EXECUTIVE SUMMARY
Of Report PM 94-01 Entitled
"A regional simulation model for silverleaf whitefly in the Imperial Valley"

Environmental Monitoring and Pest Management
Department of Pesticide Regulation

PURPOSE:

The Environmental Monitoring and Pest Management Branch, Pest Management Analysis and Planning Program (PMAP), developed a computer simulation model to evaluate various regional strategies for managing the silverleaf whitefly (SLWF) in the Imperial Valley of California. These strategies include changing the acreage planted to various crops and varying planting and harvesting dates,.

BACKGROUND:

The silverleaf whitefly (formerly called strain B of the sweetpotato whitefly) was first detected in California in the Imperial Valley in 1989. Since then, the number of SLWF has increased dramatically causing extensive damage to many crops. This insect has been difficult to control because of its resistance to many pesticides, its ability to attack many different plants, its high rate of population growth, and its readiness to disperse from field to field.

Because of these characteristics, the adequate management of this pest may require regional strategies, such as coordinated planting and harvesting dates of different crops. Regional strategies, however, are difficult to investigate experimentally, especially when there are many possible spatial and temporal arrangements of many different crops. In this situation, simulation models can be useful because they allow the exploration of hundreds or even thousands of different scenarios involving large regions.

STUDY METHODS:

Computer models are special computer programs that usually generate numbers that represent certain quantities such as the number of insects on different crops over a period of time. These models are designed to imitate the dynamics of real systems. All models simplify reality in order to facilitate understanding or to predict particular aspects of interest to an investigator. They are not meant to replace the study of real systems. However, they are useful in suggesting the best management strategies among almost endless possibilities.

The SLWF model is a computer program which represents the spatial and temporal arrangement of 19 crops and the number of adult whiteflies on these crops. In the model, the Imperial Valley is subdivided into one-square-mile sections and the model keeps track of which crops are grown in each section and the number of whiteflies on each crop in each section. Whitefly population dynamics are modelled using equations that predict the number of whiteflies each week from the rate of whitefly reproduction and the rate of whitefly dispersal from section to section. The model also represents environmental conditions such as temperature and wind and the effect of these on whitefly growth and migration.

To produce reliable results, the model must be based on sufficient empirical data. Unfortunately, because SLWF is such a recent pest, there is little data for some important parameters. To address this problem, the model was run for different values of the unknown parameters. If the results of the simulation did not change, then accurate values of the parameter were probably not important. If the results were sensitive to different values, then the model served to identify parameters that need additional research. Also, despite the lack of data, the valley's small size and isolation make it a manageable system to model.

The model was run varying the parameters for the general rate of whitefly population growth; the rates of whitefly growth on alfalfa, cauliflower, cotton, lettuce, and melon; whitefly survival from alfalfa harvest; rate of whitefly dispersal; and for initial whitefly densities. Management strategies tested included varying the number of acres planted to alfalfa, cotton, cauliflower, spring melons, and fall melons; and varying the planting or harvesting dates of alfalfa, cotton, and fall melons. The number of whiteflies on each crop generated by the model was compared to help in the evaluation of the different regional management strategies.

For alfalfa, which is harvested repeatedly during the season and not planted annually, equivalent harvest dates for the model were based on "dry down" dates when irrigation is interrupted and above ground parts of the alfalfa die back. It is assumed that whiteflies can not feed on the alfalfa during the dry-down period. Equivalent planting dates correspond to dates when irrigation is resumed.

RESULTS:

Effect of whitefly parameter values:

Higher rates of whitefly growth in the model produced substantially larger simulated whitefly populations. Different values for following parameters did not have a large effect on the whitefly population: the rate of growth on all individual crops except alfalfa, whitefly survival from alfalfa harvest, the dispersal parameter, and the initial number of whiteflies. The growth rate of whiteflies on alfalfa did cause large changes in the whitefly population.

Effect of number of acres planted:

Alfalfa had the largest effect on whitefly populations. Planting no alfalfa reduced the regional whitefly population by about 67%. Not planting spring melons reduced the population by about 57%, fall melons by 47%, cotton by 46%, and cauliflower by 35%. As greater acres of each crop were planted, the number of whiteflies increased approximately linearly.

Effect of dates planted and harvested:

A simulated alfalfa dry down from 23 July to 8 October reduced the whitefly population by 58%, nearly as large as having no alfalfa. Periods of dry down shorter than 6 weeks were not nearly as effective in reducing whitefly populations and longer periods of dry down did not further suppress the whitefly population.

Different simulated cotton planting dates had no consistent effect on whitefly densities, though both earlier and later plantings tended to generate fewer whiteflies. Earlier cotton harvest dates did result in generally lower number of whiteflies. For example, harvesting cotton four weeks earlier reduced whitefly populations by 19%. Later planting dates for fall melons had a substantial effect in lowering whitefly populations. For example, planting fall melons four weeks later reduced whitefly populations by 33%.

In none of the management scenarios discussed so far was the total regional whitefly population reduced more than 67%. However, by simultaneously harvesting cotton early, planting fall melons later, and drying alfalfa during the summer the whiteflies were reduced by 90%.

CONCLUSIONS:

The qualitative results of the model were consistent with what is known about the population dynamics of SLWF in the Imperial Valley. That is, both in the field and in the model, whitefly populations remain low during the early part of the year and then start to increase in the spring, especially on spring melons. When spring melons are harvested, there is a drop in the regional population of whiteflies, but many migrate to cotton where the populations grow exponentially. At cotton harvest there is again a slight decrease in the regional whitefly population but many migrate to fall melons and cole crops and continue their exponential growth.

In addition, the model was quite robust to changes in most whitefly parameter values. That is, the results of the simulations did not change much when different values were used for the parameters that did not involve management scenarios. Robust models are considered more reliable than non-robust models. The primary exception was that changes in the values used for population growth rate of whiteflies on alfalfa resulted in large differences in whitefly densities. Thus, one conclusion from this modelling is that research on the growth of whiteflies on alfalfa should be a high priority.

The results of various simulations support the use of alfalfa summer dry-down as an important strategy in a regional pest management system especially when used in conjunction with other practices such as early harvest of cotton and late planting of fall melons.

The results from the model need to be interpreted with care, however, because of the lack of empirical data, especially the population growth rate on alfalfa, and because it has not been extensively validated. In addition, as with all models, this regional simulation model is a simplification of reality. It may be that some of the characteristics of the system that were not included could have important effects on the results. Presently, the model should be used primarily to help users gain a better understanding of the some of the possible interactions between various factors and to suggest fruitful avenues for further research.

A regional simulation model for silverleaf whitefly in the Imperial Valley

by

Larry Wilhoit[1], Steve Schoenig[2], David Supkoff[1], and Bruce Johnson[3]

[1]Department of Pesticide Regulation, Pest Management Analysis and Planning Program,

1020 N St., Sacramento CA 95814

[2]Department of Food and Agriculture, Biological Control Program,

3288 Meadowview Road, Sacramento CA 95832

[3]Department of Pesticide Regulation, Environmental Hazards Assessment Program,

1020 N St., Sacramento CA 95814

January 1994

Pest Management Analysis and Planning Program

Abstract

The silverleaf whitefly (formerly called strain B of the sweetpotato whitefly, *Bemisia tabaci*) has become one of the most devastating pests in the Imperial Valley of California in the last few years. It attacks hundreds of different plants, has a high rate of growth, has few effective natural enemies, and readily migrates between fields. These traits not only make it a serious pest but also make it difficult to control on a field by field basis. An effective management program for this whitefly must be on a regional scale considering regional spatial and temporal patterns of all its hosts. Management strategies that have been either implemented or proposed include eliminating fall melon, harvesting cotton early, suspending irrigation of alfalfa for a period during the summer (dry down), and reducing the interval between alfalfa cuttings. However, there are many other potential strategies involving planting and harvest dates of each of the crops and their spatial arrangements.

Unfortunately, determining the most effective regional pest management program experimentally is an extremely difficult task because there are an almost unlimited number of possible planting and harvesting dates and spatial arrangements for all the different whitefly hosts. In such a situation, computer simulation models can be very useful because they allow one to explore hundreds or even thousands of different scenarios. A simulated year can be run in a matter of seconds and any of the system characteristics can be varied at will. A model can not, however, replace field investigation because there are too many system characteristics to include in a model. A model is a simplification of reality which includes only the most important characteristics for a particular purpose.

The silverleaf whitefly model described in this report was developed to gain insights into the regional pest management strategies most likely to be effective in managing whiteflies in the

Imperial Valley. The model includes the planting and harvest dates and the number of acres planted in each square mile section of the Imperial Valley for 19 different crops; the rate of growth of whiteflies on each of the crops; weekly wind direction and speed and the effect of this on the direction and distance adult whiteflies are dispersed; and temperature dependent development of whiteflies. Given particular initial conditions, cropping patterns, temperature, wind data, and parameter values, the model predicts the number of adult whiteflies on each crop in each section for each week during a year. Because data is lacking or of poor quality on some aspects of whitefly biology such as rate of growth on some crops and winter survival, the results of the model should only be interpreted qualitatively especially at the end of the year.

Numerous simulations were run varying parameters for whitefly population growth rate on several different crops; rates of whitefly dispersal; initial whitefly densities; acres planted of alfalfa, cotton, cauliflower, spring and fall melon; different periods and regional proportion of alfalfa dry down; and planting and harvest dates for cotton and fall melon. The qualitative results of the model were not greatly effected by varying values for initial whitefly densities, dispersal factors, and growth rates on most crops. However, growth rate on alfalfa did have a large effect on whitefly dynamics. Of the management strategies tested by the model, the best appeared to be a summer dry down of alfalfa especially when combined with an early cotton harvest and a late fall melon planting. Therefore, it would probably be most productive to focus experimental investigations on the effect of alfalfa on silverleaf whitefly population dynamics.

Acknowledgments

We would like to thank Joseph Ball, Michael Pitcairn, and Charles Pickett of the California Department of Food and Agriculture, Biological Control Program for discussions and information on the ecology of silverleaf whitefly.

We would also like to thank David Ritter, Roger Jacobs, and Michael Sites of the Imperial County Department of Agriculture for providing information on the dynamics of silverleaf whitefly in the Imperial Valley.

We appreciate the help provided by Lyn Hawkins, Department of Pesticide Regulation, Environmental Monitoring and Pest Management, in developing information on cropping patterns for SLWF hosts in the Imperial Valley; Brad Winters, Department of Pesticide Regulation, Environmental Monitoring and Pest Management, for help in programming; and James Hurt, Pest Control Advisor, for providing information on SLWF dynamics.

Disclaimer

The mention of commercial products, their source or use in connection with material reported herein is not to be construed as either an actual or implied endorsement of such product.

Table of Contents

List of figures

List of tables

Introduction

Background

Clouds of whiteflies inundated the Imperial Valley, California, during the fall and winter of 1991-92, destroying crops, causing a loss of $111 million and a loss of about 6000 jobs (Gonzalez et al. 1992). Recently, these whiteflies have been detected in the San Joaquin Valley threatening a much larger agricultural region (Gruenhagen et al. 1993). These insects were originally thought to be a new biotype (called "strain B" in contrast to the previous "strain A") of the sweetpotato whitefly, *Bemisia tabaci* (Gennadius) (Perring et al. 1991), but recent research suggests that they may be a new species, and the insects were given a common name of the "silverleaf whitefly" (SLWF) (Perring et al. 1993a, b; see Campbell et al. 1993 and Bartlett and Gawel 1993 for objections to giving SLWF species status). This report, although primarily concerned with the SLWF, also discusses the sweetpotato whitefly because more is known about it and because, even if they are different species, the two whiteflies are very similar biologically. They differ primarily in their rates of population growth on different plants and in their ability to transmit different plant diseases (Cohen et al. 1992, Perring et al. 1993a).

