sDem);

    CreateLabels(g, 'serial transmission rate', 'serTransm', newDescr, newIdent);
    DeclIndP(serTransm[g],serTransm[g],0.0,1.0,rtc,newDescr,newIdent,'%/100',
      mInd,sDem);
  END(*FOR*);

  (*vector abundance in the field*)
  DeclIndMV(nAphCol[col], 0.0,numbAph,'col. aphids', 'nAphCol[col]', '#/pl',
    noF,noT,noG,mInd,demo);
  DeclIndMV(nAphCol[nCol],0.0,numbAph,'noncol. aphids','nAphCol[ncol]', '#/pl',
    noF,noT,noG,mInd,demo);
  DeclIndMV(nAphTot, 0.0,numbAph,'total aphids', 'nAphTot', '#/pl',
    noF,noT,noG,mInd,sDem);

```

```

DeclIndMV(inoculationsF, 0.0,100.0,'inoc. by flight',
  'inocsF', '#/(pl*d)',noF,noT,noG,mInd,int2);
DeclIndMV(inoculationsWL,0.0,100.0,'inoc. by walking by leaf bridges',
  'inocsWL', '#/(pl*d)',noF,noT,noG,mInd,int2);
DeclIndMV(inoculationsWS,0.0,100.0,'inoc. by walking on soil',
  'inocsWS', '#/(pl*d)',noF,noT,noG,mInd,int2);
DeclIndMV(inoculationsIm,0.0,100.0,'inoc. by immigrating aphids',
  'inocsIm', '#/(pl*d)',noF,noT,noG,mInd,int2);
DeclIndMV(inoculations, 0.0,100.0,'total inoculations',
  'inocs', '#/(pl*d)',noF,noT,noG,mInd,demo);

DeclIndMV(vecPressT,0.0,10.0,'vector pressure suct. trap','vectPressST','#',
  'noF,noT,noG,mInd,demo');
DeclIndMV(vecPressf, 0.0,10.0,'vector pressure field',      'vectPressF', '#/pl',
  'noF,noT,noG,mInd,demo');

DeclIndP(mortality[col], 0.09346,0.0,100.0,rtc,'mortality[col]',
  'mortality[col]', 'd~1',mInd,sDem);
DeclIndP(mortality[ncol],0.25, 0.0,100.0,rtc,'mortality[ncol]',
  'mortality[ncol]', 'd~1',mInd,sDem);

DeclIndP(sett1Rate[col], 0.2629,0.0,1.0,rtc,'settling rate[col]',
  'sett1Rate[col]', 'd~1',mInd,sDem);
DeclIndP(sett1Rate[ncol],0.0, 0.0,1.0,rtc,'settling rate[ncol]',
  'sett1Rate[ncol]', 'd~1',mInd,sDem);

DeclIndP(emigrProb,0.05,0.0,1.0,rtc,'emigration probability',
  'emigrProb', '%/100',mInd,sDem);

DeclIndP(interLeafMoveRate[col], 67.72,0.0,1000.0,rtc,'interLeafMoveRate[col]',
  'interLeafMoveRate[col]', 'd~1',mInd,sDem);
DeclIndP(interLeafMoveRate[ncol],33.05,0.0,1000.0,rtc,'interLeafMoveRate[ncol]',
  'interLeafMoveRate[ncol]', 'd~1',mInd,sDem);

DeclIndP(touchWithin, 0.367, 0.0,1.0,rtc, 'threshold for touching within row',
  'touchWithin', '%/100',mInd,sDem);
DeclIndP(touchBetween,Pi/4,0.0,0.1,0,rtc, 'threshold for touching between rows',
  'touchBetween', '%/100',mInd,sDem);
DeclIndP(lowLeafarea, 0.069, 0.0,1.0,rtc,           'area of the lowest leaves',
  'lowLeafarea', 'm^2', mInd,sDem);
DeclIndP(movePlProb, 1.0/18.0, 0.0,1.0,noRtc, 'interplant movement probability',
  'movePlProb', '%/100',mInd,sDem);

DeclIndP(depositRate,0.0237,0.0,100.0,rtc,'deposition rate','depositRate',
  'm~2',mInd,sDem); (* after Taylor & Palmer,1972 *)
DeclIndP(concFactor,40.0,0.0,1000.0, rtc,'concentration factor','concFact',
  '%/100',mInd,sDem); (* after Carter,et al. 1982 *)

(* parameters for structural and sensitivity analysis *)
DeclIndP(flightEff, 1.0,0.0,1.0,rtc,'effect of flight 0-no,1-normal',
  'flightEff', '%/100',mInd,int1);
(*flightEff=0 suppresses inoculation by flying aphids*)
DeclIndP(probEff, 1.0,0.0,1.0,rtc,'effect of probing 0-no,1-normal',
  'probEff', '%/100',mInd,int1);
(*probEff=0 sets the probability to probe on a plant to 1.0*)
DeclIndP(apteraeEff, 0.0,0.0,300.0,rtc,"effect of apterae",
  'apteraeEff', '%/100',mInd,int1);

DeclIndP(sett1RateSA, 1.0,0.0,upLimSA,noRtc,'sett1RateSA',
  'sett1RateSA', '%/100',mInd,int1);
DeclIndP(interLeafMoveRateSA,1.0,0.0,upLimSA,noRtc,'leafMoveRatesA',
  'leafMoveRateSA', '%/100',mInd,int1);
DeclIndP(vectorEffSA, 1.0,0.0,upLimSA,noRtc,'vectorEffSA',
  'vectorEffSA', '%/100',mInd,int1);
DeclIndP(retFactSA, 1.0,0.0,upLimSA,noRtc,'retFactSA',
  'retFactSA', '%/100',mInd,int1);
DeclIndP(serTransmSA, 1.0,0.0,upLimSA,noRtc,'serTransmSA',
  'serTransmSA', '%/100',mInd,int1);
DeclIndP(mortalitySA, 1.0,0.0,upLimSA,noRtc,'mortalitySA',
  'mortalitySA', '%/100',mInd,int1);

(*parameters that change the frequencies resp. durations of behaviour states*)
DeclIndP(freqSA[f],1.0,0.0,0.10000.0,noRtc,'freqSA[flight]',
  'freqSA[f]', '%/100',mInd,int1);
DeclIndP(freqSA[w],1.0,0.0,0.10000.0,noRtc,'freqSA(walking)',
  'freqSA[w]', '%/100',mInd,int1);
DeclIndP(freqSA[r],1.0,0.0,0.10000.0,noRtc,'freqSA(resting)',
  'freqSA[r]', '%/100',mInd,int1);
DeclIndP(freqSA[p],1.0,0.0,0.10000.0,noRtc,'freqSA(probing)',
  'freqSA[p]', '%/100',mInd,int1);
DeclIndP(freqSA[l],1.0,0.0,0.10000.0,noRtc,'freqSA(longPenetration)',
  'freqSA[l]', '%/100',mInd,int1);

```

```

DeclIndP(durSA[f],1.0,0.0,10000.0,noRtc,'durSA[flight]',           'durSA[f]',
'#/100',mInd,int1);
DeclIndP(durSA[w],1.0,0.0,10000.0,noRtc,'durSA[walking]',          'durSA[w]',
'#/100',mInd,int1);
DeclIndP(durSA[r],1.0,0.0,10000.0,noRtc,'durSA[resting]',          'durSA[r]',
'#/100',mInd,int1);
DeclIndP(durSA[p],1.0,0.0,10000.0,noRtc,'durSA[probing]',          'durSA[p]',
'#/100',mInd,int1);
DeclIndP(durSA[l],1.0,0.0,10000.0,noRtc,'durSA[longPenetration]', 'durSA[l]',
'#/100',mInd,int1);

FOR i:= 1 TO nTreatments DO (*time of treatments*)
  AddIndex(newIdent,'aphicideTime',i,TRUE);
  DeclIndP(aphicideTime[i],120.0+FLOAT(i)*10.0,1.0,366.0,rtc,newIdent,newIdent,
  'jul day',mInd,sDem);
END(*FOR*);
DeclIndP(aphicideFact,1.0,0.0,100.0,rtc,'aphicide induced increase of mortality',
'aphicideFact','-',mInd,sDem);
DeclIndP(aphicideDur, 1.0,0.0,20.0,rtc,'duration of aphicide action',
'aphicideDur', 'd',mInd,sDem);
DeclIndP(aphicideTreatment,0.0,0.0, 1.0,rtc,'0=no/1=aphicide treatment',
'aphicideTreatment','-',mInd,sDem);

DeclIndP(behChanged[col],1.0,0.0,1.0,rtc,'0=unchanged,1=changed',
'behChanged[col]','-',mInd,int1);
DeclIndP(behChanged[ncol],1.0,0.0,1.0,rtc,'0=unchanged,1=changed',
'behChanged[ncol]','-',mInd,int1);
END DeclModelObjects;

PROCEDURE DeclInocs;
BEGIN
  DeclIndM(inocsM, RungeKutta4, Initialize, Input, Output, Dynamic, NoTerminate,
  DeclModelObjects, 'EPOVIR Inoculation submodel', 'PAVINocs', About,mInd);
  ReadMatrices;
  StandardBehAuxs;
END DeclInocs;

PROCEDURE UndeclInocs;
BEGIN
  RemoveM(inocsM);
END UndeclInocs;

END PAVInocs.

```

PAVSTOCHINFS [14] DI (INFECTION SUBMODEL OF EPOVIR)

```

DEFINITION MODULE PAVStochInfs;
(*
  Purpose
    Stochastic spatial infection submodel of EPOVIR
  Programming
    * Design:      T. Nemecek   13.02.89
    * Implementation: T. Nemecek   13.02.89
  Last revision of definition: 11.05.92 tn *)
FROM PAVTypes IMPORT healthy,removed;
FROM Matrices IMPORT Matrix;
FROM CAModBase IMPORT CAMod;
FROM SimBase IMPORT Model,Parameter,AuxVar;
FROM StochStat IMPORT StatArray;

CONST
  nRogTimes = 5;      (* max. # of roguing times per season *)

TYPE
  OutProc = PROCEDURE;
  TermProc = PROCEDURE;

VAR
  stochInfsM:           Model;
  infState:             CAMod;  (*states, discrete*)
  infTubers,infPlants: CAMod;  (*outputs, continuous*)

(* parameter *)
  nRows, nPlants:      INTEGER;
  nRowsR, nPlantsR,    (* margin around the validated field plot      [m] *)
  margin,               (* initial disease incidence                      [%/100] *)
  initInf,              (*

```

```

basicInf,          (* initial disease incidence around the
                   experimental plot                                [%/100 *]
latPerMean, latPerSD, (* latent period                           [DD] *)
infTubDel, infTubSD, (* tuber infection delay                      [DD] *)
propInfLeaves,    (* conditional prob. to become source                  [%/100 *]

(* dispersal parameter *)
alpha, beta,       (* parameters of the Weibull distribution of the flight
                   distances *)                                         [m] *)
kappa,             (* parameter of von Mises distribution of the flight
                   direction *)                                         [m] *)
walkDist,          (* mean walking distance                                     [m] *)

(* switch pseudo-parameter *)
torus,             (* decides on simulation of torus (-1)
                   or finite field (=0) *)                               [m] *)
randomize,         (* this option determines the initialization of the seeds
                   for the random number generator (rng) and the initial
                   distribution of secondary sources:
                   0 - reset the initial distribution of the sec.
                   sources used in the last run, seed of rng are
                   unchanged
                   1 - determine the initial distribution by stochastic
                   sampling (default)
                   2 - use the predefined initial distribution (from
                   data frame) *)                                     [m] *)
distrType,          (* decides, whether the inoculations are randomly or
                   nonrandomly distributed around the infection sources *) [m] *)
rogEffs,            (* roguing efficiency for secondary infection sources *) [m] *)
rogEffP,            (* roguing efficiency for primary infection sources *) [m] *)
Parameter;
rogTime;           (* times of roguing                                         [julDay] *)
ARRAY [1..nRogTimes] OF Parameter;

(* MVs *)
cumPP1:             ARRAY [healthy..removed] OF AuxVar;
cumPSIDisp,cumPPIDisp,cumPlatDisp,pInffTub,pInftTubDisp,pInffPl,pInftPlDisp,ageSens:
AuxVar;

(* StatArrays *)
stArCumSecInf,stArCumPrimInf,stArCumLatent,stArInfTubers,stArInfPlants:
StatArray;

PROCEDURE InfPlantProb(infTubs: REAL): REAL;
(* returns the binomial probability that at least one infected tuber is sampled
   from a plant having a fraction 'inftub' of the tubers infected. The procedure
   uses the current value of 'nTubSample'. *)

PROCEDURE SetEvalPlantCond;
(* statistics is calculated only for a subset of the cells for which validation
   data are available. This option is used in validation experiments. *)
PROCEDURE SetAllPlantCond;
(* statistics is calculated for all cells. This option is used in all other
   simulation experiments. *)

PROCEDURE SetInitialFieldState(fieldState: Matrix);
(* sets the initial DiseaseState of the plants in the field. Following codes
   are used:
   missingVal = plants not sampled (belongs to 'margin')
   0           = healthy plant
   3           = secondary infection source
   4           = plant lacking *)
(* The following procedures install and calculate statistics for different types
   of experiments:
   Norm      single stochastic experiment: the output values are average at
             each day for n runs
   StructExp structured experiment: the values are averaged at 3 times during
             a simulation run,
             using the module StochStat
   SensAna   sensitivity analysis: same as 'StructExp', but the statistics are
             calculated by the module PAVSensAna and not by StochStat *)
PROCEDURE DeclSAsNorm;
PROCEDURE DeclSAsStructExp;
PROCEDURE DeclSAsSensAna;

PROCEDURE PutValuesNorm;
PROCEDURE PutValuesStructExp;
PROCEDURE PutValuesSensAna;

PROCEDURE UndeclStochInfsStatArrays; (* removes all stat arrays of the module *)
PROCEDURE SetOutProc( op: OutProc); (* allows to install a procedure which is
   called at the end of the Output-procedure of the model *)

```

```

PROCEDURE GetOutProc(VAR op: OutProc);
PROCEDURE SetTermProc( tp: TermProc); (* allows to install a procedure which is
    called at the end of the Terminate-procedure of the model *)
PROCEDURE GetTermProc(VAR tp: TermProc);

(* output procedure *)
PROCEDURE GetInflTubMeans(VAR infTub: ARRAY OF REAL);
(* returns an array of the fraction of tubers infected for 3 times calculated
    during a 'StructExp'-experiment *)
PROCEDURE DeclStochInfs;
PROCEDURE UndeclStochInfs;
END PAVStochInfs.