The Imperial Valley is an irrigated desert region in southern California that lies below sea level. It extends south of the Salton Sea for about 50 miles and is about 40 - 60 miles wide. During the summer temperatures sometimes exceed 125 °F, it almost never rains, and winds are often high. About 70 different crops are grown there and many of the major annual crops are hosts to SLWF.

B. tabaci was first detected in California in the Imperial Valley in 1928 on cotton, and though the whitefly appeared on crops for the next five decades, it only occasionally became a pest (Gill 1992). During this period, outbreaks of *B. tabaci* were usually associated with intensive pesticide applications. However, in 1981 populations reached damaging levels and intermittent problems occurred during the rest of 1980's. In 1989, the new strain or species (SLWF) was first detected in California. There was an explosive increase in whitefly numbers, reaching devastating levels in the fall of 1991.

SLWF attacks over 600 different plants, but is especially damaging to melon, cotton, cole crops, tomato, lettuce, and alfalfa. It causes damage by extracting plant nutrients, resulting in stunted plants and reduced yields; producing a sticky honey dew which contaminates crops and supports the development of sooty mold which is especially damaging to cotton lint; transmitting several important plant diseases such as lettuce infectious yellows, squash leaf curl, and tomato mottle; and causing plant physiological disorders such as tomato irregular ripening, squash silverleaf, and light stalk in broccoli.

SLWF are difficult to control because of their wide host range, high rate of population growth, year long breeding, high dispersal abilities, distribution on the underside of leaves, and resistance to many insecticides (Butler et al. 1986, Henneberry and Butler 1992) and because of low populations of natural enemies in the Imperial Valley (Meyerdirk et al. 1986, Bellows and Arakawa 1988). SLWF are especially difficult to control on a field by field basis when adults descend in huge numbers on fields, overwhelming most control methods (Watson et al. 1992). Because of the many susceptible hosts that are grown in the Imperial Valley and the extensive whitefly migrations from field to field, a successful pest management strategy should be a regional one.

In 1992, changes in the cropping practices in the Imperial Valley appeared to mitigate the problems with whiteflies (UC Riverside 1993), though weather or other factors may have contributed to the reduced whitefly populations. These changes included eliminating the planting of fall melon, reducing cotton acreages by half, eliminating summer irrigation of alfalfa for some period of time (dry down), and moving to a 28 day alfalfa cutting cycle. However, the effort to develop improved management options is hampered by the lack of data on many of the critical aspects of whitefly biology. It is also difficult to test many different strategies on a regional basis since there are so many crops planted at different times.

For such a situation, computer simulation models can be very useful in exploring the possible consequences of various management strategies. A simulation model is a computer program that mimics particular features of the real world. When critical data are lacking as in the whitefly situation, the model can be useful in suggesting strategies and characteristics that may have significant impact on the system's functioning. These features should receive top priority for experimental investigation. As better data are gathered, the model will be more useful in suggesting the best management strategies among the almost endless possibilities. Conducting field experiments on all these possibilities is expensive in time and money and may be impossible at the regional scale but can be done in a matter of minutes on the computer. It should be emphasized that models are not meant to mimic reality perfectly but to aid in the understanding of complex interactions among organisms, their hosts, and the environment. Because of the weakness of some of the data the model is based on, the simulations should not be relied on to provide the definitive solutions, but to suggest the strategies most likely to succeed and thus help set priorities for experimental investigations.

This paper discusses a computer simulation model that was developed for the exploration of regional pest management strategies for whitefly management in the Imperial Valley. Despite the lack of important experimental data, there are some factors that make it reasonable to model this system. The valley's small size and isolation also make it a more manageable system reducing the complexities of external influences. Several models have been developed for *B.*

tabaci (von Arx et al. 1983, Horowitz et al. 1984, Zalom et al. 1985, Baumgärtner and Yano 1990), but none have incorporated regional dynamics and dispersal.

Whitefly biology and model assumptions

Modelling the population dynamics of SLWF requires knowledge of its life history and birth, death, and immigration rates. Because SLWF is such a recent pest, little information is available on its biology. However, in most characteristics it is probably similar to the strain A of *B. tabaci* (Perring et al. 1993a).

Life history. *B. tabaci*, as with most whiteflies, has six life stages: egg, crawler (1st nymphal instar), two sessile nymphal instars, "pupa" (4th nymphal instar), and adult. All immature stages except for the first are attached to the underside of a leaf and do not move. Adults are winged and readily disperse after emerging. All stages feed by piercing the plant surface with their sucking mouth parts and tapping into the plant phloem (Byrne 1991).

B. tabaci develops throughout the year producing many generations. The development time of strain A from egg to adult is 16.25 days at 30°C with lower and upper developmental thresholds of 10 and 32°C (Zalom et al. 1985). Development time of poikilothermic animals, such as whiteflies, depends on temperature. In order to develop a method for predicting developmental events that does not explicitly refer to temperature, time can be expressed in physiological units. The simplest physiological unit of measure is the degree day which is the sum of the number of degrees above a lower temperature threshold and less than an upper temperature threshold times the number of days. Expressed in degree days, the generation time of *B. tabaci* is (16.25 days) X (30° - 10°) = 325 degree days.

Only adult whiteflies are included in the model because only adults disperse and thus are most important for predicting regional populations dynamics.

Fecundity and mortality: population growth. Fecundity and mortality of whiteflies are determined primarily by the quality of their food, their population density, the environmental conditions, and the population of natural enemies. The model assumes that food quality is determined solely by the crop the whiteflies are on, that there is a maximum density of whiteflies that can survive on each crop, that all environmental conditions except for temperature are constant, and that there are no natural enemies. In the Imperial Valley these assumptions are probably not too unrealistic. Crop plants are the primary food source for whiteflies in the Imperial Valley. Crops are often harvested before whiteflies reach their maximum level on that crop so that density effects are often not important. Fertilization and irrigation practices for a given crop are similar and probably lead to uniform crop quality for a given crop within the valley. During the summer, rain and other weather conditions that might effect whiteflies are extremely rare. Though a few natural enemies are present, their impact on whiteflies is small (Meyerdirk et al. 1986, Gill 1992). Because of lack of data on the effects of winter conditions such as rain and cold weather on whitefly survival these are not included in the model. Thus the results of the model during the winter should not be relied on too heavily.

The most important factors determining whitefly abundance are crop species and temperature. The number of eggs produced by a single female will vary between 43 to 253 eggs depending on the crop (Powell and Bellows 1992). The effect of temperature can be introduced by using degree days as the measure of time in the growth equations.

Under these assumptions, although mortality and fecundity may differ for different stages, they will be constant over time on any one crop. It can be proven mathematically that under these assumptions, all life stages will increase exponentially at a rate equal to the rate of increase of the population as a whole (Leslie 1945). This result provides justification for modeling only one life stage of the whitefly.

The adult population rate of growth has been modelled using an exponential equation developed for *B. tabaci* strain A on cotton in the Imperial Valley (Zalom et al. 1985):

$$W_t = W_0 \exp(0.00175 d_t) \qquad [1]$$

where W_t is the adult whitefly population at time t, W_0 is the initial population, and d_t is the number of degree days accumulated between times 0 and t. This equation was developed for the biotype A but by changing the value for the rate of increase, the equation should be applicable to SLWF as well.

To get estimates of the growth rate of the whitefly on other crops, various pest control advisors were asked to estimate the growth rate on the different crops relative to cotton. These values were interpreted to represent the increase of whitefly population on the crop relative to that on cotton after one generation.

The meaning of relative growth rates can be seen more clearly in Fig. 1 which shows the population growth of whiteflies on cotton and on some other crop x assuming that the number of whiteflies is initially the same on both crops. The growth equation on cotton is exp(rt) and the growth equation on crop x is $\exp(f_X rt)$, where r is the intrinsic rate of growth of whiteflies on cotton, $f_X r$ is the rate of growth of whiteflies on crop x, and t is the time in degree days. After one generation, t = G, the whitefly population has increased on cotton by ΔW_C and on crop x by ΔW_X. The relative growth rate estimated by the pest control advisors, g_X, is defined as

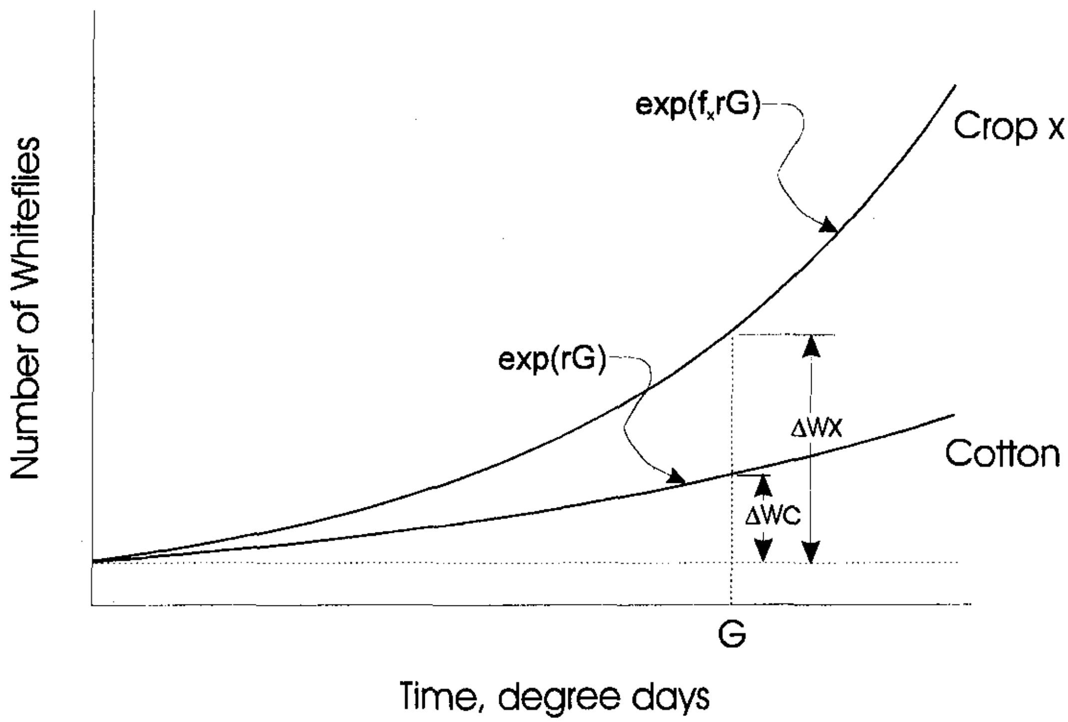


Fig. 1. The number of whiteflies on cotton and on crop x as a function of the number of degree days. After one generation, which occurs in G degree days, the whitefly population has increased on cotton by ΔW_c and on crop x by ΔW_x

$$g_x = \Delta W_x / \Delta W_c. \quad [2]$$

The relative growth rate could have been defined as W_x / W_c, that is, as $\exp(f_x rG)/\exp(rG)$, but this does not fit the intuitive sense of relative growth rate as well as that provided in Eq. [2]. For example, with this meaning if the relative growth rate was 0.5 and $rG < \ln 2$, the whitefly population on crop x would actually be decreasing. With the definition in Eq. [2], the only way to get a negative growth rate is with a negative g_x.

In order to find the growth of whiteflies on crop x, express the unknown parameter f_x in terms of the other known parameters. Eq. [2] can be expressed as

$\Delta W_x = g_x \Delta W_c$ or

$$\exp(f_x rG) - \exp(f_x r0) = g_x\,[\exp(rG) - \exp(r0)] \quad [3]$$

Solving for f_x gives:

$$f_x = \ln\{g_x[\exp(G\,r) - 1] + 1\} / (G\,r) \quad [4],$$

To more easily change the rate of growth of whiteflies on all crops by the same factor for different simulations, another parameter, called the growth factor, g_f, was introduced. This growth factor, g_f, is multiplied by the parameter g_x in Eq. [4]. Bringing these equations together, the model calculates the current whitefly density on crop x, W_t, from the previous week's density on crop x, W_{t-1}, using:

$$W_t = W_{t-1} \exp(f_x' r d), \qquad [5]$$

where,

$$f_x' = \ln\{g_f g_x[\exp(G\,r) - 1] + 1\} / (G\,r) \qquad [6]$$

and d is the number of degree days accumulated during the current week.

This equation was assumed to govern the rate of population growth until whitefly density reached the carrying capacity on its crop at which time the density remained constant at its maximum value.

Dispersal. *B. tabaci* adults exhibit both short-distance and long-distance dispersal (Bellows et al. 1988, Byrne 1991). The dispersal of whiteflies is important in the regional build up of populations since they can move from field to field. Adults usually emerge from lower leaves and fly to upper leaves to feed and oviposit. Whiteflies are poor flyers but long range dispersal does occur when adults fly off their plants, get caught in air currents, and are carried passively by the wind. They may be carried several kilometers.

They will leave a plant when they are in their dispersal phase, they are on an unacceptable host plant, or when their host plant is in an unacceptable condition such as senescence. There are two types of adults, one with a low and another with a high propensity to disperse (Byrne and Houck 1990).

The model assumes that all adults are identical, that they leave a plant only when it is harvested, and that they are dispersed passively by the wind. It is also assumed that dispersal is not affected by temperature. Again, these are simplifications of reality and this must be borne in mind when interpreting the results.

Materials and methods

Whitefly model description

The program was written in the C++ computer language for the MS-DOS® operating system. A description of the programming aspects of the model, along with the full source code, is given in Appendices 1 and 2.

The program simulates the week by week abundance and spatial distribution of adult whiteflies on several crops in the Imperial Valley gridded into one square mile sections. The population size is determined by the number of acres and spatial distribution of each crop, the growth rate of whiteflies on each crop, the average number of degree days during each weekly time step, and adult immigration and emigration between sections.

There are three primary parts of the whitefly model. One part determines which crops are planted and their acreage in each section for each week. Another part simulates the population dynamics and dispersion of whiteflies. The last part displays a graphical representation of the whitefly density in each section for each week.

A flow chart of the program's operation is given in Fig. 2. The program first reads in data from several input files, which are explained more fully in the next section. There are (1) crop files specifying the acreage and planting and harvest dates of each crop in each section, (2) dispersion files specifying the whitefly dispersion factors in each section for each week, (3) a summary crop

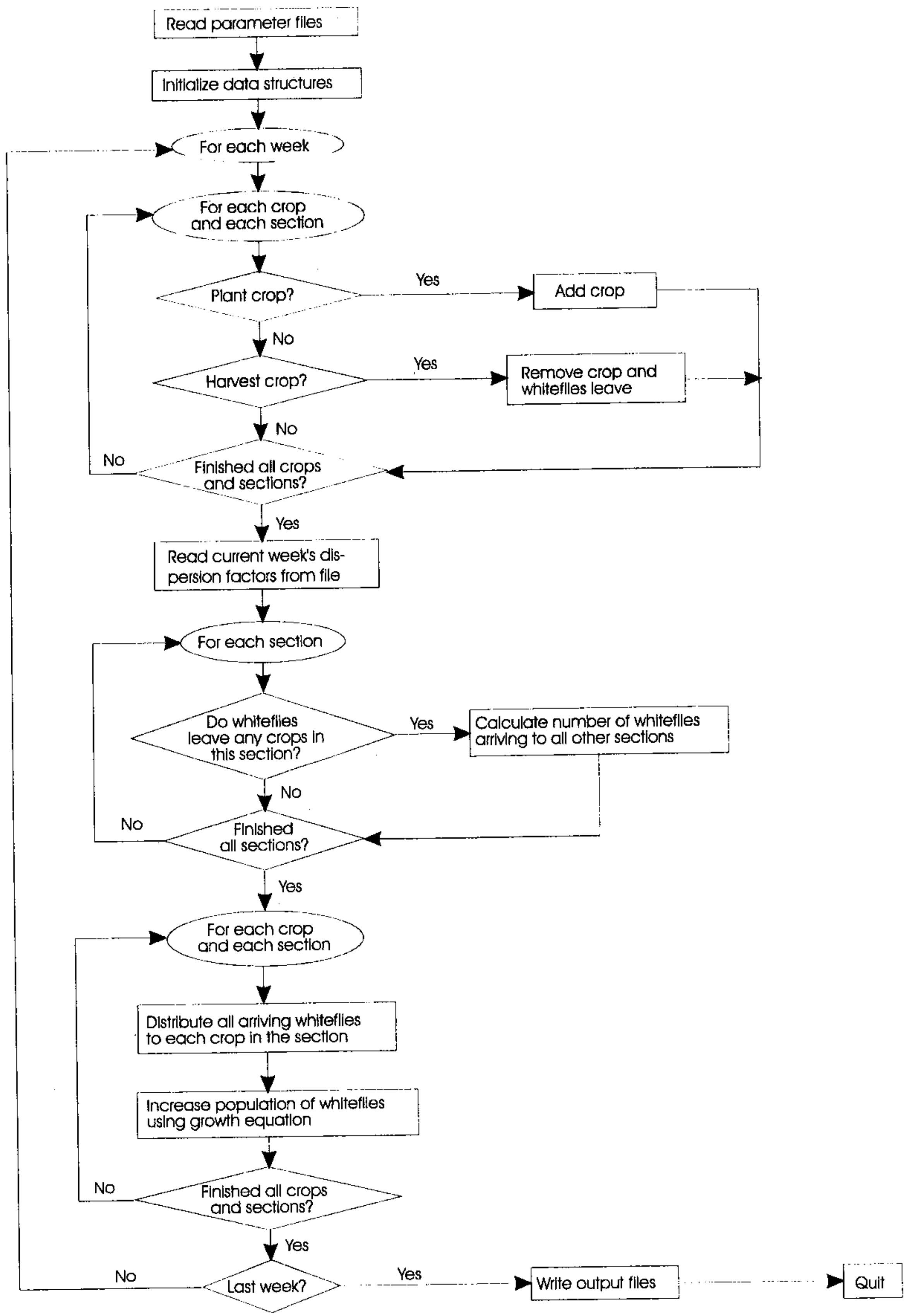


Fig. 2. A flow chart of the operation of the SLWF model.

file, and (4) a whitefly parameter file. The summary crop file lists which crops will be included in the simulation and the acres and planting and harvest dates of each crop. The summary crop file differs from the individual crop files by not including information for each section. The whitefly parameter file lists parameter values for whiteflies, the name of the output files, and the weekly degree days.

Next, the program calculates the number of whiteflies on each crop in each section for each week based on the number from the previous week, the growth rate of whiteflies on each crop, the number leaving each crop, the dispersal distribution calculated from the wind data, and the number arriving to each section.

Finally, the program summarizes the information and reads it into output files and optionally displays it graphically on screen. As an auxiliary to the model a Microsoft ® Excel 4.0 macro is provided that opens the simulation output files for the crops and whitefly densities and displays the results in graphs. The graphs generated show the whitefly densities on each crop and the acres of each crop planted through the entire simulation period. Examples of such output are shown in Figs. 3 and 4.

The program also generates a file giving the number of whiteflies in each section (summed over all crops) for each week. An example of such an output is shown in Fig. 5 which demonstrates the spatial dispersion of whiteflies that are initially present in only three sections

The files provided with the program have default values based on data from various sources discussed in the following sections. These default values for most of the general parameters are given in Table 1. The default values for planting and harvesting dates and acres of each crop are given graphically in Fig. 3.

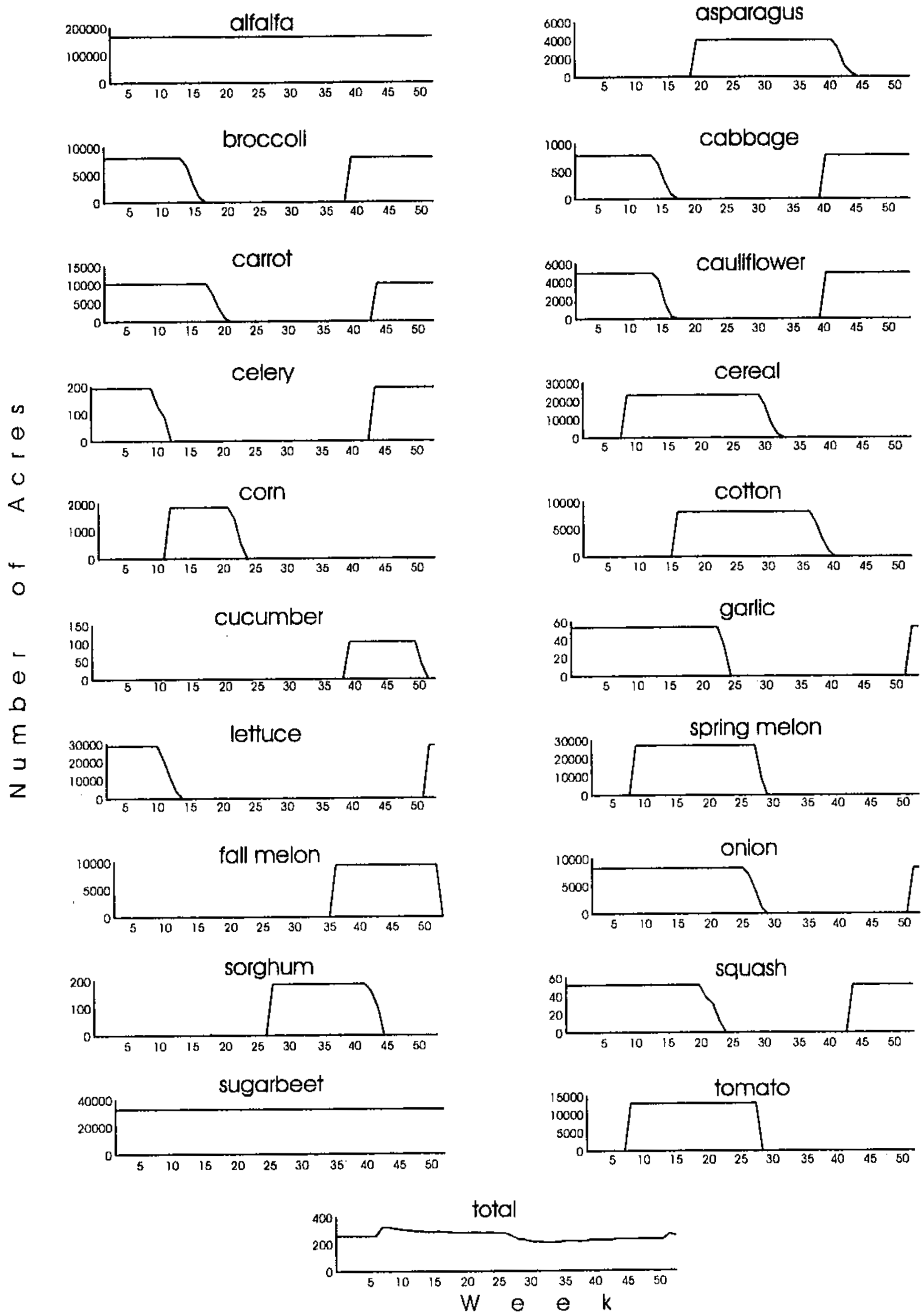


Fig. 3. The default total number of acres planted for each of the crops in the simulation at each week during the simulated year.