-----
IMPLEMENTATION MODULE PAVStochInfs;
(* other modules *)
FROM Vectors2D      IMPORT Vector,IntVector,PolyVector,TransPolToCartVector,
    AddVectors;
FROM StochStat      IMPORT StatArray,DeclStatArray,RemoveStatArray,
    GetSingleStatistics,PutValue,DeclDispMV,noExistingStatArray,Prob2Tail;
FROM RandGens        IMPORT RandExp,RandWeibull,SetVMPars,VM,UGeneral;
FROM Matrices        IMPORT Matrix,GetMatrixEle,GetMatrixDim,noExistingMatrix;
FROM MyMathLib        IMPORT Sign;
FROM CAModBase       IMPORT CAMod,CAarC,CAarD,UpdateMode,StateType,nonexistentCAMod,
    DeclCAMod,GetStatePtrsC,GetStatePtrsD,GetCASpecs,RemoveCAMod,UpdateCAState,
    CA Frequencies,CA Mean,DoForAllCells,FillCAModD,CAExists,SetCellID,SetStatsCond;
(* MW modules *)
FROM RandGen         IMPORT U,SetSeeds,GetSeeds;
FROM RandNormal       IMPORT InstallU,SetPars,N;
FROM SimGraphUtils    IMPORT timelapseIndep;
FROM SimBase          IMPORT IntegrationMethod,StashFiling,Tabulation,Graphing,
    RTCType,GetGlobSimParams,CurrentStep,CurrentTime,LineStyle,Stain,RemoveM,SetP,
    DoNothing,Parameter,AuxVar;
(* DM modules *)
FROM DMStrings        IMPORT Append;
FROM MathProcs        IMPORT Round,Rmax,Pi,PowerI;
FROM DMWindowIO       IMPORT WriteString,WriteLn;
(* Johnson-soil-model modules *)
FROM PotatoMod        IMPORT rowDist,plantDist,pA,youngLeafTissue;
(* PAV modules *)
FROM PAVTypes          IMPORT healthy,latente,primInf,secInf,removed,SpeciesGroup;
FROM PAVBase           IMPORT diseaseStateLabel,noI,noG,noF,DeclIndP,DeclIndMV,
    SetDeflAndCurP,EPOVIRSpatMind,DeclIndM,calcEvenness,demo,int2,sDem,UserLevel,
    SetDeflCurveAttrForExMV,nSensTub,AddIndex;
FROM PAVAux            IMPORT ModAngle,CompassToAngle,PosToPlant,PlantToPos,
    InfectionSource,RandRealToInt,Torus,FillArrayR,Logit;
FROM PAVValid          IMPORT emergD,killD;
FROM PAVSensAna        IMPORT AddsAOutVar,CollectOutValue;
FROM PAVInocs          IMPORT inoculationsF,inoculationsWL,inoculationsWS,
    inoculationsIm,walkProbWithin,contribGrp;
FROM PAVSpatDistr      IMPORT InitSpatDistVectors,AddCurPatternToData,DisplaySimData,
    InstallSpatPats,RemoveSpatPats;
FROM PAVMetHour        IMPORT GetWind;

CONST
  mind = EPOVIRSpatMind;
  true = 1; false = 0; (* constants for boolean CAs *)

VAR
  (* states *)
  infStateK, infStateK1:          (* disease states *)
    CAarD;

  (* input *)
  plantInoculated,plantRogued:   (*discrete: boolean CA*)
    CAmod;
  plantInoculatedK,plantRoguedK:
    CAarD;

  (* output *)
  infTubersK,                  (* fraction of tubers infected at k *)
  infPlantsK:                   (* probability that a sample contains at least one *)
    CAarC;                      (* infected tuber *)

  (* parameter *)
  immigrTranSm,                (* fraction of immigrating aphids coming from sources *)
  orientation,                 (* orientation of the field [0..2pi] *)
  nTubSample,                  (* # of tubers sampled per plant *)
  (* initial seeds of random number generator: *)
  seedXInitR,seedYInitR,seedZInitR:
    Parameter;

```

```

(* StatArrays *)
stArSensTub:           (* to calculate the mean prop. of inf. tubers in
                        structured experiments *)
    ARRAY [1..nSensTub] OF StatArray;
sensTubInd:             (* indices for statistics *)
    ARRAY [1..nSensTub] OF INTEGER;

(* auxiliary & MVs *)
numbOfPlants,            (* # of cells in simulated field *)
windDir,                 (* mean wind direction during periods favourable for
                        flight *)
effInfections, logitDiseased, inf:
    AuxVar;
propPlants, propEvalPlants:
    ARRAY [healthy..removed] OF AuxVar;
sensitivity,             (* age-dependent sensitivity factors *)
tuberInfAge,             (* time at which the tubers of a plant become infected*)
outbreakAge:              (* time at which the plant becomes source *)
    CAMod; (*continuous*)
initState,                (* initial disease states of the field *)
evalPlant:                (* determines, whether the plant is included in
                        statistical analysis *)
    CAMod; (*discrete*)

sensitivityC, tuberInfAgeC, outbreakAgeC: CAArC;
initStateD, evalPlantD: CAArD;
ind: INTEGER;             (* index for different monitoring times *)
fieldSize: IntVector;     (* field size in # of rows and columns *)
fieldSizeM: Vector;       (* field size in m *)

(* outputs and MVs for the estimation of the the contribution of the vector groups
   to the infections *)
infectionsGrp,
propInfGrp: ARRAY [MIN(SpeciesGroup)..MAX(SpeciesGroup)] OF AuxVar;
propInfCol: AuxVar;

initStateMatrix:          (* stores the initial state of the field *)
    Matrix;
curOP: OutProc;
curTP: TermProc;
isTorus: BOOLEAN;

(*****)
(* Auxiliary procedures *)
(*****)

PROCEDURE SetState(x,y,ds: INTEGER; infTub, infPl, sensitivity: REAL);
(* Sets the state of cell (x,y) to disease state 'ds' and all other
   attributes to the corresponding values *)
BEGIN
    infStateK^ [x]^ [y] := ds;
    initStateD^ [x]^ [y] := ds;
    infTubersK^ [x]^ [y] := infTub;
    infPlantsK^ [x]^ [y] := infPl;
    sensitivityC^ [x]^ [y] := sensitivity;
END SetState;

(* The following 3 procedures set the states of a cell to secInf, healthy resp.
   removed *)
PROCEDURE SetSecInf (ca:CAMod;x,y:INTEGER);
BEGIN SetState(x,y,secInf, 1.0,1.0,0.0) END SetSecInf;

PROCEDURE SetHealthy(ca:CAMod;x,y:INTEGER);
BEGIN SetState(x,y,healthy,0.0,0.0,1.0) END SetHealthy;

PROCEDURE SetRemoved(ca:CAMod;x,y:INTEGER);
BEGIN SetState(x,y,removed,0.0,0.0,0.0) END SetRemoved;

PROCEDURE SetInitState(ca: CAMod; x,y: INTEGER);
(* assigns the initial state to a cell *)
BEGIN
    CASE initStateD^ [x]^ [y] OF
        healthy : SetHealthy(ca,x,y);
        | secInf : SetSecInf (ca,x,y);
        | removed : SetRemoved(ca,x,y);
    END(*CASE*);
END SetInitState;

PROCEDURE InitAuxiliaries;
BEGIN
    fieldSizeM.x := nRowsR *rowDist; fieldSizeM.y := nPlantsR*plantDist;

```

```

numbOfPlants := nRowsR*nPlantsR;
nRows      := Round(nRowsR);           nPlants      := Round(nPlantsR);
fieldSize.x := nRows;                 fieldSize.y := nPlants;

SetVMPars(0.0, kappa);

isTorus      := torus > 0.5;
END InitAuxiliaries;

(* ****)
(* CA-management *)
(* ****)

(* condition procedures for statistical analysis of the field state *)
PROCEDURE EvalPlantCond(ca: CAMod; x, y: INTEGER): BOOLEAN;
(* only plants, for which validation data are available are used for statistical
analysis *)
BEGIN
  RETURN evalPlantD^(x)^y = true
END EvalPlantCond;

PROCEDURE EvalNotRemPlantCond(ca: CAMod; x, y: INTEGER): BOOLEAN;
(* statistics are calculated for all except the removed plants *)
BEGIN
  RETURN (evalPlantD^(x)^y = true) AND (infStateK^ (x)^y # removed)
END EvalNotRemPlantCond;

PROCEDURE AllPlantCond(ca: CAMod; x, y: INTEGER): BOOLEAN;
(* all plants included in statistical analysis *)
BEGIN
  RETURN TRUE
END AllPlantCond;

PROCEDURE AllNotRemPlantCond(ca: CAMod; x, y: INTEGER): BOOLEAN;
(* all but removed plants included in statistical analysis *)
BEGIN
  RETURN (infStateK^ (x)^y # removed)
END AllNotRemPlantCond;

PROCEDURE SetEvalPlantCond;
(* only plants, for which validation data are available are used for statistical
analysis *)
BEGIN
  SetStatsCond(infState, EvalPlantCond);
  SetStatsCond(infTubers, EvalNotRemPlantCond);
  SetStatsCond(infPlants, EvalNotRemPlantCond);
END SetEvalPlantCond;

PROCEDURE SetAllPlantCond;
(* all plants are used included in statistical analysis *)
BEGIN
  SetStatsCond(infState, AllPlantCond);
  SetStatsCond(infTubers, AllNotRemPlantCond);
  SetStatsCond(infPlants, AllNotRemPlantCond);
END SetAllPlantCond;

PROCEDURE DeclCAMods(x, y: INTEGER);
(* declare the CAs for the size {x,y} *)
BEGIN
  DeclCAMod (infState,          discrete, x,y,healthy,removed,withNewState);
  DeclCAMod (plantInoculated,   discrete, x,y,false,  true,  noUpdate);
  DeclCAMod (plantRogued,       discrete, x,y,false,  true,  noUpdate);
  DeclCAMod (infTubers,         continuous,x,y,0,    1,    noUpdate);
  DeclCAMod (infPlants,         continuous,x,y,0,    1,    noUpdate);
  DeclCAMod (sensitivity,      continuous,x,y,0,    1,    noUpdate);
  DeclCAMod (tuberInfAge,      continuous,x,y,0,  1000,  noUpdate);
  DeclCAMod (outbreakAge,      continuous,x,y,0,  1000,  noUpdate);
  DeclCAMod (initState,         discrete, x,y,healthy,removed,noUpdate);
  DeclCAMod (evalPlant,         discrete, x,y,false,  true,  noUpdate);
END DeclCAMods;

PROCEDURE RemoveCAMods;
BEGIN
  RemoveCAMod(infState);      RemoveCAMod(plantInoculated); RemoveCAMod(plantRogued);
  RemoveCAMod(infTubers);     RemoveCAMod(infPlants);        RemoveCAMod(sensitivity);
  RemoveCAMod(tuberInfAge);   RemoveCAMod(outbreakAge);     RemoveCAMod(initState);
  RemoveCAMod(evalPlant);
END RemoveCAMods;

```

```

PROCEDURE GetStatePtrs;
  (* retrieves the pointers to the data arrays of the CAMods. Using these pointers
   to address cells directly is more efficient, but bears the risk to destroy data,
   if incorrect indices are used. Therefore this way of access is used only for
   reading/Writing is performed with the procedures provided by the module CAModBase
   by checking the index range *)
  VAR dummyD: CAarD; dummyC: CAarC; (* unused pointers *)
BEGIN
  GetStatePtrsD(infState,           infStateK,           infStateK1);
  GetStatePtrsD(plantInoculated,    plantInoculatedK,    dummyD);
  GetStatePtrsD(plantRogued,       plantRoguedK,       dummyD);
  GetStatePtrsC(inTubers,          infTubersK,          dummyC);
  GetStatePtrsC(inPlants,          infPlantsK,          dummyC);
  GetStatePtrsC(sensitivity,      sensitivityC,        dummyC);
  GetStatePtrsC(tuberInfAge,      tuberInfAgeC,        dummyC);
  GetStatePtrsC(outbreakAge,      outbreakAgeC,        dummyC);
  GetStatePtrsD(initState,         initStateD,          dummyD);
  GetStatePtrsD(evalPlant,         evalPlantD,          dummyD);
END GetStatePtrs;

PROCEDURE InstallCAModel(x, y: INTEGER); (* installs the CA-model *)
  VAR InfAgeK1, infChangeK1: CAarD;
BEGIN
  RemoveCAMods;
  DecICAMods(x, y);
  GetStatePtrs;
  RemoveSpatPats; (* the spatial pattern analysis depends on the CA *)
  InstallSpatPats;
END InstallCAModel;