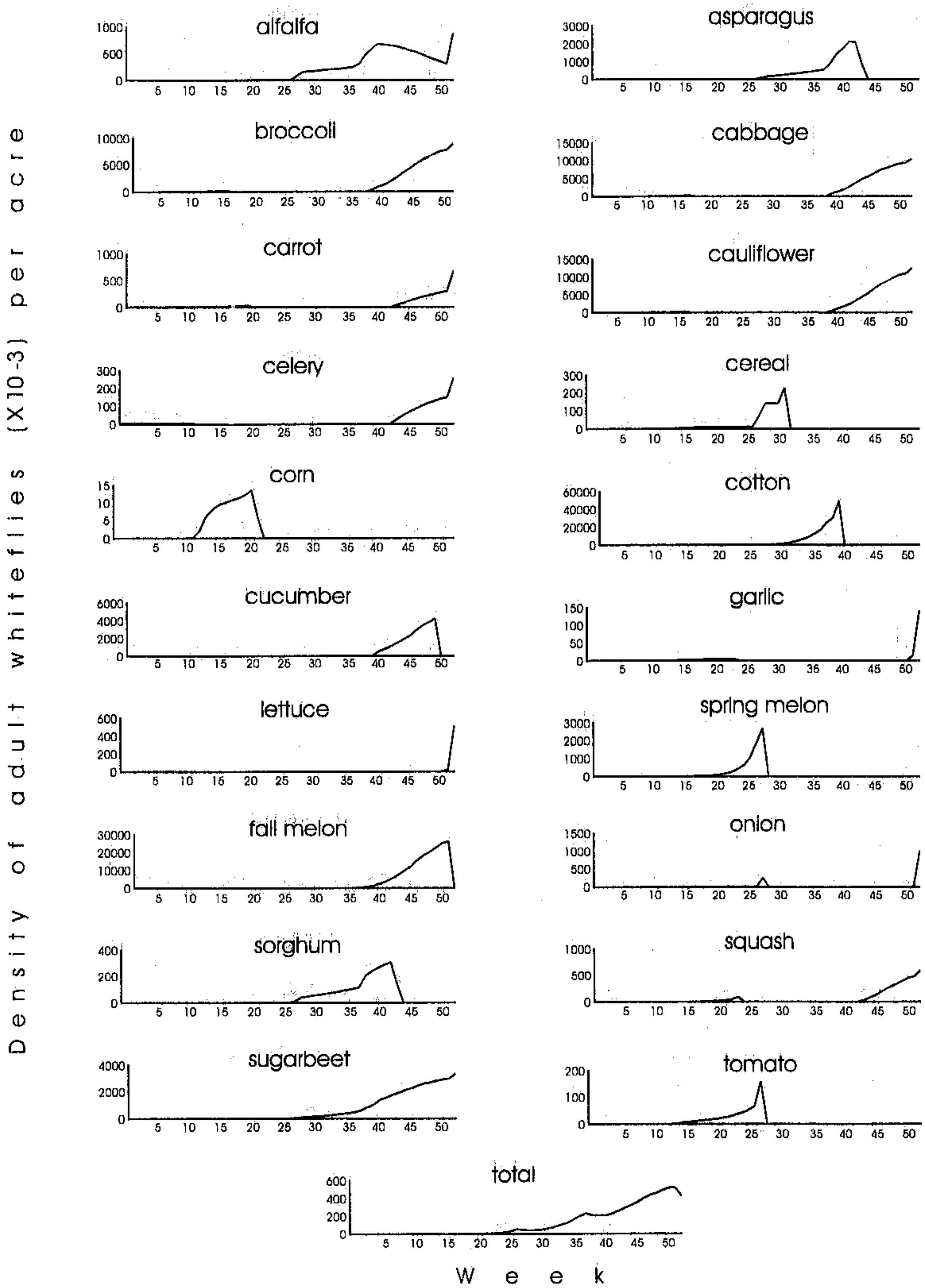


Fig. 4. The density of adult whiteflies ($\times 10^{-3}$ per acre) on each crop at each week from the simulation with the default parameter values given in Table 1. The last plot, labelled "total" gives the total regional population of adult whiteflies ($\times 10^{-9}$ adults).

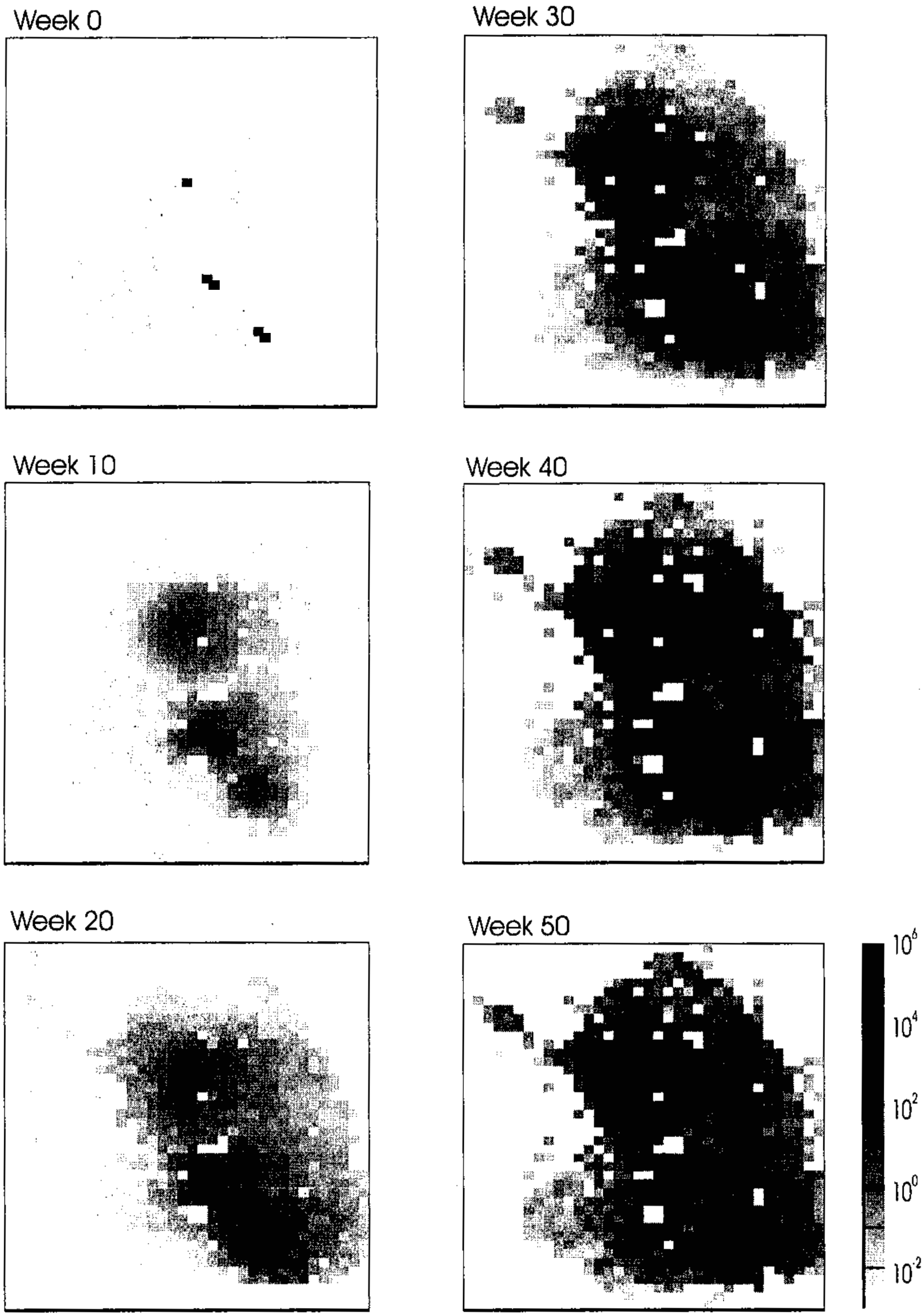


Fig. 5. The number of adult whiteflies ($\times 10^{-3}$) in each square mile section in Imperial Valley as given by the simulation model starting with whiteflies present in only 3 sections. Results are given at 6 different weeks of the year.

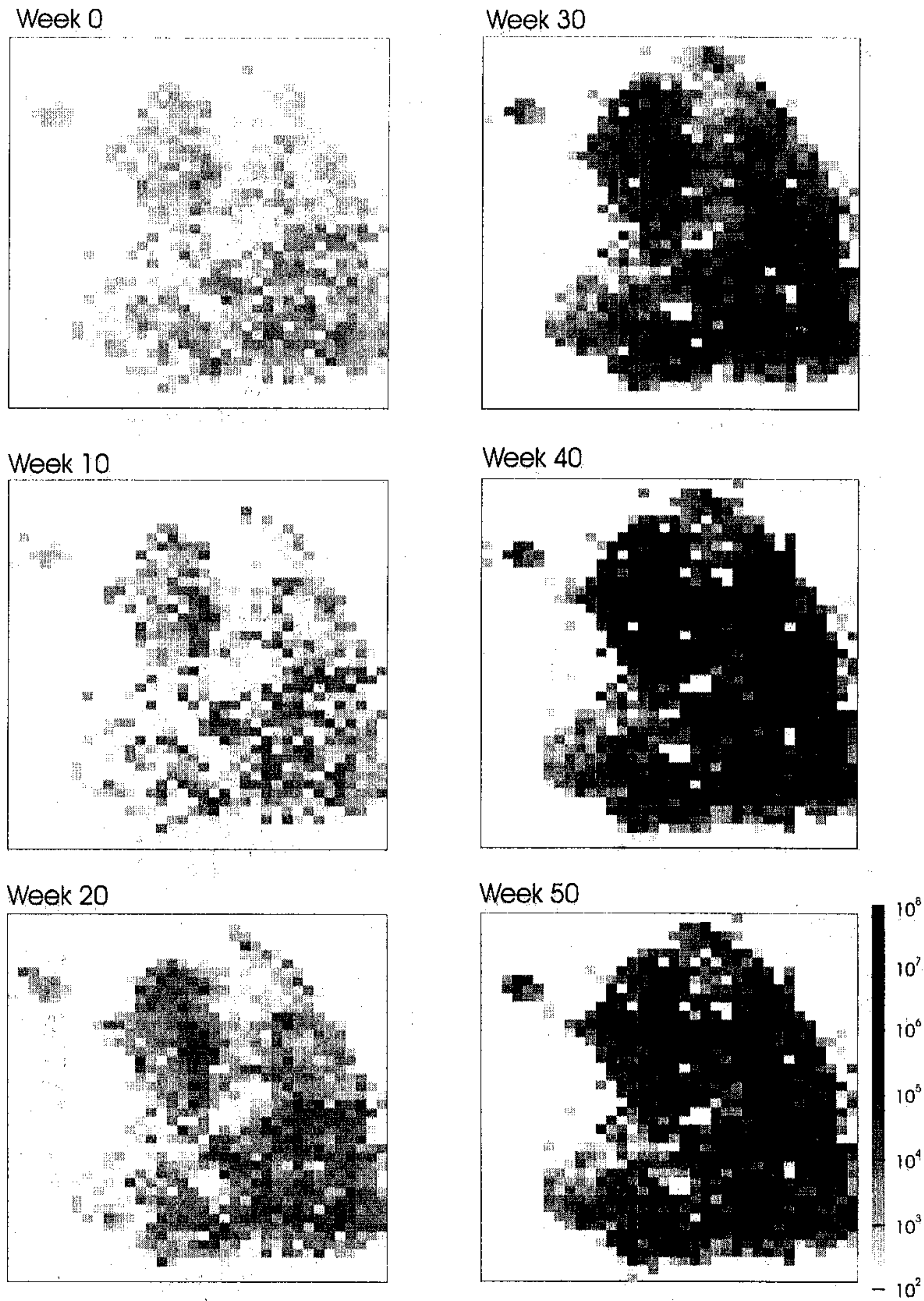


Fig. 6. The number of adult whiteflies ($\times 10^{-3}$) in each square mile section in Imperial Valley as given by the simulation model using the default values given in Table 1. Results are given at 6 different weeks of the year.

Table 1. The default parameter values used for all simulation runs with symbols used in Eq. [5] or in Tables 3 - 25.

Number of weeks simulation run	52
Whitefly population rate of increase, r	0.00175
Rate of growth factor, g_f	2.5
Generation time in degree days, G	325.0
Proportion of disperers leaving a section, D	0.25
Alfalfa weekly harvest proportion, A_h	0.25
Whitefly survival during alfalfa harvest, W_s	0.80
Growth factor on alfalfa, g_{alf}	0.385
Growth factor on asparagus, g_{asp}	0.150
Growth factor on broccoli, g_{bro}	1.080
Growth factor on cabbage, g_{cab}	1.080
Growth factor on carrot, g_{car}	0.200
Growth factor on cauliflower, g_{cau}	1.080
Growth factor on celery, g_{cel}	0.015
Growth factor on cereal, g_{cer}	0.015
Growth factor on corn, g_{cor}	0.015
Growth factor on cotton, g_{cot}	1.000
Growth factor on cucumber, g_{cuc}	1.000
Growth factor on garlic, g_{gar}	0.015
Growth factor on lettuce, g_{let}	0.200
Growth factor on melon, g_{mel}	1.540
Growth factor on onion, g_{oni}	0.015
Growth factor on sorghum, g_{sor}	0.015
Growth factor on squash, g_{squ}	1.000
Growth factor on sugarbeet, g_{sug}	0.200
Growth factor on tomato, g_{tom}	0.390
Adult whitefly carrying capacity on all crops	100 000 000/acre

A fuller explanation of all the input files and the different parts of the model is given in the following sections.

Setting up crop acreage and planting and harvest dates

Because the model simulates the regional population dynamics of whiteflies on different crops in the Imperial Valley, a realistic approximation of cropping patterns for hosts of SLWF in this region was created, based on data from the 1990 Pesticide Use Report (PUR). The spatial arrangement of these crops are represented in a rectangular grid of one square mile sections with 36 rows and 42 columns for a total of 1512 sections. Thus the location of each section is given by its row and column number with section (0,0) at the north west corner.

The model is provided with separate files for each crop. Currently, files are included with data on alfalfa, asparagus, broccoli, cabbage, carrot, cauliflower, celery, cereal, corn, cotton, cucumber, garlic, lettuce, fall and spring melon, onion, sorghum, squash, sugarbeet, and tomato. Within each crop file there is a separate line for each section the crop is grown in. Each line lists the row and column number of a section which has the crop, the acres of the crop planted in that section, the week the crop is planted and the week the crop is harvested in that section, and the initial number of whiteflies present on that crop in that section.

The default acreage in each section for these crops was based on data in the 1990 PUR. The acreage of each crop in each section was found by first sorting the PUR data by crop, section, and field and then selecting the highest number of acres treated for each field and summing up the acreages on all fields in each section. The highest reported acreage for each field was selected to avoid adding the same field several times when there were multiple applications of pesticides to the same field. The data required further adjustment because some fields were given more than one field ID and so were counted more than once, and some fields were not treated and thus their acreages were not recorded. For example, in some cases total acreage within a section exceeded one square mile. The sectional acreage for each crop was adjusted proportionately so that the total crop acreage for each crop equaled that reported in the 1991 Agricultural Crop and Livestock Report for the Imperial Valley. In addition, if the total acreages of all crops in any section exceeded 640 acres (the size of a section), all acreages were scaled so that they summed to 640 acres. Estimates of planting and harvest dates were obtained from data from the Agricultural Extension Service, University of California, Imperial County, Bulletin No. 1075.

Because one of the primary uses of the model is to examine the effects of cropping patterns, it is necessary to have a method for easily changing the spatial and temporal patterns of all the crops simulated. This could be done by changing the values for acres planted and the planting and harvest dates in the crop files. However, because there are 1512 sections that a crop could be planted in, it would be very tedious to change these values in a crop that was planted in many sections. Thus, the program is provided with another file that gives summary values for all crops. This *summary* crop file specifies which crops should be included in the simulation, the names of the *individual* crop files, and whether the values in each individual crop file should be used or not. If the default values in an individual crop file are not to be used, the program uses the values listed in the summary crop file for crop planting dates (first and last weeks of planting), harvest dates (the first and last weeks of harvest), the proportional change in sectional acreages, the proportion of the sections in the crop file that will be included in the simulation, the proportion of the crop that will be harvested in each section, the proportion of the crop that is harvested each week (important for alfalfa), the proportion of the whitefly population that survives a harvest, the maximum whitefly density that the crop can support, and the rate of growth of whiteflies on that crop relative to cotton.

If the default planting dates, harvest dates, or acreage in a crop file are not used, the program will set new values for these parameters based on the values provided in the summary crop file. For example, if the summary file gives values for the first and last date of planting, then the program will not use the values for planting dates for each section in the individual crop files. Rather, the program creates a uniform distribution of new planting dates for each crop by setting the planting date in each section to a randomly chosen date between the first and last planting date in the summary crop file. Random dates were generated using the standard C language random number function.

There are two ways of changing the number of acres planted to a crop. One method changes the number of acres planted in each of the sections listed in a crop file by multiplying these acres by a factor given in the summary crop file. This gives a proportional change in acreage throughout the region with no effect on location of the fields. The other method removes crops completely from different sections that had that crop in the default individual crop files. The proportion of the sections removed is given in the summary crop file. The sections removed are determined by randomly choosing sections from those listed in the crop file, again using the C random number generator.

After the new parameters are specified, the program determines whether the total number of acres of all crops planted in each section exceeds 640 acres, and, if it does, the acreages of each crop are scaled so that the sum of the acreages equals 640.

Whitefly population dynamics

The program reads from the whitefly general parameter file the names of the output files, the number of simulated weeks, the intrinsic rate of increase of whiteflies on cotton, a general growth rate parameter, the generation time in degree days, the proportion of dispersing whiteflies that leave a section, the initial number of whiteflies as a proportion of the values in the crop files, and whether or not to use the values for initial number of whiteflies listed in the crop files. If the initial numbers listed in the crop files are not used, the program continues to read from the general parameter file values for the number of sections that initially have whiteflies, the sections (given by their row and column numbers) that have whiteflies at the start of the simulation, and the number of whiteflies initially in each of those sections. The whitefly parameter file also has a parameter for initializing the pseudo random number generator and values for the number of degree days during each week of the year.

The number of whiteflies in the model is determined by population growth rate and dispersal as explained in the next sections.

Whitefly population growth. Whiteflies are assumed to grow exponentially until they reach the carrying capacity of the crop. After it reaches the carrying capacity, the population is assumed to remain constant. There are no data on carrying capacities so there were estimated by setting the values on all crops to 100 000 000 adults per acre, which is about twice the highest whitefly density found on cotton (Riley and Wolfenbarger 1993, and see Results). Whitefly populations seldom reach the carrying capacity in the field because crops are usually harvested before that occurs.

Table 2. The degree days for each week of 1990 from the IMPACT weather station at Brawley using a lower threshold of 10° C and an upper threshold of 32° C. The degree days were computed by adding each day's degree days

DAY OF YEAR	MONTH	DAY OF MONTH	DEGREE DAYS
7	1	7	23.24
14	1	14	41.36
21	1	21	19.06
28	1	28	32.63
35	2	4	26.75
42	2	11	31.39
49	2	18	24.75
56	2	25	43.77
63	3	4	66.87
70	3	11	41.49
77	3	18	42.32
84	3	25	76.99
91	4	1	59.49
98	4	8	70.71
105	4	15	85.95
112	4	22	74.84
119	4	29	80.84
126	5	6	88.06
133	5	13	86.66
140	5	20	85.90
147	5	27	104.54
154	6	3	90.07
161	6	10	118.45
168	6	17	103.63
175	6	24	121.47
182	7	1	135.69
189	7	8	134.23
196	7	15	142.10
203	7	22	141.74
210	7	29	126.61
217	8	5	133.55
224	8	12	142.43
231	8	19	127.61
238	8	26	115.22
245	9	2	134.34
252	9	9	133.20
259	9	16	135.26
266	9	23	104.68
273	9	30	108.74
280	10	7	108.92
287	10	14	85.14
294	10	21	88.77
301	10	28	78.63
308	11	4	64.94
315	11	11	55.72
322	11	18	67.43
329	11	25	48.96
336	12	2	29.66
343	12	9	33.71
350	12	16	28.78
357	12	23	12.36
364	12	30	12.63

The number of whiteflies on each crop in each section is a function of the number of whiteflies on that crop in that section during the previous week, the rate of growth on that crop, the immigration to the crop, the number of degree days during the week, and Eq. [5]. The equation includes effects of both fertility and natural mortality as observed in the Imperial Valley on cotton.

Degree days were obtained from the IMPACT weather data system at the University of California, Davis, for BRAWLEY.A weather station for 1990. They consist of weekly sums of degree days between 10° C and 32° C (Table 2).

Whitefly dispersal. The model assumes that whiteflies leave a crop only when it is harvested. This is done in the crop setup function where the crop's planting status is determined. If a crop has been harvested during the past week, then it is assumed that all the whiteflies on the crop in that section disperse. Only a proportion of the dispersing whiteflies leave the section they developed in. This proportion is given by a value in the whitefly parameter file (the default value is 0.25, see Table 1). The whiteflies that stay in the section stay on the crops they developed on.

It was assumed that whiteflies are passively distributed by the wind in the same way as any small particle. Thus, use was made of a Gaussian plume function that determined the distribution of particles under various wind conditions downstream from a point source (see Appendix 2). This dispersal function was used to generate a 35 by 35 two dimensional grid of numbers giving the proportional distribution of particles around a central point reflecting average weekly wind conditions for the Imperial Valley in 1990. These values were stored in a set of 52 files for each week of the year. The whitefly model reads in these grid values for the current week from the appropriate file. Whiteflies are dispersed from a section by centering the dispersal grid on that section and multiplying the total number of whiteflies dispersing from that section by each dispersal factor in the grid. This gives the number of whiteflies which disperse from the dispersal section to each surrounding section within the 35 × 35 grid. The grid is then moved to the next section and dispersal is calculated for the next section. The net dispersal into a section is the sum of all incoming dispersals from the other sections.

Finally, the dispersing whiteflies arriving to each section are distributed among all crops in the section by multiplying the total number of arrivals by the proportion of the area of a section occupied by each crop.

Results of whitefly simulations

Simulations were carried out for different parameter values, crop acreages, and planting and harvest dates for a series of simulated years. The weekly whitefly densities on each crop for a simulated year using the default parameter values are shown in Fig. 4, and the spatial distributions of whiteflies at six times during the year are shown in Fig. 6. Subsequent simulations were compared with the results obtained using these default parameter values.

Few if any field studies have provided good estimates of absolute numbers of whiteflies per acre. However, one study (Riley and Wolfenbarger 1993) found a maximum of about 25 SLWF adults per cotton leaf which is consistent with counts of about 200 nymphs per leaf found in other studies. Assuming that 10 leaves per plant have this number of whiteflies and that there are 180 000 cotton plants per acre, this translates to 45 000 000 whitefly adults per acre, which is close to 50 000 000, the maximum density on cotton from the default simulation run. The maximum number of adults per section would then be 640 X 45 000 000 or about 29 000 000 000 which is close to 20 000 000 000, the maximum number per section shown in Fig. 6.

The results of a series of simulations in which parameter values were varied are presented in Tables 3 - 25. These tables present the maximum adult whitefly densities on each crop during the year and the total whitefly population in the entire region (for all crops). The densities are in units of 1000 adults per acre and the region population size are units of 1 000 000 adults. However, to aid in comparing the results, the whitefly densities from each simulation in Fig. 7 - 29 were scaled by the density from the default scenario. That is, the y axis in the figures presents the ratio of the maximum whitefly densities on each crop for each simulation scenario to the maximum density on that crop in the default scenario. In the discussion, these ratios will be referred to as relative whitefly densities. Thus a relative density of 1.0 indicates that the maximum density is the same as that in the default scenario.