PROCEDURE InitCAModel(x, y: INTEGER);
  (* declares resp. redefines the model (if size changed) x,y = size of CA-model *)
  VAR stateType: Statetype; mode: UpdateMode;
  nRows, nCols, minState, maxState: INTEGER;
BEGIN
  IF NOT CAExists(infState)
  THEN InstallCAModel(x, y)
  ELSE GetCASpecs (infState, stateType, nRows, nCols, minState, maxState, mode);
  IF (x # nRows) OR (y # nCols)
  THEN InstallCAModel(x, y) (* dimensions have been changed *)
  END(*IF*);
  END(*IF*);
END InitCAModel;

{*****
(* Exported procedures *) (* comments see PAVStochInfs.DEF *)
*****}

PROCEDURE InfPlantProb(infTubs: REAL): REAL;
BEGIN
  RETURN 1.0 - PowerI(1.0 - infTubs, Round(nTubSample));
END InfPlantProb;

PROCEDURE SetOutProc( op: OutProc); BEGIN curOP := op END SetOutProc;
PROCEDURE GetOutProc(VAR op: OutProc); BEGIN op := curOP END GetOutProc;

PROCEDURE SetTermProc( tp: TermProc); BEGIN curTP := tp END SetTermProc;
PROCEDURE GetTermProc(VAR tp: TermProc); BEGIN tp := curTP END GetTermProc;

PROCEDURE SetInitialFieldState(fieldState: Matrix);
  (* assigns a predefined intial distribution of secodary sources to the model *)
  VAR x,y,marginX,marginY,nC,nR: INTEGER;
  stateR: REAL;
BEGIN
  (* update 'initStateMatrix' *)
  initStateMatrix := fieldState;

  (* number of plants and rows in margin *)
  marginX := Round(margin/rowDist); marginY := Round(margin/plantDist);

  (* total number of rows and plants per row *)
  GetMatrixDim(fieldState, nR, nC);
  nRows := nC + 2*marginX; nPlants := nR + 2*marginY;
  SetDefltAndCurP(stochInfsM, nRowsR, FLOAT(nRows));
  SetDefltAndCurP(stochInfsM, nPlantsR, FLOAT(nPlants));

  InitCAModel(nRows, nPlants);

```

```

(* initial values of the cells *)
FOR x := 1 TO nRows DO
  FOR y := 1 TO nPlants DO
    IF (x <= marginX) OR (x > nRows -marginX)
    OR (y <= marginY) OR (y > nPlants-marginY) THEN (* margin area *)
      IF 0() <= basicInf THEN SetCellID(initState, x, y, secinf);
      ELSE SetCellID(initState, x, y, healthy) END(*IF*);
    SetCellID(evalPlant, x, y, false);
  ELSE
    GetMatrixEle(fieldState, nR+1-(y-marginY), x-marginX, stateR); (*the indices
      in the procedure call are exchanged, because matrix columns represent
      agronomic rows and matrix row indices the plant number within the row *)
    IF stateR >= 0.0
    THEN initStateD[x][y] := Round(stateR); (* plant was sampled *)
    ELSE initStateD[x][y] := healthy; (* plant was not sampled *)
    evalPlantD[x][y] := false END(*IF*);
  END(*IF*);
  END(*FOR*);
END(*FOR*);
InitAuxiliaries;
END SetInitialFieldState;

(******)
(* Statistical analysis of stochastic experiments *)
(******)

PROCEDURE DeclSAsNorm;
  VAR i,n: INTEGER; t0,tend,h,er,c,hm: REAL;
BEGIN
  GetGlobSimPars(t0, tend, h, er, c, hm);
  n := Round(tend-t0+1.0);

  DeclStatArray(stArCumSecInf, n); DeclStatArray(stArCumPrimInf);
  DeclStatArray(stArCumLatent, n); DeclStatArray(stArInftubers, n);
  DeclStatArray(stArInfPlants, n);

  DeclDispMV(stArCumSecInf, stochInfsM,cumPSIDisp, stochInfsM,timeIsIndep);
  DeclDispMV(stArCumPrimInf, stochInfsM,cumPFDisp, stochInfsM,timeIsIndep);
  DeclDispMV(stArCumLatent, stochInfsM,cumPLatDisp,stochInfsM,timeIsIndep);
  DeclDispMV(stArInftubers, stochInfsM,pInftubDisp,stochInfsM,timeIsIndep);
  DeclDispMV(stArInfPlants, stochInfsM,pInfp1Disp, stochInfsM,timeIsIndep);
END DeclSAsNorm;

PROCEDURE DeclSAsSensAna;
BEGIN
  AddSAOutVar(pInftub, sensTubInd[1], 'Inftub1');
  AddSAOutVar(pInftub, sensTubInd[2], 'Inftub2');
  AddSAOutVar(pInftub, sensTubInd[3], 'Inftub3');
END DeclSAsSensAna;

PROCEDURE DeclSAsStructExp;
  VAR i: INTEGER;
BEGIN
  FOR i:= 1 TO nSensTub DO
    DeclStatArray(stArSensTub[i], 1);
  END(*FOR*);
  InitSpatDistrVectors;
END DeclSAsStructExp;

PROCEDURE UndeclStochInfsStatArrays;
  VAR i: INTEGER;
BEGIN
  RemoveStatArray(stArCumSecInf); RemoveStatArray(stArCumPrimInf);
  RemoveStatArray(stArCumLatent); RemoveStatArray(stArInftubers);
  RemoveStatArray(stArInfPlants);
  FOR i:= 1 TO nSensTub DO
    RemoveStatArray(stArSensTub[i]);
  END(*FOR*);
END UndeclStochInfsStatArrays;

PROCEDURE SampleTime(): BOOLEAN; (* checks whether a sample time has been reached *)
  VAR k: INTEGER;
BEGIN
  k := CurrentStep();
  RETURN (k = Round(killD))
    OR (k = Round(emergD) + Round(killD-emergD) DIV nSensTub)
    OR (k = Round(emergD) + (2*Round(killD-emergD)) DIV nSensTub);
END SampleTime;

```

```

PROCEDURE PutValuesNorm;
  VAR t: REAL;
BEGIN
  t := CurrentTime();
  PutValue(stArCumSecInf, ind.t.cumPPl[secInf]);
  PutValue(stArCumPrimInf, ind.t.cumPPl[primInf]);
  PutValue(stArCumLatent, ind.t.cumPPl[latent]);
  PutValue(stArInfTubers, ind.t.pInfTub);
  PutValue(stArInfPlants, ind.t.pInfPl);
  INC(ind);
END PutValuesNorm;

PROCEDURE PutValuesSensAna;
BEGIN
  IF SampleTime() THEN
    CollectOutValue(sensTubInd[ind]);
    AddCurPatternToData(ind);
    DisplaySimData(ind);
    INC(ind);
  END (*IF*);
END PutValuesSensAna;

PROCEDURE PutValuesStructExp;
BEGIN
  IF SampleTime() THEN
    PutValue(stArSensTub[ind], 1, 0.0, pInfTub);
    IF calcEvenness THEN
      AddCurPatternToData(ind);
      DisplaySimData(ind);
    END (*IF*);
    INC(ind);
  END (*IF*);
END PutValuesStructExp;

PROCEDURE GetInfTubMeans(VAR infTub: ARRAY OF REAL);
  VAR i: INTEGER; dummy: REAL;
BEGIN
  FOR i := 1 TO nSensTub DO
    GetSingleStatistics(stArSensTub[i].l,dummy,dummy,dummy,dummy,dummy,infTub[i-1],dummy,
                        dummy, prob950);
  END (*FOR*);
END GetInfTubMeans;

(* **** Simulation activities *)
(* ****)

PROCEDURE ResetInputCAs;
BEGIN
  FillCAModD(plantInoculated, false);
  FillCAModD(plantRogued, false);
END ResetInputCAs;

PROCEDURE CalcInfPlant(ca: CAMod; x,y: INTEGER);
  (* prob. that plant has at least 1 inf. tuber in the sample *)
BEGIN
  infPlantsK^x^y := InfPlantProb(infTubersK^x^y);  (*part of Eq 6.28*)
END CalcInfPlant;

PROCEDURE ContribGrps; (*Eq 7.2*)
  VAR g: SpeciesGroup;
  sumInfs: REAL; (* infections during one time step *)
BEGIN
  sumInfs := 0.0;
  FOR g:= MIN(SpeciesGroup) TO MAX(SpeciesGroup) DO
    infectionsGrp[g] := infectionsGrp[g] + contribGrp[g]*effInfections;
    sumInfs := sumInfs + infectionsGrp[g];
  END (*FOR*);
  FOR g:= MIN(SpeciesGroup) TO MAX(SpeciesGroup) DO
    IF sumInfs > 0.0
    THEN propInfGrp[g] := infectionsGrp[g]/sumInfs;
    ELSE propInfGrp[g] := 0.0;
    END (*IF*);
  END (*FOR*);
  propInfc0 := propInfGrp[grp1] + propInfGrp[grp2a];
END ContribGrps;

```

```

PROCEDURE CalcOutput;
  VAR nInState: ARRAY [healthy..removed] OF LONGINT;
BEGIN
  SetEvalPlantCond; (* 'eval' plants *)
  CAFrequencies(infState, nInState, propEvalPlants, TRUE);
  CAmean      (inftubers, pInftTub, TRUE);
  CAmean      (infPlants, pInftPl, TRUE);

  SetAllPlantCond; (* all plants *)
  CAFrequencies(infState, nInState, propPlants, TRUE);

  (* cummulated proportions *)
  cumPP1[secInf] := propEvalPlants[secInf];
  cumPP1[primInf] := propEvalPlants[secInf] + propEvalPlants[primInf];
  cumPP1[latent] := cumPP1[primInf] + propEvalPlants[latent];
  cumPP1[healthy] := cumPP1[latent] + propEvalPlants[healthy];
  cumPP1[removed] := 1.0;

  inf := propPlants[secInf] + propPlants[primInf];
  Logit(stochInfsM, logitDiseased, cumPP1[latent]);

  ContribGrps;
END CalcOutput;

PROCEDURE ChoosePlantRand;
  (* random sampling of infection sources for initial state *)
  VAR infPos: Vector; infP1No: IntVector;
BEGIN
  infPos.x := UGeneral(0.0, fieldSizeM.x);
  infPos.y := UGeneral(0.0, fieldSizeM.y);
  PosToPlant(infPos, infP1No, rowDist, plantDist);
  IF infStateK'[infP1No.x][infP1No.y] = secInf THEN ChoosePlantRand END;
  SetSecInf(infState, infP1No.x, infP1No.y);
END ChoosePlantRand;

PROCEDURE GetSeedP; (* update the parameters to the current seed *)
  VAR seedXInit, seedYInit, seedZInit: INTEGER;
BEGIN
  GetSeeds(seedXInit, seedYInit, seedZInit);
  SetP(stochInfsM, seedXInitR, FLOAT(seedXInit));
  SetP(stochInfsM, seedYInitR, FLOAT(seedYInit));
  SetP(stochInfsM, seedZInitR, FLOAT(seedZInit));
END GetSeedP;

(******)

PROCEDURE Initial;
  VAR i,n: INTEGER;
BEGIN
  InitAuxiliaries;
  InitCModel(nRows, nPlants);

  SetSeeds(Round(seedXInitR), Round(seedYInitR), Round(seedZInitR));
  (* if the parameters have been changed, the changed values are used, otherwise
   * the seeds collected at the end of the last run are reassigned *)
  IF randomize < 0.5 THEN (*set initial distribution of previous run*)
    DoForAllCells(infState, SetInitState, FALSE);
  ELSEIF randomize < 1.5 THEN (* sample new initial distribution *)
    FillCModD (evalPlant, true); (* use all cells for statistics *)
    DoForAllCells(infState, SetHealthy, FALSE);
    n := Round(initInf*numOfPlants);
    FOR i:=1 TO n DO ChoosePlantRand END; (* set exactly 'n' sources *)
  ELSEIF randomize < 2.5 THEN (* use initial distribution from data file *)
    SetInitialFieldState(initStateMatrix); (* assign observed initial distribution *)
    DoForAllCells(infState, SetInitState, FALSE);
  END(*IF*);

  ResetInputCAs;

  effInfections := 0.0;
  FillArrayR(infectionsGrp, MAX(INTEGER), 0.0);

  CalcOutput;
  ind := 1; (*for statistics*)
END Initial;

PROCEDURE Terminate;
BEGIN
  GetSeedP;
  curTP;
END Terminate;

```

```

(*****)
(* Inoculations *)
(*****)

PROCEDURE InocPlantF(sourcePos: Vector; VAR inocPlNo: IntVector);
  (* inoculations by aphids dispersing by flight *)
  VAR diffPosPol: PolVector; inocPos, diffPos: Vector;
BEGIN
  diffPosPol.angle := ModAngle(windDir+VM()); (* direction follows von Mises
                                                distribution *)
  diffPosPol.length := RandWeibull(alpha, beta);(* distance Weibull distributed *)
  diffPos          := TransPolToCartVector(diffPosPol);
  inocPos          := AddVectors(sourcePos, diffPos);
  PosToPlant(inocPos, inocPlNo, rowDist, plantDist);
END InocPlantF;

PROCEDURE InocPlantIm(VAR inocPlNo: IntVector);
  (* inoculations by immigrating aphids *)
  VAR inocPos: Vector;
BEGIN
  inocPos.x := UGeneral(0.0, fieldSizeM.x);
  inocPos.y := UGeneral(0.0, fieldSizeM.y); (* random choice of cell, Eq 6.21 *)
  PosToPlant(inocPos, inocPlNo, rowDist, plantDist);
  SetCellD(plantInoculated, inocPlNo.x, inocPlNo.y, true);
END InocPlantIm;

PROCEDURE InocPlantWL(source: IntVector; VAR inocPlNo: IntVector);
  (* inoculation of plants by aphids walking over leaf bridges (Eq 6.19) *)
BEGIN
  IF U() <= walkProbWithin THEN (* movement within row *)
    inocPlNo.x := source.x;
    inocPlNo.y := source.y+TRUNC(Sign(U())-0.5);
  ELSE
    (* movement across rows *)
    inocPlNo.x := source.x+TRUNC(Sign(U())-0.5);
    inocPlNo.y := source.y;
  END (*IF*);
END InocPlantWL;

PROCEDURE InocPlantWS(sourcePos: Vector; VAR inocPlNo: IntVector);
  (* inoculation of plants by aphids walking on bare soil *)
  VAR diffPosPol: PolVector; inocPos, diffPos: Vector;
BEGIN
  diffPosPol.angle := UGeneral(0.0, 2.0*pi()); (* uniform circular distribution *)
  diffPosPol.length := RandExp(walkDist); (* exponential distribution *)
  diffPos          := TransPolToCartVector(diffPosPol);
  inocPos          := AddVectors(sourcePos, diffPos);
  PosToPlant(inocPos, inocPlNo, rowDist, plantDist);
END InocPlantWS;

PROCEDURE InoculatePlant(inocPlNo: IntVector); (* set the state change in case of
  inoculation. Cell coordinates outside the field plot are transformed first *)
  VAR inside: BOOLEAN;
BEGIN
  Torus(inocPlNo, fieldSize, inside); (* transform coordinates *)
  IF isTorus OR inside THEN SetCellD(plantInoculated, inocPlNo.x, inocPlNo.y, true)
END (*IF*);
END InoculatePlant;

PROCEDURE InternInocs(ca: CAMod; x, y: INTEGER);
  (*inoculations from within the field*)
  VAR i,InocInt: INTEGER; inocPlNo,source: IntVector; sourcePos: Vector;
BEGIN
  IF InfectionSource(infStateK^|x|^|y|) THEN (* plant is an infection source *)
    source.x := x; source.y := y;
    PlantToPos(source, sourcePos, rowDist, plantDist);
    (* calculate metric coordinates *)

    (*** inoculation by flight ***)
    nInocInt := RandRealToInt(inoculationsF);
    FOR i := 1 TO nInocInt DO
      IF distrType < 0.5
      THEN InocPlantIm (inocPlNo);
      ELSE InocPlantF(sourcePos, inocPlNo);
      END(*IF*);
      InoculatePlant (inocPlNo);
    END(*FOR*);
  END

```

```

(** inoculation by walking over leaf bridges **)
nInocInt := RandRealToInt(inoculationsWL);
FOR i := 1 TO nInocInt DO
  IF distrType < 0.5
  THEN InocPlantIm      (inocPlNo);
  ELSE InocPlantWL(source, inocPlNo);
  END(*IF*);
  InoculatePlant      (inocPlNo);
END(*FOR*);

(** inoculation by walking on soil **)
nInocInt := RandRealToInt(inoculationsWS);
FOR i := 1 TO nInocInt DO
  IF distrType < 0.5
  THEN InocPlantIm      (inocPlNo);
  ELSE InocPlantWS(sourcePos,inocPlNo);
  END(*IF*);
  InoculatePlant      (inocPlNo);
END(*FOR*);
END (*IF*);
END InternInocs;

PROCEDURE ExternInocs; (*inoculations by immigrating aphids*)
  VAR i,nInocInt: INTEGER; inocPlNo: IntVector;
BEGIN
  nInocInt := RandRealToInt(inoculationsIm*immigrTransm*numOfPlants); (*Eq 6.20*)
  FOR i := 1 TO nInocInt DO
    InocPlantIm      (inocPlNo);
    InoculatePlant(inocPlNo);
  END(*FOR*);
END ExternInocs;

PROCEDURE NewState(ca: CAMod; x,y: INTEGER);
(*sets the new disease state to a cell (Eq 6.24) *)
PROCEDURE LatPer(sens: REAL); (* latent period of a infected plant *)
BEGIN
  IF U() < sens*propInLeaves
  THEN SetPars(latPerMean,latPerSD);
  RETURN Rmax(0.0, N());
  ELSE RETURN 1.0E30; (* 'infinity', i.e. the plant becomes never a source *)
  END(*IF*);
END LatPer;

PROCEDURE InfTubDelay(): REAL;
(* delay between plant infection and tuber infection *)
BEGIN
  SetPars(infTubDel,infTubSD);
  RETURN Rmax(0.0, N());
END InfTubDelay;

BEGIN (* NewState *)
  IF CurrentTime() <= killID THEN
(* Roguing: secInf/primInf --> removed *)
    IF PlantRoguedK^(x)^[y] = true
    THEN infStateK^(x)^[y] := removed;
(* Outbreak: latent --> primInf *)
    ELSIF (infStateK^(x)^[y] = latent) AND (pA >= outbreakAgeC^(x)^[y])
    THEN infStateK^(x)^[y] := primInf;
(* Infection: healthy --> latent *)
    ELSIF (infStateK^(x)^[y] = healthy) AND (plantInoculatedK^(x)^[y] = true)
    THEN infStateK^(x)^[y] := latent; effInfections := effInfections+1.0;
        sensitivityC^(x)^[y] := ageSens;
        outbreakAgeC^(x)^[y] := pA + LatPer(sensitivityC^(x)^[y]);
        tuberInfAgeC^(x)^[y] := pA + InfTubDelay();
(* No state change *)
  ELSE infStateK^(x)^[y] := infStateK^(x)^[y];
  END(*IF*);

  ELSE (* no state change after haulm-killing *)
    infStateK^(x)^[y] := infStateK^(x)^[y];
  END(*IF*);
END NewState;

PROCEDURE InfTubers(ca: CAMod; x,y: INTEGER);
(* fraction of tubers infected (Eq 6.25) *)
BEGIN
  IF CurrentTime() <= killID THEN
    IF ((infStateK^(x)^[y] = primInf) OR (infStateK^(x)^[y] = latent))
    AND (pA >= tuberInfAgeC^(x)^[y])
    THEN infTubersK^(x)^[y] := sensitivityC^(x)^[y]; (*tubers partly infected*)
    ELSIF (infStateK^(x)^[y] = secInf)
    THEN infTubersK^(x)^[y] := 1.0; (*tubers 100 % infected*)
  END(*IF*);
END InfTubers;

```

```

    ELSE infTubersK^|x|^|y| := 0.0;                                (*no tubers infected*)
    END(*IF*);
END(*IF*);
END InfTubers;

PROCEDURE Output;
BEGIN
  DoForAllCells(infState, InfTubers, FALSE);
  DoForAllCells(infPlants, CalcInfPlant, FALSE);
  CalcOutput;

  curOP; (* client procedure *)
END Output;

PROCEDURE RoguePlant(ca: CAMod; x,y:INTEGER);
BEGIN
  IF ((infStateK^|x|^|y| = secInf) AND (U() <= rogEffS))
  OR ((infStateK^|x|^|y| = primInf) AND (U() <= rogEffP))
  THEN plantRogueK^|x|^|y| := true;
  END(*IF*);
END RoguePlant;

PROCEDURE Roguing;
  VAR rogInd: INTEGER;
BEGIN
  rogInd := 1;
  WHILE (rogInd<=nRogTimes) AND (CurrentStep()#Round(rogTime[rogInd])) DO
    INC(rogInd);
  END(*WHILE*);
  IF rogInd<=nRogTimes THEN
    DoForAllCells(infState, RoguePlant, FALSE);
  END(*IF*);
END Roguing;

PROCEDURE Input;
  VAR curWindDir: PolVector;
BEGIN
  ResetInputCAs;
  Roguing;
  ageSens := youngLeafTissue;
  curWindDir := GetWind(); (*input from PAVMetHour*)
  effInfections := 0.0;
  windbir := CompassToAngle(curWindDir.angle-orientation-Pi());
  IF (inoculationsF+inoculationsWL+inoculationsWS+inoculationsIm > 0.0)
  (* inoculations occur *)
  AND (propPlants[healthy] > 0.0) THEN
    DoForAllCells(infState, InternInocs, FALSE);
    ExternInocs;
  END(*IF*);
END Input;

PROCEDURE Dynamic;
BEGIN
  DoForAllCells(infState, NewState, FALSE);
  UpdateCAState(infState);
END Dynamic;

(*****)
(* Declarations *)
(*****)

PROCEDURE InfoWindow;
BEGIN
  WriteString('"PAVStochInfs"'); WriteLn;
  WriteLn;
  WriteString(' Stochastic spatial infection model of the EPOVIR model.');// WriteLn;
  WriteString(' Stochastic cellular automaton, modelling the plants');// WriteLn;
  WriteString(' as cells of a cellular automaton.');// WriteLn;
  WriteLn;
  WriteString(' Uses: PAVInocs & PotatoMod.');// WriteLn;
END InfoWindow;

PROCEDURE ModelObjects;
  VAR ds,i: INTEGER; labelStr: ARRAY [0..50] OF CHAR;
BEGIN
  FOR ds:= healthy TO removed DO
    labelStr := 'prop'; Append(labelStr, diseaseStatusLabel(ds));

```

```

DeclIndMV(propPlants[ds],0.0,1.0,labelStr,diseaseStateLabel[ds],
  '/100',noF,noT,noG,mInd,demo);
END(*FOR*);

DeclIndMV(cumPSIDisp,          0.0,1.0,'stats of cum prop sec inf',
  'cumPSIDisp','/100',noF,noT,noG,mInd,demo);
DeclIndMV(cumPPI[primInf],0.0,1.0,'cum prop prim inf',
  'cumPPI[primInf]','/100',noF,noT,noG,mInd,sDem);
DeclIndMV(cumPPIDisp,          0.0,1.0,'stats of cum prop prim inf',
  'cumPPIDisp','/100',noF,noT,noG,mInd,demo);
SetDefltCurveAttrForExMV(stochInfsM,cumPSIDisp,sapphire,unbroken,OC);
  cumPSIDisp := -1.0;
SetDefltCurveAttrForExMV(stochInfsM,cumPPI[primInf],ruby,spotted,'L');
SetDefltCurveAttrForExMV(stochInfsM,cumPPIDisp,    ruby,spotted,'L');
  cumPPIDisp := -1.0;

DeclIndMV(cumPPI[latent],0.0,1.0,'cum prop latent',
  'cumPPI[latent]','/100',noF,noT,noG,mInd,sDem);
DeclIndMV(cumPlatDisp,        0.0,1.0,'stats of cum prop latent',
  'cumPlatDisp','/100',noF,noT,noG,mInd,demo);
SetDefltCurveAttrForExMV(stochInfsM,cumPPI[latent],sapphire,unbroken,OC);
SetDefltCurveAttrForExMV(stochInfsM,cumPlatDisp,    sapphire,unbroken,OC);
  cumPlatDisp := -1.0;

DeclIndMV(pInfTub,            0.0,1.0,'Infected tubers',
  'pInfTub','/100',noF,noT,isy,mInd,sDem);
DeclIndMV(pInfTubDisp,        0.0,1.0,'Infected tubers stat',
  'pInfTubDisp','/100',noF,noT,noG,mInd,demo);
SetDefltCurveAttrForExMV(stochInfsM,pInfTub,      pink,spotted,'T');
SetDefltCurveAttrForExMV(stochInfsM,pInfTubDisp,pink,spotted,'T');
  pInfTubDisp := -1.0;

DeclIndMV(pInfP1,             0.0,1.0,'Infected Plants',
  'pInfP1','/100',noF,noT,noG,mInd,demo);
DeclIndMV(pInfP1Disp,         0.0,1.0,'Infected Plants stat',
  'pInfP1Disp','/100',noF,noT,noG,mInd,demo);
SetDefltCurveAttrForExMV(stochInfsM,pInfP1,      turquoise,spotted,'P');
SetDefltCurveAttrForExMV(stochInfsM,pInfP1Disp,turquoise,spotted,'P');
  pInfP1Disp := -1.0;

DeclIndMV(inf,                0.0, 1.0,'infectious',           'inf',
  '/100',noF,noT,noG,mInd,demo);
DeclIndMV(logitDiseased,-5.0, 10.0,'Logit(diseased)',       'logitDiseased',
  '-',noF,noT,noG,mInd,demo);
DeclIndMV(effInfections,0.0,10000.0,'effective infections','effInfections',
  '-',noF,noT,noG,mInd,int2);
DeclIndMV(ageSens,0.0,1.0,'age dependent sensibility','ageSens',
  '/100',noF,noT,noG,mInd,sDem);
SetDefltCurveAttrForExMV(stochInfsM,ageSens,emerald,broken,OC);

DeclIndMV(propInfGrp(grp1), 0.0,1.0,
  'Contribution of vector group 1 to the infections','propInfGrp(grp1)',
  '/100',noF,noT,noG,mInd,int2);
DeclIndMV(propInfGrp(grp2a), 0.0,1.0,
  'Contribution of vector group 2a to the infections','propInfGrp(grp2a)',
  '/100',noF,noT,noG,mInd,int2);
DeclIndMV(propInfGrp(grp2b), 0.0,1.0,
  'Contribution of vector group 2b to the infections','propInfGrp(grp2b)',
  '/100',noF,noT,noG,mInd,int2);
DeclIndMV(propInfGrp(grp3), 0.0,1.0,
  'Contribution of vector group 3 to the infections','propInfGrp(grp3)',
  '/100',noF,noT,noG,mInd,int2);
DeclIndMV(propInfCol,        0.0,1.0,
  'Contribution of colonizing vectors to the infections','propInfCol',
  '/100',noF,noT,noG,mInd,int2);

DeclIndP(nRowsR, 40.0,1.0,1000.0,noRtc,'# of rows',           'nRowsR',
  '#',mInd,sDem);
DeclIndP(nPlantsR,80.0,1.0,1000.0,noRtc,'# of plants in row',     'nPlantsR',
  '#',mInd,sDem);
DeclIndP(initInf, 1.0,0.0,  1.0,noRtc,'prop initial infection', 'initInf',
  '/100',mInd,sDem);
DeclIndP(basicInf, 0.0,0.0,  1.0,noRtc,'basic initial infection','basicInf',
  '/100',mInd,sDem);