One potential problem with the use of maximum values is that no account is made of the period of time whiteflies are present. For example, the total number of whiteflies over the season may be greater than suggested by the maximum value if there are moderate number of whiteflies present for a long period of time. To examine this effect, the total cumulative number of whiteflies over the season on each crop was also calculated for each scenario. However, these values are not presented because they were qualitatively the same as the analysis based on maximum density.

Change in whitefly parameter values

Whitefly population growth rate. The general whitefly population growth rate, parameter g_f in Eq. [6], represents a proportional rate of growth compared to the rate of growth found by Zalom et al. (1985). Thus a value of 2.0 means that the whitefly population is twice as great as that found by Zalom et al. (1985) after one generation. Changing g_f from 1.0 to 3.8 had a large effect

Table 3. Effect of varying whitefly population growth rate on the maximum adult whitefly density ($\times 10^{-3}$) per acre during the year and on the total region wide adult whitefly population ($\times 10^{-9}$). The values in the top row of the table are the population growth rates relative to the growth rate of SLWF on cotton(g_f). Values marked with an * are the default values.

g_f	1.0	1.5	2.0	2.2	2.5*	2.8	3.0	3.2	3.4	3.8
alfalfa	7	43	222	404	889	1624	2131	2750	3583	7056
asparagus	33	190	909	1550	2175	3184	4296	5853	7988	14810
broccoli	112	544	3880	5617	8934	13398	17838	23842	30511	48271
cabbage	101	712	5526	7105	10397	15367	19816	25880	32051	45119
carrot	14	33	197	340	682	1283	1820	2571	3425	5807
cauliflower	138	578	3797	6956	12494	18449	24231	32963	44197	64280
celery	6	10	60	110	257	506	733	1054	1478	2662
cereal	16	41	101	141	228	359	479	631	823	1359
corn	5	8	10	12	14	16	17	19	21	24
cotton	287	2176	12443	23266	50204	58373	63277	66959	72032	86413
cucumber	10	112	851	1768	4268	5680	7029	8742	10759	16287
garlic	2	6	48	101	142	205	253	316	395	587
lettuce	2	20	143	262	506	885	1169	1502	1816	2309
spr melon	103	380	1103	1610	2721	4406	5950	7919	10401	17015
fall melon	98	1223	7283	12508	26315	47063	58282	70266	81701	92837
onion	18	88	583	732	1021	1508	1837	2111	2361	2856
sorghum	6	30	129	213	306	454	606	821	1108	2000
squash	13	28	116	231	600	1265	1992	2651	3452	5667
sugarbeet	38	196	924	1605	3382	6357	9306	13221	17763	28897
tomato	14	34	77	104	160	238	307	393	497	777
total	3	26	159	268	529	922	1238	1650	2137	3317

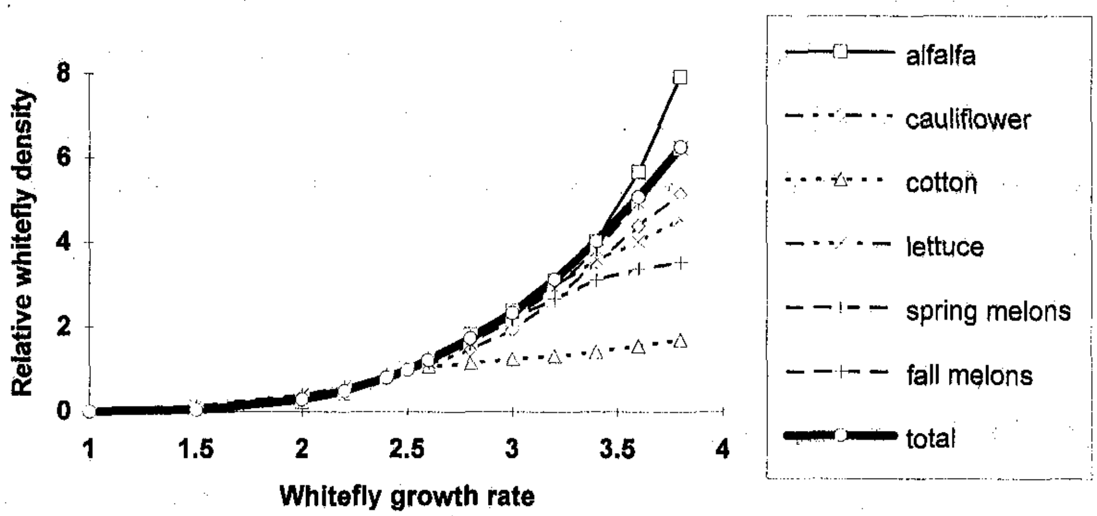


Figure 7. The ratio of the maximum whitefly density in simulation runs with different whitefly growth rates to the maximum density in the default run

on the whitefly population (Table 3, Fig. 7). The total regional population size as well as the density on many crops differed by over 1000 times. The densities on each crop changed proportionally for lower growth rates, but there were large disproportionate differences at the higher growth rates. The reason for this is that the whitefly populations were highest on these cotton and fall melon, and at the highest growth rates the populations reached their carrying capacity on these crops and so could not increase further.

Percentage of dispersing whiteflies that leave their section. When a crop is harvested, all the whiteflies that were on that crop disperse. One parameter measuring dispersal is the proportion of these dispersing whiteflies that leave their origination section. Varying this parameter from 0 to 50% had only a small effect on the total density. Generally more than 50% whiteflies leaving a section resulted in lower densities (Table 4, Fig. 8). Relative densities varied mostly between 0.7 and 1.2. The biggest effect was on cauliflower (and other cole crops). On some crops, such as cucumber, garlic, and sorghum, the effect was opposite to the general trend. For example, in cucumber, the relative density increased from 0.008 to 3.5 as percent dispersing whiteflies leaving a section increased from 0 to 100.

Whitefly survival from weekly alfalfa harvest. Not surprisingly, increasing the proportion of whiteflies that survive alfalfa harvesting increased the density on all crops (Table 5, Fig. 9). However, the effect was not dramatic, even on alfalfa where the relative density increased from 0.34 to 1.68.

Whitefly growth rate on individual crops. Varying the growth rate of whiteflies on a specific crop generally did not cause large effects, except on that crop (Tables 6 - 10, Figs. 10 - 14). The most dramatic effect occurred when the growth rate on alfalfa was greater than 0.5 (Fig. 10). As whitefly growth rate was changed from 0.5 to 2.0 , the relative density on alfalfa increased from

Table 4. Effect of whitefly dispersal on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table are the percentages of whiteflies leaving a crop being harvested that leave the section they were in (D). Values marked with an * are the default values.

D	0	1	10	25*	50	75	95	100
alfalfa	958	965	968	889	747	691	657	642
asparagus	2500	2485	2358	2175	2027	1956	1835	1816
broccoli	10761	10626	9695	8934	8516	8344	7572	7395
cabbage	11402	11346	10925	10397	9880	8793	6428	5926
carrot	566	571	613	682	748	757	733	718
cauliflower	16407	16150	14307	12494	10056	8405	6968	6605
celery	45	55	141	257	390	498	580	592
cereal	238	237	235	228	213	193	174	169
corn	16	16	15	14	11	9	8	8
cotton	46033	46267	48147	50204	49579	48440	47308	46980
cucumber	35	210	1768	4268	8153	11690	14303	14928
garlic	3	9	59	142	283	287	307	319
lettuce	447	454	487	506	526	510	483	467
spring melon	2703	2703	2711	2721	2734	2741	2741	2741
fall melon	25225	25534	26712	26315	24709	23767	22549	21948
onion	1004	1009	1039	1021	890	678	477	434
sorghum	14	26	127	306	574	794	798	794
squash	400	407	477	600	760	878	951	965
sugarbeet	2674	2714	3030	3382	3707	3827	3719	3661
tomato	157	158	159	160	157	151	145	143
total	527	530	536	529	503	481	447	435

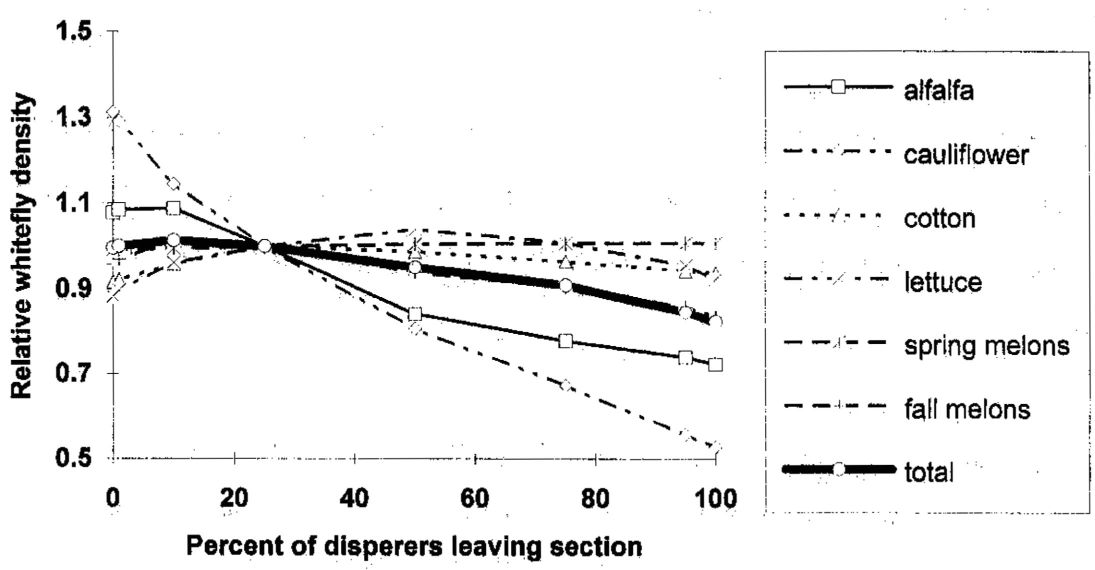


Figure 8. The ratio of the maximum whitefly density in simulation runs with different percentages of dispersing whiteflies that leave a section to the maximum density in the default run.

Table 5. Effect of alfalfa harvest on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values in the top row of the table are the proportion of whiteflies that survive each alfalfa harvest(H_a). Values marked with an * are the default values.

H_a	0.0	0.2	0.4	0.6	0.8*	1.0
alfalfa	306	361	435	537	889	1492
asparagus	1617	1720	1842	1989	2175	2438
broccoli	4165	4924	5892	7161	8933	11761
cabbage	6539	7142	7923	8955	10397	12437
carrot	105	163	258	414	682	1164
cauliflower	5242	6592	7998	9815	12493	16408
celery	23	48	87	150	257	427
cereal	134	151	172	197	228	269
corn	10	11	12	13	14	15
cotton	39870	42493	45476	48157	50204	51805
cucumber	3405	3564	3753	3981	4268	4645
garlic	88	93	101	115	142	194
lettuce	162	206	268	358	506	735
spring melon	2117	2226	2358	2519	2721	2979
fall melon	6731	9002	12472	17858	26309	37813
onion	657	697	760	859	1021	1275
sorghum	137	162	194	238	306	417
squash	52	110	202	351	600	1049
sugarbeet	1614	1868	2212	2688	3381	4462
tomato	135	140	145	152	160	170
total	180	224	287	380	529	762

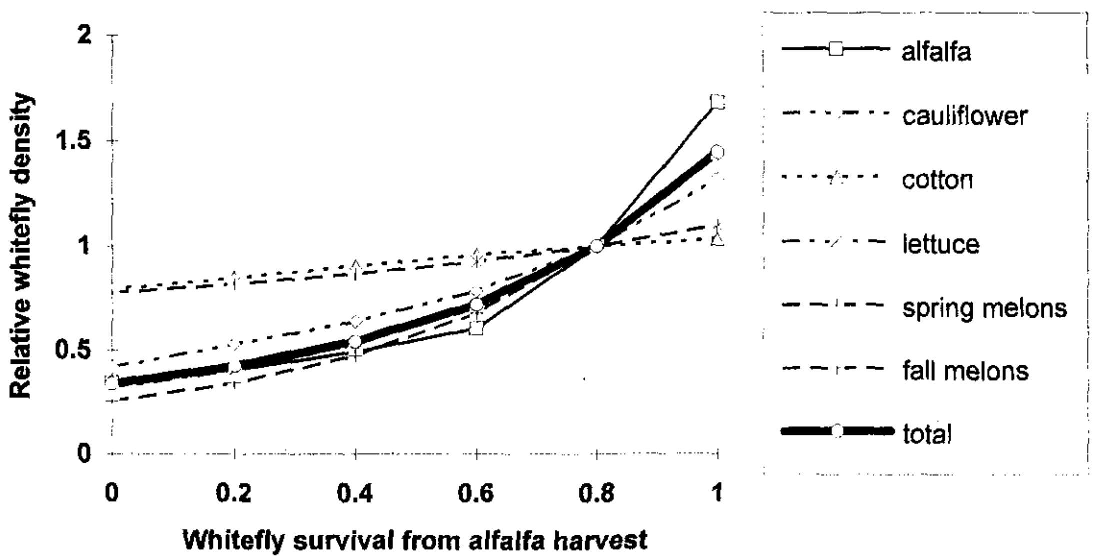


Figure 9. The ratio of the maximum whitefly density in simulation runs with different proportions of whiteflies surviving alfalfa harvest to the maximum density in the default run.

Table 6. Effect of varying whitefly growth rate on alfalfa on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values in the top row of the table are the population growth rates relative to growth of SLWF on cotton (g_{alf}). Values marked with an * are the default values.

g_{alf}	0.0	0.385*	0.5	0.8	1.0	1.5	2.0
alfalfa	252	889	1390	3584	13232	64663	94660
asparagus	1893	2175	2327	3154	5145	17462	40155
broccoli	5803	8933	11306	22207	41269	86226	97728
cabbage	7967	10397	12079	19174	32290	78213	95495
carrot	207	682	1066	2910	8698	44044	64861
cauliflower	7888	12493	15966	32656	60243	90907	97553
celery	67	257	404	1070	3159	22598	45719
cereal	189	228	243	283	340	549	1014
corn	12	14	14	15	16	18	20
cotton	47802	50204	50748	52207	54245	59699	70279
cucumber	3855	4268	4497	5448	7847	28969	66587
garlic	98	142	178	336	863	7042	12580
lettuce	230	506	695	1300	2711	8361	10916
spring melon	2532	2721	2780	2916	3065	3418	3870
fall melon	10315	26309	35685	53877	73664	92903	98019
onion	740	1021	1202	1790	3007	9305	12949
sorghum	197	306	380	706	1553	8832	30651
squash	202	600	869	2226	6803	58087	90076
sugarbeet	2323	3381	4035	6674	13423	46989	73900
tomato	151	160	163	171	181	208	252
total	261	529	718	1488	3647	14110	20552

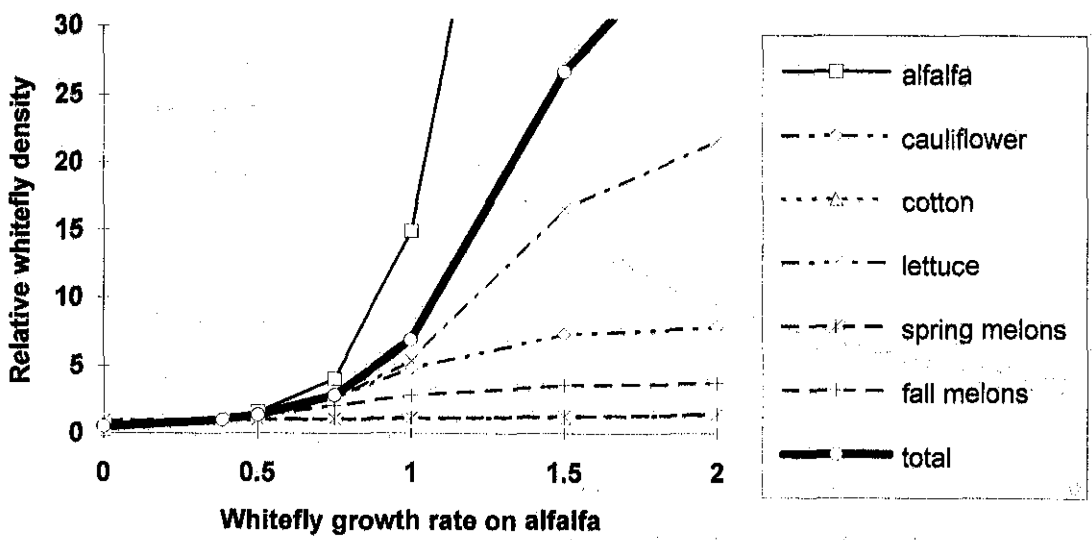


Figure 10. The ratio of the maximum whitefly density in simulation runs with different whitefly growth rates on alfalfa to the maximum density in the default run.

Table 7. Effect of varying whitefly growth rate on cauliflower on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values in the top row of the table are the population growth rates on cauliflower relative to growth of SLWF on cotton (g_{cau}). Values marked with an * are the default values.

g_{cau}	0.0	0.5	1.08*	1.5	2.0
alfalfa	665	749	885	997	1132
asparagus	1973	2053	2178	2293	2546
broccoli	8003	8307	8763	9166	9712
cabbage	9398	9772	10333	10852	11553
carrot	533	594	685	767	876
cauliflower	1869	5454	12241	17921	26768
celery	185	213	251	284	322
cereal	162	187	229	268	322
corn	9	11	15	19	24
cotton	38630	45391	50361	52235	52714
cucumber	4095	4130	4186	4235	4302
garlic	126	131	140	148	158
lettuce	397	438	505	562	634
spring melon	1897	2192	2701	3174	3847
fall melon	20626	22776	26302	29305	32894
onion	915	955	1019	1076	1148
sorghum	240	263	304	342	398
squash	467	505	563	618	690
sugarbeet	2467	2827	3393	3894	4542
tomato	121	135	158	180	209
total	376	432	526	605	715

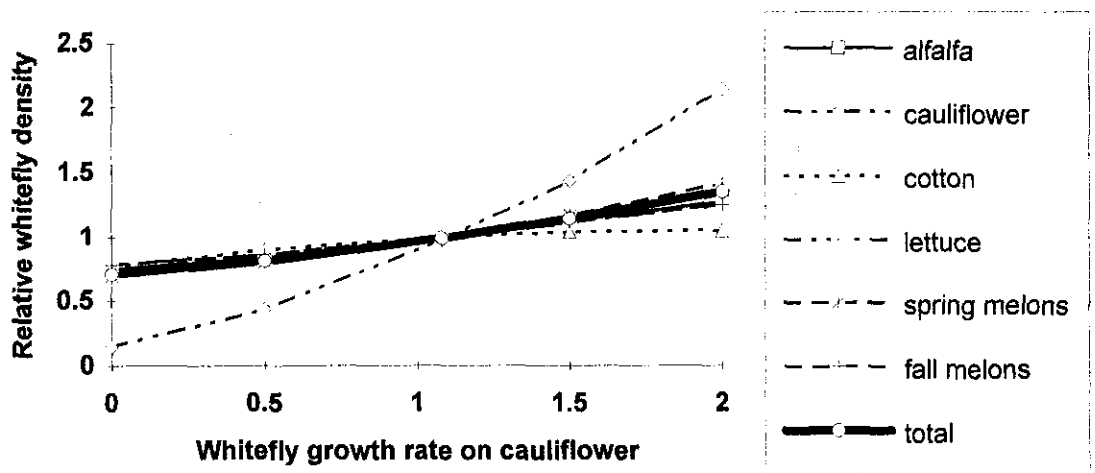


Figure 11. The ratio of the maximum whitefly density in simulation runs with different whitefly growth rates on cauliflower to the maximum density in the default run

Table 8. Effect of varying whitefly growth rate on cotton on the maximum adult whitefly density ($\times 10^{-3}$) per acre during the year and on the total region wide adult whitefly population ($\times 10^{-9}$). The values in the top row of the table are the population growth rates relative to growth of SLWF on cotton (g_{cot}). Values marked with an * are the default values.

g_{cot}	0.0	0.5	1.0*	1.5	2.0
alfalfa	627	688	889	1299	1683
asparagus	1115	1342	2175	2424	2520
broccoli	2775	4589	8934	10368	10861
cabbage	2199	5326	10397	15870	17425
carrot	429	501	682	867	914
cauliflower	3896	5465	12494	13590	13906
celery	135	152	257	577	872
cereal	228	228	228	228	228
corn	14	14	14	14	14
cotton	227	4107	50204	70019	95098
cucumber	272	454	4268	5574	5781
garlic	42	70	142	200	209
lettuce	317	376	506	593	621
spring melon	2721	2721	2721	2721	2721
fall melon	18605	21321	26315	32397	35293
onion	334	712	1021	1378	1492
sorghum	159	190	306	396	490
squash	118	155	600	1522	1650
sugarbeet	1856	2047	3382	6168	8359
tomato	160	160	160	160	160
total	289	345	529	736	863

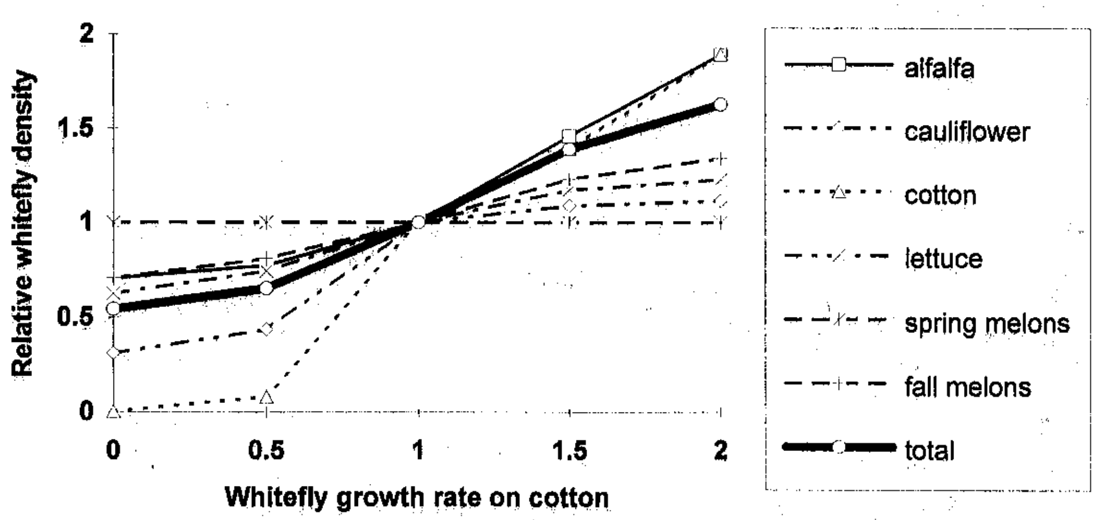


Figure 12. The ratio of the maximum whitefly density in simulation runs with different whitefly growth rates on cotton to the maximum density in the default run.