DeclIndP(infTubDel,63.2,0.0,1000.0,rtc,'tuber infection delay (mean)',
  'infTubDel','DD',mInd,sDem);
DeclIndP(infTubSD, 26.4,0.0,1000.0,rtc,'tuber infection delay (stDev)',
  'infTubSD','DD',mInd,sDem);

DeclIndP(latPerMean,233.2,0.0,2000.0,rtc,'latency period mean',
  'latPerMean','DD',mInd,sDem);
DeclIndP(latPerSD, 90.0,0.0,1000.0,rtc,'latency period stDev',
  'latPerSD','DD',mInd,sDem);

```

```

DeclIndP(propInfLeaves,0.692,0.0,1.0,rtc,'propInfLeaves',
  'propInfLeaves','%/100',mInd,sDem);
DeclIndP(immigrTransm,0.0,0.0,1.0,rtc,
  'prop. immigrants coming from sources','immigrTransm','%/100',mInd,sDem);

DeclIndP(alpha,0.657,0.0, 10.0,rtc,'alpha flight distance', 'alpha',
  '-',mInd,sDem);
DeclIndP(beta, 9.613,0.0,100.0,rtc,'beta flight distance', 'beta',
  '-',mInd,sDem);
DeclIndP(kappa,0.345,0.0, 6.2,rtc,'kappa flight direction','kappa',
  '-',mInd,sDem);

DeclIndP(walkDist,0.81,0.0,10.0, rtc,'mean walking distance','walkDist',
  'm',mInd,sDem);
DeclIndP(nTubSample,3.0,1.0,1000.0,rtc,'tubers per sample',
  'nTubSample','#/pi',mInd,sDem);
IF UserLevel() demo THEN margin:=13.0 ELSE margin:=5.0 END(*IF*);
DeclIndP(margin,margin,0.0, 100.0,rtc,'margin around field plot',
  'margin','m',mInd,sDem);
DeclIndP(orientation,0.0,0.0,2.0*pi(),rtc,'orientation of field',
  'orientation','rad',mInd,sDem);

(*switch parameter*)
DeclIndP(torus, 1.0,0.0,1.0,noRtc,'0=no torus,1=torus',
  'torus','-',mInd,sDem);
DeclIndP(randomize,1.0,0.0,2.0, rtc,'0=reset,1=normal,2=predef.',
  'randomize','-',mInd,sDem);
DeclIndP(distrType,1.0,0.0,1.0, rtc,'distr.type: 0=random,1=EPOVIR',
  'distrType','-',mInd,sDem);

FOR i:= 1 TO nRogTimes DO
  AddIndex(labelStr,'rogTime',1,TRUE);
  DeclIndP(rogTime[i],140.0+FLOAT(i*10),0.0,366.0,rtc,'roguing time',labelStr,
  '-',mInd,sDem);
END(*FOR*);
DeclIndP(rogEffs, 0.0, 0.0,1.0,rtc,'roguing efficiency sec.',
  'rogEffs','%/100',mInd,sDem);
DeclIndP(rogEffP, 0.0, 0.0,1.0,rtc,'roguing efficiency prim.',
  'rogEffP','%/100',mInd,sDem);

DeclIndP(seedXInitR,0.0,FLOAT(MIN(INTEGER)),FLOAT(MAX(INTEGER)),rtc,'seedXInit',
  'seedXInit','-',mInd,sDem);
DeclIndP(seedYInitR,0.0,FLOAT(MIN(INTEGER)),FLOAT(MAX(INTEGER)),rtc,'seedYInit',
  'seedYInit','-',mInd,sDem);
DeclIndP(seedZInitR,0.0,FLOAT(MIN(INTEGER)),FLOAT(MAX(INTEGER)),rtc,'seedZInit',
  'seedZInit','-',mInd,sDem);
GetSeedP;
SetDefAndCurP(stochInfsM,seedXInitR,seedXInitR);
SetDefAndCurP(stochInfsM,seedYInitR,seedYInitR);
SetDefAndCurP(stochInfsM,seedZInitR,seedZInitR);
END ModelObjects;

PROCEDURE DeclStochInfs;
BEGIN
  DeclIndM(stochInfsM,discreteTime,Initial,Input,Output,Dynamic,Terminate,
    ModelObjects,'EPOVIR spatial infection model','PAVStochInfs',InfoWindow,mInd);
  InitCAModel(Round(nRowsR),Round(nPlantsR));
  InstallU(U);
END DeclStochInfs;

PROCEDURE UndeclStochInfs;
BEGIN
  RemoveM(stochInfsM);
  RemoveCAMods;
  UndeclStochInfsStatArrays;
END UndeclStochInfs;

PROCEDURE InitVars;
  VAR i: INTEGER;
BEGIN
  infState := nonexistentCAMod; plantInoculated := nonexistentCAMod;
  plantRogued := nonexistentCAMod; infTubers := nonexistentCAMod;
  infPlants := nonexistentCAMod; sensitivity := nonexistentCAMod;
  tuberInfAge := nonexistentCAMod; outbreakAge := nonexistentCAMod;
  initState := nonexistentCAMod; evalPlant := nonexistentCAMod;

  stArCumSecInf := notExistingStatArray; stArCumPrimInf := notExistingStatArray;
  stArCumLatent := notExistingStatArray; stArInftTubers := notExistingStatArray;
  stArCumPlants := notExistingStatArray;
  FOR i:= 1 TO nSensTub DO stArSensTub[i] := notExistingStatArray END(*FOR*);
  curOP := DoNothing; curTP := DoNothing;

```

```
initStateMatrix := notExistingMatrix;
END InitVars;
```

```
BEGIN
  InitVars;
END PAVStochInfs.
```

PAVSPATDISTR [15] D

```
DEFINITION MODULE PAVSpatDistr;
(* **** *)
  Purposes
    - assignment of the initial distribution of secondary sources
    - statistical analysis of the spatial pattern by means of the
      distance class statistics

  Programming
    • Design: T. Nemecek 10.01.91
    • Implementation: T. Nemecek 10.01.91

    Last revision of definition: 09.11.92 TN
(* **** *)
FROM DDFData IMPORT DataFrame;
FROM SimBase IMPORT Model;
FROM PAVBase IMPORT nSensTub;
FROM SpatPattern IMPORT 'SpatPat';

VAR spatDistrMod: Model;
  simSP: ARRAY [1..nSensTub] OF SpatPat; (* simulated spatial patterns *)

PROCEDURE InstallSpatPats;
PROCEDURE RemoveSpatPats;
(* declare/remove the data structures for the analysis of the spatial pattern *)
PROCEDURE SetClassLimits(plD: REAL);
(* assigns the class limits, which are calculated as multiples of the planting
  distance 'plD' *)
PROCEDURE SetInitSpatDistr(dt: DataFrame; ident: ARRAY OF CHAR);
(* assign an initial distribution of secondary source, healthy and
  removed plants to the cellular automaton model in PAVStochInfs *)

PROCEDURE InitSpatDistrVectors;
(* resets the distance class statistics for the next experiment *)
PROCEDURE AddCurPatternToData(indl: INTEGER);
(* adds the statistics of the current spatial pattern to the distance class
  arrays. 'indl' is the index of the distance class pattern tp which the
  results are added, namely 'simSP[indl]' *)
PROCEDURE DisplaySimData(il: INTEGER);
(* displays a simulated spatial pattern in a window *)
PROCEDURE DumpCurSimPat;
(* dumps the current pattern on file PAVBase.resF, if this file is open,
  otherwise on a new file, created by means of a dialog box *)

PROCEDURE GetEvennessValues(VAR evennessVals: ARRAY OF REAL);
(* returns the evenness values for the 3 monitoring times *)

PROCEDURE DeclPAVSpatDistr;
PROCEDURE UndeclPAVSpatDistr;
END PAVSpatDistr.
```

PAVMONIT [16] D

```
DEFINITION MODULE PAVMonit;
(* **** *)
  Purpose: provides monitoring procedures for the cellular automaton
  and for stochastic simulation experiments

  Programming
    • Design: T. Nemecek 6.2.90
    • Implementation: T. Nemecek 6.2.90

    Last revision of definition: 29.01.92 tn
(* **** *)
FROM DMFiles IMPORT TextFile;
FROM SimBase IMPORT Model;
```

```

VAR pavMonitMod: Model;

PROCEDURE DeclPAVMonitoring;
PROCEDURE UndeclPAVMonitoring;
(* installation and deinstallation of the model *)
PROCEDURE InstallMonitCommands;
(* installs/removes the commands for monitoring *)
PROCEDURE OpenCAMon;
PROCEDURE CloseMonWind;
(* opens/closes a window with a 2D-graph showing the current state of the
cellular automaton *)

PROCEDURE DoMonitoring;
(* In case the output window is open, this procedure shows the current
state of the cellular automaton. Otherwise it has no effect. *)

PROCEDURE DisplayStatData;
(* displays the data series (means and (optionally) conf. intervals of several
stochastic runs) in the MR-graph *)
PROCEDURE DumpStatData(VAR f: TextFile);
(* dumps the summary statistics of stochastic experiments onto file 'f' *)
END PAVMonit.

```

PAV Management Modules

PAVSETGETOBJ [17] D

```

DEFINITION MODULE PAVSetGetObj;
(* **** *)
Purpose
- coupling of data frames with simulation models
of the PAV-simulation-system
- management of SVs and MVs

Programming
* Design: T. Nemecek 08.11.1990
* Implementation: T. Nemecek 08.11.1990

Last revision of definition: 09.11.92 TN
*****)
FROM DDFData IMPORT DataFrame;

TYPE
MonitType = (noMonit, normal, ITMon, srcMon, validation);

PROCEDURE SetGeneralObjs (df: DataFrame); (* general objects
(GlobSimPars etc.) *)

PROCEDURE SetValidModObjs (df: DataFrame); (* validation model *)
PROCEDURE SetInputDataObjs (df: DataFrame); (* general input models *)
PROCEDURE SetVanderplankObjs(df: DataFrame);
PROCEDURE SetSignalObjs (df: DataFrame);
PROCEDURE SetRuesinkObjs (df: DataFrame);
PROCEDURE SetJohnsonObjs (df: DataFrame);
PROCEDURE SetEPOVIRDetObjs (df: DataFrame);
PROCEDURE SetEPOVIRSpObjs (df: DataFrame);

PROCEDURE SetMonitoring( monitType: MonitType );
PROCEDURE GetMonitoring(VAR monitType: MonitType);

PROCEDURE InitPAVSetGetObj;
PROCEDURE TermPAVSetGetObj;
END PAVSetGetObj.

```

PAVMOEDMNGR [18] D

```

DEFINITION MODULE PAVMoEDMngr;
(* **** *)
Purpose
Experiment definition in the PAV simulation system. Allows to load data
frames and models interactively and under program control

```

```

Abbreviations:
  DF    = data frame
  M     = model

Programming
  • Design:      T. Nemecek      26.09.90
  • Implementation: T. Nemecek   07.11.90

Last revision of definition: 08.04.92 TN
*****)

PROCEDURE ReadAndSetDF(dfInd: INTEGER); (* reads the DF with index number 'dfInd'
  from the file and assigns its objects to all active models. This version keeps
  only the currently active data frame in memory. *)

PROCEDURE ActivateModel  (mIndex: INTEGER);
PROCEDURE DeactivateModel (mIndex: INTEGER);

PROCEDURE InitPAVMoEDMngr; (* initializes the PAV simulation system *)
PROCEDURE TermPAVMoEDMngr;

PROCEDURE ShowLogo(init: BOOLEAN);
(* shows the PAV-logo. 'init' indicates if the logo is shown during initialization
  (automatic removal) or not (removal by mouse click in the window *) )
PROCEDURE HideLogo; (* closes the logo window *)

(* 'AddInitSimSessionProc' allows to install an additional InitSimSession
  procedure. Use this mechanism to install such a procedure, since PAVMoEDMngr
  installs already installs one. *)
PROCEDURE AddInitSimSessionProc( iss: PROC);
PROCEDURE GetInitSimSessionProc(VAR iss: PROC);
END PAVMoEDMngr.

```

PAVEXPMNTS [19] D

```

DEFINITION MODULE PAVEExpmts;
(* ****)
  Purpose
    simulation experiment management of the PAVSS

  Programming
    • Design:      T. Nemecek      28.01.91
    • Implementation: T. Nemecek      28.01.91

  Last revision of definition: 21.02.91 TN
*****)

PROCEDURE InstallExperiment; (* installs the model and the standard experiment *)
PROCEDURE RemoveExperiment;
END PAVEExpmts.

```

PAVMASTER [20] I

```

MODULE PAVMaster;
(* Master module of the PAV-Simulation-System. Standard version *)
FROM PAVMoEDMngr  IMPORT InitPAVMoEDMngr,TermPAVMoEDMngr>ShowLogo;
FROM PAVEExpmts  IMPORT InstallExperiment;
FROM SIMMaster   IMPORT RunSimMaster;
FROM PAVBase     IMPORT SetUserLevel,demo,int2;
FROM DMLanguage   IMPORT SetLanguage,Language;

PROCEDURE InitPAV;
BEGIN
  InitPAVMoEDMngr;
  InstallExperiment;
END InitPAV;

BEGIN
  SetLanguage (English);
  ShowLogo (TRUE);
  SetUserLevel(int2); (*. <== define the user level here *)
  RunSimMaster(InitPAV);
  TermPAVMoEDMngr;
END PAVMaster.

```

APPENDIX V: LISTINGS OF AUXILIARY LIBRARY MODULES

Experiment Definition Modules

The 3 following definition modules contain 83 lines (28 lines/module) with 4900 characters (1629 characters/module) in the source code (comments and empty lines not included); the corresponding implementation modules 1220 lines (407 lines/module) with 41600 characters (13866 characters/module) and the corresponding object code 19200 bytes (6395 bytes/module, 'Compile20' used).

DFDATA D

```

DEFINITION MODULE DFDATA;
(* ****)
  Purpose
    provides means to store data frames

  Remarks
    This module is independent of the file format used in ReadWriteDF and from
    the simulation environment used in SetGetDF. It stores an arbitrary number
    of DataFrames. The objects are accessed by their identifiers, which may be
    qualified by a model identifier. The (optionally qualified) identifiers of
    the Scalars and the DFMATRIXObjs must be unique within these two groups.
    Identifier conflicts are checked and reported. It is not possible to add a
    second object with the same name to the data frame. In the current version
    it is not necessary that the identifiers of the data frames are unique.
    Conflicts of these identifiers are not checked.

  Legend
    DF      = data frame
    GSP     = global simulation parameters
    Scalar  = an object specifying a real value together with an identifier
              a descriptor, unit, minimum and maximum value.
    DFMATRIX = a matrix of real values, containing further a title and
               and subTitle and the labels of the matrix columns
    DFMATRIXObj = this object contains an identifier and a filename, where the
                  matrix is located

  Programming
    • Design:          T. Nemecek, A. Fischlin, O. Roth           23.10.1990
    • Implementation: T. Nemecek                                     23.10.1990

    Last revision of definition: 17.03.92 TN
(* ****)

FROM Matrices  IMPORT Matrix;
FROM DMStrings IMPORT String;

TYPE
  DataFrame;
  DFACTION      = PROCEDURE(VAR DataFrame);
  ScalarAction   = PROCEDURE(  DataFrame, ARRAY OF CHAR);
  DFMATRIXAction = PROCEDURE(  DataFrame, ARRAY OF CHAR);

  VAR
    objFound:  BOOLEAN; (* the variable has the value FALSE, if one of the
                           following objects could not be found: DFMATRIXObj, Scalar *)
    notExistingDF:           DataFrame; (* read only *)

(* ****)
(*  Data frames  *)
(* ****)

PROCEDURE DeclDF (VAR df: DataFrame; ident: ARRAY OF CHAR);
PROCEDURE RemoveDF(VAR df: DataFrame);

PROCEDURE DFExists(df: DataFrame): BOOLEAN;
PROCEDURE FindDF(ident: ARRAY OF CHAR): DataFrame;
(* If the DF with ident is not found in the list, FindDF returns NIL *)
PROCEDURE GetDFIdent(df: DataFrame; VAR ident: ARRAY OF CHAR);

```

```

PROCEDURE DoForAllDFs(do: DFACTION);
PROCEDURE SetDFTitles(df: DataFrame; title, subTitle: ARRAY OF CHAR);
PROCEDURE GetDFTitles(df: DataFrame; VAR title, subTitle: ARRAY OF CHAR);
(* global simulation parameters *)
PROCEDURE SetDFGSP (df: DataFrame; fileName: ARRAY OF CHAR;
    t0, tend, h, er, c, hm: REAL);
PROCEDURE GetDFGSP (df: DataFrame; VAR fileName: ARRAY OF CHAR;
    VAR t0, tend, h, er, c, hm: REAL);

(******)
(* Scalars *)
(******)
PROCEDURE AddScalar (df: DataFrame; ident, descr, unit: ARRAY OF CHAR;
    value, min, max: REAL);
PROCEDURE RemoveScalar (df: DataFrame; ident: ARRAY OF CHAR);
PROCEDURE RetrieveScalar (df: DataFrame; ident: ARRAY OF CHAR;
    VAR descr, unit: ARRAY OF CHAR; VAR value, min, max: REAL);
PROCEDURE ModifyScalar (df: DataFrame; ident,
    descr, unit: ARRAY OF CHAR; value, min, max: REAL);
PROCEDURE ScalarExists (df: DataFrame; ident: ARRAY OF CHAR): BOOLEAN;
PROCEDURE DoForAllScalars(df: DataFrame; do: ScalarAction);

(******)
(* DFMatrices and DFMATRIXObjs *)
(******)
(* The DFMatrices are handled in two steps:
   1. the DFMATRIXObj must be added to the data frame with AddDFMatrixObj.
   2. optionally a DFMatrix can be added to the resp. DFMATRIXObj with
      AddDFMatrix. Step 2 requires step 1. This procedure allows to fetch the
      data of the DFMatrix only in case of need. *)
PROCEDURE AddDFMatrix ('(df: DataFrame; ident: ARRAY OF CHAR; title, subTitle:
    ARRAY OF CHAR; labels: ARRAY OF String; data: Matrix);
PROCEDURE RemoveDFMatrix (df: DataFrame; ident: ARRAY OF CHAR);
PROCEDURE RetrieveDFMatrix(df: DataFrame; ident: ARRAY OF CHAR; VAR title,
    subTitle: ARRAY OF CHAR; VAR labels: ARRAY OF String; VAR data: Matrix);
PROCEDURE GetDFMatrix ('(df: DataFrame; ident: ARRAY OF CHAR; VAR title,
    subTitle: ARRAY OF CHAR; VAR labels: ARRAY OF String; VAR data: Matrix);
(* this procedure does exactly the same as RetrieveDFMatrix, but does not copy the
   matrix for efficiency reasons. If you fetch the matrix by GetDFMatrix DO NOT
   REMOVE THE MATRIX NOR THE LABELS! *)
PROCEDURE ModifyDFMatrix (df: DataFrame; ident: ARRAY OF CHAR; title,
    subTitle: ARRAY OF CHAR; labels: ARRAY OF String; data: Matrix);
PROCEDURE DFMatrixExists (df: DataFrame; ident: ARRAY OF CHAR): BOOLEAN;

(*-----)
PROCEDURE AddDFMatrixObj (df: DataFrame; ident: ARRAY OF CHAR;
    fileName: ARRAY OF CHAR);
PROCEDURE RemoveDFMatrixObj(df: DataFrame; ident: ARRAY OF CHAR);
PROCEDURE RetrieveDFMatrixObj (df: DataFrame; ident: ARRAY OF CHAR;
    VAR fileName: ARRAY OF CHAR);
PROCEDURE ModifyDFMatrixObj (df: DataFrame; ident: ARRAY OF CHAR;
    fileName: ARRAY OF CHAR);
PROCEDURE DFMatrixObjExists (df: DataFrame; ident: ARRAY OF CHAR): BOOLEAN;
PROCEDURE DoForAllDFMatrixObjs(df: DataFrame; do: DFMATRIXAction);
END DFData.

```

READWRITEDF D

```

DEFINITION MODULE ReadWriteDF;
(******)
  Purpose: Provides procedures to read and write data frame files
  Remarks: The data are read are stored in memory for further use.

  Missing files:
    If the file referencing a DFMatrix is missing, this can be denoted by the
    missingValCode of ReadData, e.g. "N".

  Programming
    • Design: T.Nemecek      23.10.90
    • Implementation: T.Nemecek 23.10.90

  Last revision of definition: 13.09.91 tn
(******)
FROM DMStrings IMPORT String;
FROM DFData    IMPORT DataFrame;

PROCEDURE ReadDataFrameFile (fn: ARRAY OF CHAR; VAR df: DataFrame;
    readDFMatrices: BOOLEAN); (* without dialog *)
PROCEDURE ReadDataFrameFile (VAR df: DataFrame; readDFMatrices: BOOLEAN);
    (* with dialog *)

```

```

PROCEDURE WriteDataFrameFile (VAR fn: ARRAY OF CHAR; df: DataFrame);
(* not yet impl. *)
(* Reads a DataFrame of a textfile with a specified format.
EBNF:
-----
DataFrame =
line      EOL      (* title *)
line      EOL      (* subTitle *)
integer   (* nScalars *)
integer   (* nMatrices *)
integer   (* nImportedDataFrames max. 32 *)
globSimPars (* global simulation parameters *)
[Scalar]
{fileObject} (* nScalars times repeated *)
{fileObject} (* data tables, nMatrices times repeated *)
{fileObject} (* imported DFs, nScalars times repeated *)

integer   = digit(digit).
number   = (digit(digit)(".")(digit)[ScaleFactor]|"N").
digit    = "0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9".
ScaleFactor = "E"[+|-]digit(digit).
stringChar = stringChar(stringChar).
stringChar = character > 32.
line     = {lineChar}.
lineChar = character ~ EOL.
globSimPars = ident fileName gsp.
gsp      = number number number number number number.
          (* to tend h relErr discStep monInt
           gsp must located on the file referenced by fileName *)
fileObject = qualIdent fileName.
Scalar    = qualIdent string string number number number.
          (* qualIdent descr unit val min max *)
ident    = letter(letter|digit).
qualIdent = [ident]."."ident.
          ~ except

Parameters:
-----
fn:           a textfile containing the data frame
df:           contains the reference of the DataFrame.
readDFMatrices: determines, whether the DFMatrices are read immedeately
                after reading of the DataFrame. If this is not the case, the
                matrices can be read later using ReadDFMatrixFile.

'Nested imports':
-----
The module allows for 'nested imports', i.e. several data frames (max. 32) can
be references from within a single data frame. These imported data frames can
themselves reference other DFs. The title, subtitle and the global simulation
parameters of the first data frame are used, all other are ignored. The
objects are simply added to the data frame df, i.e. the origin of the objects
is no longer known. *)

PROCEDURE ReadDFMatrixFile (df: DataFrame; ident: ARRAY OF CHAR);
PROCEDURE WriteDFMatrixFile (df: DataFrame; ident: ARRAY OF CHAR);
(*. not yet impl. .*)

(* Reads a DF matrix stored on a text file. The filename is stored in the
DFMatrixObj referenced by ident. The corresponding DFMatrixObj must have been
read previously by ReadDataFrameFile.
The DFMatrix on a text file must have the following format:
  1. line: title of the DFMatrix
  2. line: subtitle of the DFMatrix
  2 values of type INTEGER (1. (nRows) number of rows, 2. (nCols) number of
  columns)
  nCols labels
  the matrix: nRows*nCols values of type REAL;
  any comment can follow which will no be read

EBNF:
-----
Matrix =
line      EOL      (* title *)
line      EOL      (* subTitle *)
integer   (* nRows *)
integer   (* nCols *)
{string}   (* labels, nCols times repeated *)
{number}.  (* data, nCols*nRows times repeated*)

Example:
-----
Data Changins 1983  (* title *)
Meteo Data          (* subTitle *)
5 (*nRows*) 3 (*nCols*)
JulDay TMin[C] TMax[C] Prec[mrn]  (* labels *)
100      5.0       15.6      0.0
101      6.1       10.0      0.0

```

```

102      7.8      19.2      1.5      (* data block *)
103      4.5      16.8      0.0
104      2.2      17.2      6.8
any comment may follow - *)
END ReadWriteDF.
```

SETGETDF D

```

DEFINITION MODULE SetGetDF;
(* **** *)
Purpose
  provides procedures to set and get an experimental frame to the objects of
  ModelWorks.

Remarks
  This module is dependent on the simulation environment used.
  This version is designed and implemented for ModelWorks.

Terminology
  Set: data frame --> model
  Get: model      --> data frame

Programming
  • Design: T. Nemecsek, A. Fischlin, O. Roth   6.11.1990
  • Implementation: T. Nemecsek                 6.11.1990

Last revision of definition: 08.04.92 TN
(* **** *)
FROM SimBase IMPORT Model;
FROM DFData  IMPORT DataFrame;
FROM TabFunc IMPORT TabfUNC;

TYPE
  AssignmentMode = (onlyCur, onlyDflt, curAndDflt);
  (* determines, which values of the model objects are assigned *)

(* the identifiers of the scalars in the data frame df are compared with the
  identifiers of the model parameters and state variables. In case of a match the
  values are assigned in either direction. A match is defined:
  • Unqualified identifier in data frame (e.g. objIdent="seedWeight"):
    objIdent = objIdent
    => exact match of the identifiers
  • Qualified identifier in data frame, consisting of model identifier
    and object identifier (e.g. modIdent.objIdent="potatoMod.seedWeight"):
    (modIdent = modIdent) AND (objIdent = objIdent)
    => exact match of the model AND object identifiers *)
PROCEDURE SetScalar (df: DataFrame; ident: ARRAY OF CHAR); (* if an parameter or
  SV with identifier 'ident' is found, the scalar is assigned *)
PROCEDURE GetScalar (df: DataFrame; ident: ARRAY OF CHAR);
  (* if no scalar with 'ident' exists in df, a new scalar is created  *)
PROCEDURE SetScalarToM (df: DataFrame; ident: ARRAY OF CHAR; m: Model); (* if a
  parameter or SV with identifier 'ident' is found in 'm', the scalar is assigned*)
PROCEDURE GetScalarFromM (df: DataFrame; ident: ARRAY OF CHAR; m: Model);
  (* not yet implemented *)
  (* if no scalar with 'ident' exists in df, a new scalar is created *)
(* the following procedures require the existence of the corresponding DispData,
  TabFUNC and DFMatrix. If no corresponding object is found either in the
  DataFrame or in the simulation environment, no action is performed. The
  assignment of a DFMatrix to the sim env. requires that the number of columns
  is 2 in case of a table function, resp. 4 in case of DispData. For table
  functions, 'tabName' is used as identifier, for DispData the identifier of
  the dependent MV. The latter can be qualified by the model-name *)
PROCEDURE SetMatrix (df: DataFrame; ident: ARRAY OF CHAR);
PROCEDURE GetMatrix (df: DataFrame; ident: ARRAY OF CHAR);

PROCEDURE SetMatrices (df: DataFrame);  (* for all DFMatrices *)
PROCEDURE GetMatrices (df: DataFrame);  (* for all DFMatrices *)

PROCEDURE SetDFToM (df: DataFrame; m: Model);
  (* assigns scalars of 'df' to 'm' *)
PROCEDURE GetDFFromM (VAR df: DataFrame; m: Model);
  (* assigns Pa's and SV's from 'df' to 'm' *)

PROCEDURE SetDF      (df: DataFrame);
  (* assigns scalars and matrices of 'df' to all models *)
PROCEDURE GetDF      (VAR df: DataFrame);
  (* gets scalars and matrices from all models *)
```

```

(* assign the project description:
df.title <-> pd.title pd = project description
df.subtitle <-> pd.remark *)
PROCEDURE SetPD (df: DataFrame);
PROCEDURE GetPD (df: DataFrame);

(* assign the global simulation parameters, to the default and/or
current values, depending on the current asMode. *)
PROCEDURE SetGSP (df: DataFrame);
PROCEDURE GetGSP (df: DataFrame);

PROCEDURE SetSettingMode( useDescr,useUnit,useRange: BOOLEAN;
asMode: AssignmentMode);
PROCEDURE GetSettingMode(VAR useDescr,useUnit,useRange: BOOLEAN;
VAR asMode: AssignmentMode); (* determines, which of the values stored in the
data frame are actually assigned to the model object, and if the current,
default or both values are assigned.
defaults:
-----
useDescr = FALSE
useUnit = FALSE
useRange = FALSE
asMode = curAndDflt *)
END SetGetDF.

```

Other Auxiliary Modules

The 5 following definition modules contain 220 lines (44 lines/module) with 10400 characters (2084 characters/module) in the source code (comments and empty lines not included); the corresponding implementation modules 1840 lines (369 lines/module) with 59000 characters (11791 characters/module) in the source code, resp. 26600 bytes (5312 bytes/module) in the object code (compiled with 'Compile20').

CAMODBASE D

```

DEFINITION MODULE CAModBase;

***** Purpose
      general support for 2D cellular automata with discrete and continuous
      states

***** Terminology
      CA = cellular automaton
      update state = auxiliary variable used for update (see below)

      C = continuous
      D = discrete

***** Remarks
      3 "UpdateModes" are allowed:
      - withNewState: calling PROC UpdateCAState provokes an overwriting of
                      the state of the CA with the updateState
                      state(k+1) := updateState
      - withStateChange: calling PROC UpdateCAState provokes an addition of the
                         state of the CA and the updateState
                         state(k+1) := state(k) + updateState
      - noUpdate: calling PROC UpdateCAState provokes no action at all.
                  This mode should be chosen, if the variable caMod does
                  not reference a CA-model but an auxiliary array,
                  which is not updated. In this mode, no update-state
                  array is allocated, thus memory is saved

***** Limitations
      The maximal size of the array is 8000x8000 cells

***** Programming
      • Design:          T. Nemecek    23.4.1991
      • Implementation: T. Nemecek   23.4.1991

***** Last revision of definition: 07.08.92 TN
***** 
```

```

FROM DMFiles           IMPORT TextFile;

CONST maxlen = 8000;

TYPE
  CAMod;

CAArc  = POINTER TO RowC;           CAArd  = POINTER TO RowD;
ColCPtr = POINTER TO ColC;         ColDPtr = POINTER TO ColD;
RowC   = ARRAY [1..maxLen] OF ColCPtr; RowD   = ARRAY [1..maxLen] OF ColDPtr;
ColC   = ARRAY [1..maxLen] OF REAL;  ColD    = ARRAY [1..maxLen] OF INTEGER;

UpdateMode = (withNewState, withStateChange, noUpdate);
StateType  = (continuous, discrete);
(* continuous: REAL -values
  discrete:   INTEGER-values *)
CellProc   = PROCEDURE(CAMod, INTEGER, INTEGER);
ConditionProc = PROCEDURE(CAMod, INTEGER, INTEGER): BOOLEAN;
(* used as condition for the statistical evaluation *)
MappingProc = PROCEDURE(CAMod, REAL): INTEGER;
(* used to classify REAL-values into discrete classes for graphical monitoring
  and statistics *)
RemoveProc = PROCEDURE(CAMod);
CAModProc  = PROCEDURE(CAMod);

VAR
  nonexistentCAMod: CAMod; (* read only *)

(* **** *)
(* CA management *)
(* **** *)

PROCEDURE DeclCAMod (VAR caMod: CAMod; stateType: StateType; x, y, minState,
  maxState: INTEGER; mode: UpdateMode);
(* caMod: the reference variable for the access of the model
  stateType: see Remarks
  x,y:      the # of rows (x) and columns (y)
  minState,
  maxState: the first resp. last state. These parameters are of
  importance for the monitoring of the state in a graph.
  mode:     see Remarks *)

PROCEDURE GetStatePtrsC(caMod: CAMod; VAR stateC, updStateC: CAArc);
PROCEDURE GetStatePtrsD(caMod: CAMod; VAR stateD, updStateD: CAArd);
(* stateC, updStateC, resp. stateD, updStateD: These variables allow to
  access directly the array elements. Direct access is about 5 times faster
  than the call of the procedures SetCellC/D, resp. GetCellC/D. This can
  e.g. be done as follows: cPtr[r][c] is the element in the row r and the
  column c
  -----
  | CAUTION: no index range checking is done
  | in the case of the direct access!
  |
  If mode = noUpdate, the variables updStateC resp. updStateD are NIL *)

PROCEDURE GetCA Specs (caMod: CAMod; VAR stateType: StateType; VAR x, y, minState,
  maxState: INTEGER; VAR mode: UpdateMode);
PROCEDURE RemoveCAMod(VAR caMod: CAMod);
PROCEDURE AddRemoveProc (caMod: CAMod; rp: RemoveProc);
PROCEDURE DeleteRemoveProc(caMod: CAMod; rp: RemoveProc);
(* the remove procedure allows to remove objects, associated with caMod.
  This procedure is called BEFORE the CAMod is removed.*)

PROCEDURE CAExists(caMod: CAMod): BOOLEAN;
(* **** *)
(* Cell state *)
(* **** *)
(* The following procedures do not test for the existence of caMod. If
  required, the test can be performed with PROC CAExists. This is typically
  done at the beginning of a loop. *)
PROCEDURE SetCellC (caMod: CAMod; x, y: INTEGER; state: REAL);
PROCEDURE GetCellC (caMod: CAMod; x, y: INTEGER; VAR state: REAL);
PROCEDURE   CellC (caMod: CAMod; x, y: INTEGER):          REAL;
(* for easy access *)

PROCEDURE SetUpdCellC(caMod: CAMod; x, y: CARDINAL; updateSt: REAL);
PROCEDURE GetUpdCellC(caMod: CAMod; x, y: CARDINAL; VAR updateSt: REAL);

PROCEDURE SetCellD (caMod: CAMod; x, y: INTEGER; state: INTEGER);
PROCEDURE GetCellD (caMod: CAMod; x, y: INTEGER; VAR state: INTEGER);
PROCEDURE   CellD (caMod: CAMod; x, y: INTEGER):          INTEGER;
(* for easy access *)
(* The last 2 procedures can be used to convert the state of a CAMod with con-
  tinuous states into a discrete value using the current mapping procedure *)

```

```

PROCEDURE SetUpdCellD(caMod: CAMod; x, y: CARDINAL; updateSt: INTEGER);
PROCEDURE GetUpdCellD(caMod: CAMod; x, y: CARDINAL; VAR updateSt: INTEGER);

(*****)
(* CA state *)
(*****)

PROCEDURE InitCAState(caMod: CAMod; VAR f: TextFile); (* not yet implemented *)
(* reads the initial state from the current position of the file f,
   and assigns it to the current state of caMod *)
PROCEDURE FillCAModC(caMod: CAMod; val: REAL);
PROCEDURE FillCAModD(caMod: CAMod; val: INTEGER);
PROCEDURE UpdateCAState(caMod: CAMod);
(* updates the state of the cellular automaton, according to the current update
   mode *)
PROCEDURE ResetCAUpdateState(caMod: CAMod);
(* the elements of the update state array are reset as follows:
   updSt[i] := 0           if mode = withStateChange
   updSt[i] := st[i]       if mode = withNewState *)
(*****)
(* statistics *)
(*****)

PROCEDURE CAFrequencies(caMod: CAMod; VAR nInState: ARRAY OF LONGINT;
  VAR propInState: ARRAY OF REAL; useStatsCond: BOOLEAN);
(* calculates the statistics of the cellular automaton:
   nInState = number of cells in states 0..nStates-1
   propInState = proportion of cells in states 0..nStates-1
   If the stateType of the CA is continuous, the current mapping procedure is
   used to calculate the frequency of the discrete states. useStatsCond
   determines whether the statistics are calculated for a previously defined
   subset of cells only (TRUE) or for all cells (FALSE) *)
PROCEDURE CAMean(caMod: CAMod; VAR mean: REAL; useStatsCond: BOOLEAN);
(* calculates the mean of the cellular automaton states *)
PROCEDURE SetStatsCond(caMod: CAMod; scp: ConditionProc);
PROCEDURE GetStatsCond(caMod: CAMod; VAR scp: ConditionProc);
(* scp can be used to select a range of cells for statistical evaluation *)
PROCEDURE SetMappingProc(caMod: CAMod; mp: MappingProc);
PROCEDURE GetMappingProc(caMod: CAMod; VAR mp: MappingProc);
(* The default mapping procedure is:
   VAR c: REAL; d: INTEGER;
   d := TRUNC(MAX(FLOAT(minState), MIN(FLOAT(maxState), c))+0.5); *)

(*****)
(* auxiliary *)
(*****)

PROCEDURE SetIndexCheckMode( doCheck: BOOLEAN );
PROCEDURE GetIndexCheckMode(VAR doCheck: BOOLEAN );
PROCEDURE DoForAllCells(caMod: CAMod; p: CellProc; useStatsCond: BOOLEAN);
(* if only the indices of the procedure p are used, index range checking is
   unnecessary *)
PROCEDURE DoForAllCAMods(cp: CAModProc);
END CAModBase.

```

RANDGENS D

```

DEFINITION MODULE RandGens;
(*****)
  Purpose: A collection of random number generators
  Programming
    * Design:      T. Nemecek      20.7.90
    * Implementation: T. Nemecek 20.7.90
    Last revision of definition: 20.7.90 tn
(*****)

PROCEDURE UGeneral(min,max: REAL): REAL;
(* provides uniformly distributed random numbers in the interval [min,max]*)*
PROCEDURE RandExp(mean: REAL): REAL;
(* provides exponentially distributed random numbers with the average mean.*)*
PROCEDURE RandWeibull(alpha, beta: REAL): REAL;
(* provides Weibull distributed random variables. The 2-parametric
   Weibull distribution is used. P.d.f.:
   f(x) = alpha * beta^alpha * x^(alpha-1) * Exp(-(x/beta)^alpha) *)
PROCEDURE RandTriang(min, mode, max: REAL): REAL;
(* provides random numbers following a triangular distribution
   with the parameters min, mode, max, where
   min = lowest value
   max = highest value

```

```

    mode = coordinate of maximum *)
PROCEDURE SetVMPars(mean, kappa: REAL); (* set parameters for the von Mises
    distribution. Defaults are: mean = 0.0; kappa = 1.0 *)
PROCEDURE GetVMPars(VAR mean, kappa: REAL);
PROCEDURE VM(): REAL;
    (* provides random number from the von Mises distribution (called also the
        circular normal distribution) the values are in the interval [0, 2π *)*
END RandGens.
```

SPATPATTERN D

```

DEFINITION MODULE SpatPattern;
(* ****)
Purpose
    provides statistical procedures for the evaluation
    of distance class patterns of cellular automata

Remarks
    3 categories of cells are distinguished
    - source cells: cells from which the dispersal process starts
        (e.g. infection sources)
    - destination cells: cells where the dispersal process ends
        (e.g. infected plants)
    - neutral cells: cells which are unaffected by the dispersal process
        (e.g. healthy plants)
    - none: cells which are not included in the analysis
        (e.g. removed plants)

Statistics are calculated for the frequency distribution
of the distances between all source and all destination cells

Limitation
    a maximum of 16000 cell can be evaluated

Programming
    • Design:      T. Nemecek      3.5.1991
    • Implementation: T. Nemecek 3.5.1991

    Last revision of definition: 24.06.1991 TN
(* ****)
FROM CAModBase      IMPORT CAMod;
FROM DMFiles        IMPORT TextFile;
FROM DMWindows      IMPORT Window;

TYPE
    SpatPat;

Str255           = ARRAY [0..255] OF CHAR;
CellState         = (source, destination, neutral, none);
ClassificationProc = PROCEDURE (CAMod, VAR INTEGER, INTEGER, INTEGER):
                    CellState;
(* example:
   VAR probability: REAL;
   PROCEDURE ClassifyCell(caMod: CAMod; VAR countDown: INTEGER; x,y: INTEGER):
                    CellState;
   VAR state: INTEGER;
   BEGIN
       state := CellID(caMod, x, y);
       IF state = srce
       THEN RETURN source
       ELSE IF countDown = 0
       THEN countDown := 10
       ELSE DEC(countDown) END(*IF*);
       IF U() <= probability
       THEN RETURN destination
       ELSE RETURN neutral END(*IF*);
   END(*IF*);
END ClassifyCell;

Usage: this procedure allows the client to install a
classification procedure.
x,y:           indices of the cell
countDown: variable allowing a multiple classification of the same
            cell. The procedure is called until countDown=0. Before
            the first call of the procedure for the same cell, the
            value of countDown is 0. If the procedure is called
            only once, for each cell, is not necessary to assign
            any value to countDown. *)

VAR notExistingSpatPat: SpatPat; (* read only *)
```

```

(* **** *)
(* Management *)
(* **** *)
PROCEDURE SpatPatExists(sp: SpatPat): BOOLEAN;
PROCEDURE DeclSpatPat(VAR sp: SpatPat; caMod: CAMod; title, subTitle: Str255;
  rowD,colD: REAL; nCl: INTEGER; classLimits: ARRAY OF REAL;
  cp: ClassificationProc);

PROCEDURE GetSpatPatSpecs(sp: SpatPat; VAR caMod: CAMod; VAR title,
  subTitle: Str255; VAR rowD, colD: REAL; VAR nCl: INTEGER;
  VAR classLimits: ARRAY OF REAL; VAR cp: ClassificationProc);

PROCEDURE RemoveSpatPat(VAR sp: SpatPat);

(* **** *)
(* Statistics *)
(* **** *)
PROCEDURE InitDistClassStats      (sp: SpatPat);
PROCEDURE AnalyzedDistClassPattern(sp: SpatPat);
PROCEDURE GetDistClassStats      (sp: SpatPat; VAR relFreqs: ARRAY OF REAL;
  VAR sdFreqs: ARRAY OF REAL; VAR evenness: REAL);

(* **** *)
(* Monitoring *)
(* **** *)
PROCEDURE DisplayPattern(sp: SpatPat;   w: Window; time: REAL);
PROCEDURE DumpPattern  (sp: SpatPat; VAR f: Textfile; time: REAL;
  writeFreq,writeSdFreq,writeEveness: BOOLEAN);
END SpatPattern.

```

STOCHSTAT D

```

DEFINITION MODULE StochStat;
(* **** *)
  Purpose
    Auxiliary module for stochastic simulation. Calculates means, standard
    deviation and confidence intervals of n arrays with m observations of a
    monitorable variable and allows to display the means and the confidence
    intervals in the graph window, using the module SimGraphUtils.

  Programming
    * Design:          T. Nemecek           19.4.90
    * Implementation: T. Nemecek           24.4.90

    Last revision of definition: 24.06.91 tn
(* **** *)
FROM DMFiles        IMPORT TextFile;
FROM DMConversions  IMPORT RealFormat;
FROM SimBase         IMPORT Model;

TYPE
  StatArray;
  Prob2Tail = (prob999, prob990, prob950, prob900, prob800);
    (*2-tailed probability for confidence intervals the values mean promilles*)
  Str31 = ARRAY [0..31] OF CHAR;

VAR
  notExistingStatArray:      StatArray; (* read only *)

(* **** *)
(* StatArray management *)
(* **** *)
PROCEDURE StatArrayExists(statArray: StatArray): BOOLEAN;
PROCEDURE DeclStatArray(VAR statArray: StatArray; length: INTEGER);
  (* declares an array of data with n-length observation per run. Implicitly calls
  ClearStatArray! *)
PROCEDURE RemoveStatArray(VAR statArray: StatArray);
PROCEDURE RemoveAllStatArrays;
PROCEDURE ClearStatArray(statArray: StatArray);
  (* fills all columns of the array of data with 0.0, except the column with the
  independent variables, which is initialized to the undefined value. Resets
  the array to the initial state. *)
PROCEDURE ClearAllStatArrays;
PROCEDURE SetStatArray(statArray: StatArray; N, X, sumY, sumYSquare:
  ARRAY OF REAL);
  (* an initial state of the statArray can be set. Can be used e.g. to continue an
  experiment, which had to be aborted. AUTION: If any of the values are not
  known, set undefVal for the independent, and 0 for all N, sumY and sumYSquare! *)

(* **** *)
(* Data storage *)
(* **** *)
PROCEDURE Set.UndefValue(  undefVal: REAL);

```

```

(* has only an effect, if no array are currently declared for reasons of
   consistency*)
PROCEDURE GetUndefinedValue(VAR undefVal: REAL);
(* undefVal is assigned to any statistical value, which can not be calculated,
   because the number of observations is not sufficient, e.g. means if n=0, or
   stdDevs is n=1. This value is also used to display values in the graph, that
   could not be calculated, e.g. mean if the number of observations is 0. You
   should use an undefVal, that does not occur in your data. The default
   undefVal is -1.0E30; *)
PROCEDURE SetTolerance( tol: REAL);
PROCEDURE GetTolerance(VAR tol: REAL);
(* tol is the maximal tolerance in which values of the independent variable are
   accepted. The value of the independent variable has to lie within the
   interval [x-tol,x+tol], where x is the first value given as independent. The
   default tolerance is 10E-4 *)
PROCEDURE PutValue(statArray: StatArray; index: INTEGER; x, y: REAL); (* adds a
   value y to the stat array *)
PROCEDURE GetValue(statArray: StatArray; index: INTEGER; VAR count, x, sumY,
   sumYSquare: REAL);
(* ****)
(* Statistics      *)
(* ****)
PROCEDURE GetSingleStatistics(statArray: StatArray; index: INTEGER; VAR count, x,
   sumY, sumYSquare, meanY, stdDevsY, confIntsY: REAL; confProb: Prob2Tail);
(* gives statistical values describing a single observation point.
   count = number of observations at any observation point
   x = independent variable
   stdDevsY = standard deviation
   confIntsY = half confidence interval for confProb of any observation point
   in the array. The true mean lies within the interval [mean+confIntervalY,
   mean+confIntervalY] with a probability confProb. The statistics are given as
   follows for any observation point:
   if N = 0 --> at any observation point, sumY,sumYSquare=0, all other
   statistical values are -undefVal
   if N = 1 --> the mean,sumY & sumYSquare are the single value resp. its
   square and all other values are = undefVal
   if N ≥ 2 --> all values are calculated *)
PROCEDURE GetStatistics(statArray: StatArray;
   VAR N, x, sumY, sumYSquare, meanY, stdDevsY, confIntsY: ARRAY OF REAL;
   confProb: Prob2Tail;
   VAR length: INTEGER);
(* gives statistical values describing the data.
   N = number of observations at any observation point
   X = independent variable
   For further explanations see text of PROC GetSingleStatistics *)
(* ****)
(* Graphical display      *)
(* ****)
PROCEDURE DeclDispMV(statArray: StatArray; mDepVar: Model; VAR mvDepVar: REAL;
   minDepVar: Model; VAR mvIndepVar: REAL);
(* Each data array to be displayed in the graph window must be associated with a
   dependent and an independent variable, which should both be declared as MVs
   in the client model. If time should be the independent variable, then
   SimGraphUtils.timeIsIndep can be given as parameter. See SimGraphUtils.DEF
   for description of the monitoring mechanism. *)
PROCEDURE DisplayArray(statArray: StatArray; withErrBars: BOOLEAN;
   confProb: Prob2Tail);
(* The data are displayed in the graph if the following conditions are met:
   1. the associated MV must be set as isY
   2. the associated indepVar must be set as isX, respectively if the
      simulation time is chosen, none of the MVs must be set as isX.
   error bars with probability confProb are displayed, if withErrBars=TRUE and
   all observation points have an N ≥ 2. If no values have been stored at any
   observation point, these values are displayed as undefVal. Make sure that
   undefVal lies outside your scaling range. *)
PROCEDURE DisplayAllArrays(withErrBars: BOOLEAN; confProb: Prob2Tail);
(* The data of all array are displayed. You can select the variables you ant to
   display as isY. *)
(* ****)
(* File output      *)
(* ****)
(* supports the file output of StatArray data together with labels, written on
   the top of the data and the independent variable values, written in the
   leftmost column. The data are written from the current position of the file f,
   which should be open. *)
TYPE
  RealFileFormat = RECORD rf: RealFormat; n, dec: CARDINAL END;

```

```

FileOutFormat =
RECORD
  means, counts, sumsY, sumsYSquare, stdDevsY, confIntsY: BOOLEAN;
  indepsFormat, meansFormat, sumsYFormat, sumsYSquareFormat, stdDevsYFormat,
  confIntsYFormat: RealFileFormat;
  confProb: Prob2Tail;
END;
(* The labels are written with the following suffixes:
  mean           -s
  count          -N'
  sum Y          -ΣY
  standard deviation -stdev
  confidence interval -CIL resp. -CIH for low and high limit *)
VAR (* read only! *)
  meansOnly, (* writes only means *)
  meansSDCI, (* writes means, standard deviations and confidence intervals *)
  allVals: (* writes all stored and calculated values *)
  FileOutFormat;
  (* default RealFormat:
    ff = ScientificNotation;
    n = 10
    dec = 5
    default confProb = prob950*)
PROCEDURE DumpStatArray (VAR f:TextFile;label: Str31;
  statArray: StatArray; fo:FileOutFormat);
PROCEDURE DumpStatArrays(VAR f:TextFile;labels:ARRAY OF Str31;
  statArrays: ARRAY OF StatArray; fo:FileOutFormat; nAr: INTEGER);
(* The independent values of the first StatArray are written in the leftmost
  column. In case the arrays have not the same length, the length of the first
  array determines the number of values written. A character "N" is written in the
  positions where data are missing. Only the first nAr statArrays are dumped to
  the file. *)
END StochStat.

```

VECTORS2D D

```

DEFINITION MODULE Vectors2D;
(* ****
  Purpose
    provides some procedures for calculation with 2D-vectors in cartesian and
    polar coordinates.

  Programming
    * Design:      T. Nemecek      10.2.89
    * Implementation: T. Nemecek      10.2.89

  Last revision of definition: 28.06.91 tn
  ****)
TYPE
  IntVector = RECORD x, y: INTEGER END;
  Vector = RECORD x, y: REAL END;
  PolVector = RECORD angle, length: REAL END;

PROCEDURE TransPolToCartVector(polV: PolVector): Vector;
  (* calculates a vector in cartesian coordinates from a vector in polar
  coordinates. *)
PROCEDURE TransCartToPolVector(vect: Vector): PolVector;
  (* calculates a vector in polar coordinates from a vector in cartesian
  coordinates. *)
PROCEDURE TransCartToIntVector(vect: Vector): IntVector;
  (* calculates a vector in integer coordinates from a vector in real
  coordinates. *)
PROCEDURE TransIntToCartVector(intVect: IntVector): Vector;
  (* calculates a vector in real coordinates from a vector in integer
  coordinates. *)
PROCEDURE AddVectors(vect1, vect2: Vector): Vector;
  (* adds 2 vectors *)
PROCEDURE AddPolVectors(polVect1, polVect2: PolVector): PolVector;
  (* adds 2 vectors in polar coordinates. The call of this procedure implies a
  transformation of the polar vectors into vectors in cartesian coordinates,
  the call of AddVectors and a transformation of the result vector back into
  polar coordinates. If the result vector is used later in cartesian form, it
  is recommended to transform the vectors into cartesian form and to use
  AddVectors for efficiency reasons.*)
PROCEDURE MeanAngle(angles: ARRAY OF REAL; n: INTEGER): REAL;
  (* calculates the mean angle of the first n elements of an array of angles
  [rad]. Reference: Zar, J.R.(1974): Biostatistical analysis, Prentice-Hall,
  Engle Woods Cliffs, N.J., pp 313-314*)
END Vectors2D.

```

CURRICULUM VITAE

I was born in Ostrava (Czech Republic) on 3rd April 1962. In 1970 my family emigrated to Switzerland. I attended the elementary school in Obersiggenthal (AG). In 1979 I was naturalized in Switzerland. I went to the college in Baden (AG), where I obtained my Matura type C (mathematics and natural sciences) in 1982. Thereafter I started my studies in agriculture at the Swiss Federal Institute of Technology in Zürich, where I specialized in plant protection and tropical agriculture. In 1987 I carried out the final diploma work in a Swiss rice pest research project in Madagascar under the direction of Prof. V. Delucchi. From 1988 to 1992 I worked as Ph. D. student and assistant in the Systems Ecology, Institute of Terrestrial Ecology, Department of Environmental Sciences at the Swiss Federal Institute of Technology in Zürich, directed by Dr. A. Fischlin. In 1989 I married Marianne Bertani. In 1990 our son David was born, then in 1991 our daughter Tabea. Since 1992 I work at the Federal Agricultural Research Station of Changins on a project with the aim of making the results of the modelling and simulation work of this thesis practically applicable for forecasting of virus diseases in the seed potato production.

ACKNOWLEDGEMENTS

I owe thanks to Prof. Dr. V. Delucchi, who accepted me as Ph. D. student and thoroughly reviewed the manuscript and to Prof. Dr. W. Nentwig, who kindly accepted to be co-examiner.

I would like to express my warm thanks to Dr. A. Fischlin, who supervised this thesis and gave me much freedom to realize the case study. He introduced me into the systemanalytic approach and into the art of programming; further, as the main author, he provided the simulation software. In numerous discussions he helped to shape this thesis.

I owe deep thanks to Dr. J. Derron, who helped me to find solutions during many fruitful discussions. He supported the project in many ways and provided most data for the simulation studies. He is also the main author of the soil water balance submodel.

I am thankful to Dr. O. Roth to whom I owe a lot of ideas. During many discussions, he patiently helped me to carry out the project and to learn programming. He spent a lot of time working on the plant growth submodel, which was first implemented by Dr. M. Ulrich.

Dr. P. Gugerli shall be thanked for his suggestions and for kindly providing me many experimental results. The Systems Ecology, especially H. Bugmann and D. Gyalistras are thanked for numerous interesting and fruitful discussions. Dr. J. Casas shared with me his experiences in observing and analyzing insect behaviour.

I thank the Swiss National Science Foundation for providing the grants for the project and the Swiss Federal Institute of Technology for supporting the project. The Federal Agricultural Research Station of Changins and its employees helped me by giving me the opportunity to carry out experiments and provided experimental data.

Finally, I want to express my deep thanks to my wife Marianne, who patiently supported me during this study.