Table 9. Effect of varying whitefly growth rate on lettuce on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values in the top row of the table are the population growth rates on lettuce relative to growth of SLWF on cotton (g_{let}). Values marked with an * are the default values.

g_{let}	0.0	0.2*	0.5	1.0	1.5
alfalfa	882	889	900	916	932
asparagus	2158	2175	2200	2240	2281
broccoli	8912	8934	8966	9019	9070
cabbage	10341	10397	10480	10617	10753
carrot	675	682	692	709	725
cauliflower	12461	12494	12540	12615	12689
celery	252	257	264	275	285
cereal	226	228	233	240	246
corn	14	14	14	14	14
cotton	50096	50204	50361	50618	50869
cucumber	4257	4268	4283	4308	4333
garlic	141	142	143	145	148
lettuce	492	506	525	552	575
spring melon	2674	2721	2792	2908	3023
fall melon	26046	26315	26713	27342	27896
onion	1017	1021	1027	1037	1046
sorghum	303	306	311	318	326
squash	594	600	609	624	639
sugarbeet	3357	3382	3420	3481	3540
tomato	157	160	164	170	177
total	525	529	535	545	553

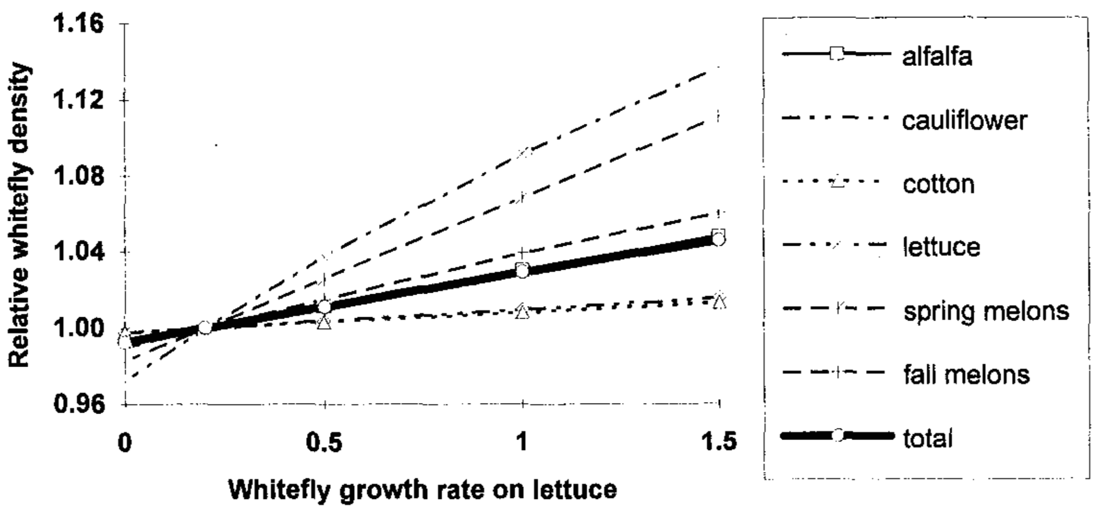


Figure 13. The ratio of the maximum whitefly density in simulation runs with different whitefly growth rates on lettuce to the maximum density in the default run.

Table 10. Effect of varying whitefly growth rate on melon on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values in the top row of the table are the population growth rates on melon relative to growth of SLWF on cotton (g_{mel}). Values marked with an * are the default values.

g_{mel}	0.0	0.5	1.0	1.54*	2.0	2.5
alfalfa	325	349	441	889	1722	2482
asparagus	1231	1291	1521	2175	3379	5826
broccoli	5788	6027	6765	8934	12130	16969
cabbage	7519	7718	8366	10397	13157	17568
carrot	137	192	303	682	1422	2457
cauliflower	7928	8319	9395	12494	16838	23192
celery	72	83	123	257	465	777
cereal	33	45	92	228	471	945
corn	14	14	14	14	14	14
cotton	36513	37431	41259	50204	53470	59288
cucumber	3912	3933	4017	4268	4717	5562
garlic	17	32	87	142	204	282
lettuce	26	67	175	506	1092	1699
spring melon	15	160	782	2721	6308	13468
fall melon	497	2397	7249	26315	56332	83761
onion	69	257	649	1021	1699	2202
sorghum	155	164	200	306	494	859
squash	352	372	433	600	876	1399
sugarbeet	1333	1466	1953	3382	5790	9871
tomato	84	89	108	160	248	415
total	161	187	264	529	950	1461

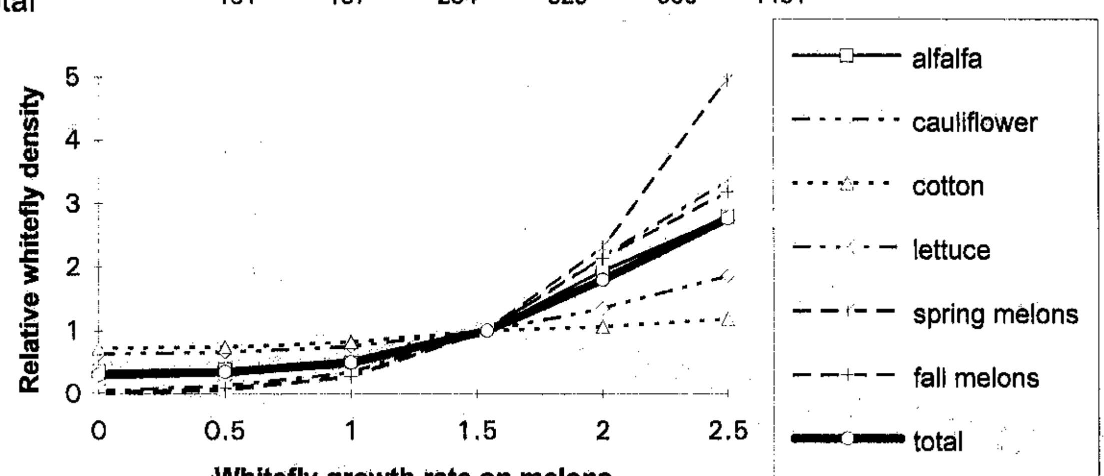


Figure 14. The ratio of the maximum whitefly density in simulation runs with different whitefly growth rates on melon to the maximum density in the default run.

1.6 to 106.4 and relative regional population increased from 1.4 to 38.8. The whitefly density on some of the other crops increased even more than the density on alfalfa when the growth rate on alfalfa was 2.0. For example, the relative density on celery was 178 and on squash 150 (Table 6).

The second most sensitive parameter was the growth rate on melon (Table 10, Fig. 14). Relative density on spring melon increased from 0.005 to 4.9 and fall melon increased from 0.019 to 3.2 when rate of growth on melon increased from 0 to 2.5. Note that densities on a crop with 0 rate of growth is not 0 because of immigration to that crop from other crops.

Change in initial whitefly densities. The relative whitefly density increased almost linearly with increasing initial density per section for low initial densities and then started to level off for higher initial densities (Fig. 15). The effect was greatest on early planted crops such as cereal, corn, spring melon, and tomato (Table 11).

Change in acres planted and harvested

One approach to reducing the area planted to a crop involved removing different number of sections in which the crop was planted in the default scenario. The numbers removed are given as a proportion of the number of sections in the default.

Alfalfa acres planted. Whitefly density generally decreased as less alfalfa was planted especially when less than 75% of the default acreage of alfalfa was planted (Table 12, Fig. 16). When no alfalfa was planted, the relative regional whitefly population was 0.33, but the densities on other crops varied considerably. For example, the relative densities on celery and squash was less than 0.09 and on tomato was 0.85. Relative densities on spring melon, cotton, and cucumber was near

Table 11. Effect of varying initial whitefly density on the maximum adult whitefly density ($\times 10^{-3}$) per acre during the year and on the total region wide adult whitefly population ($\times 10^{-9}$). The values given in the top row of the table are initial whitefly densities relative to the initial densities in the default (W_0). Values marked with an * are the default values.

W_0	0.1	0.5	1.0*	5.0	10.0
alfalfa	135	521	889	2331	3961
asparagus	371	1585	2175	6717	12294
broccoli	2516	6689	8934	18554	26098
cabbage	3623	8062	10397	20052	28080
carrot	122	414	682	2066	3232
cauliflower	2378	8875	12494	25409	38213
celery	30	137	257	860	1414
cereal	23	114	228	1142	2247
corn	1	7	14	68	136
cotton	5586	27922	50204	65557	72895
cucumber	493	2462	4268	6302	8045
garlic	30	110	142	261	340
lettuce	92	331	506	1217	1660
spring melon	272	1361	2721	13551	26060
fall melon	5066	16111	26315	59426	73972
onion	489	806	1021	1862	2592
sorghum	48	212	306	1001	1839
squash	62	307	600	1886	2606
sugarbeet	402	1830	3382	12488	20204
tomato	16	80	160	798	1593
total	96	331	529	1397	2057

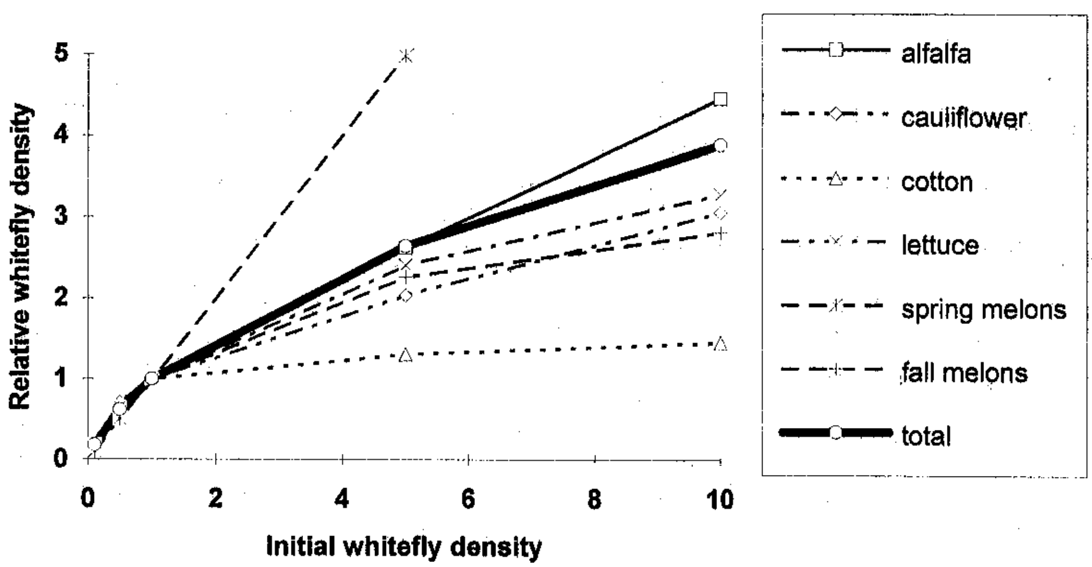


Figure 15. The ratio of the maximum whitefly density in simulation runs with different initial whitefly densities to the maximum density in the default run.

Table 12. Effect of varying alfalfa acreage on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table are the proportion of the sections planted with alfalfa in the default scenario that are used in each simulation run (A_a). Values marked with an * are the default values.

A_a	0.0	0.1	0.3	0.5	0.8	1.0*
alfalfa	0	151	364	526	498	889
asparagus	1617	1626	1627	1672	1790	2175
broccoli	4164	4260	4426	4504	4871	8934
cabbage	6539	6598	6730	6453	7136	10397
carrot	105	118	205	225	316	682
cauliflower	5242	5597	5218	6398	7355	12494
celery	23	24	33	99	105	257
cereal	134	139	140	170	160	228
corn	10	10	11	11	12	14
cotton	39870	40084	37013	42011	47414	50204
cucumber	3405	3418	3415	3430	5252	4268
garlic	88	89	88	72	101	142
lettuce	162	155	163	191	230	506
spring melon	2117	2135	2161	2275	2382	2721
fall melon	6731	6648	7624	8547	10054	26315
onion	657	660	664	529	548	1021
sorghum	137	145	145	168	725	306
squash	52	52	117	518	364	600
sugarbeet	1614	1645	1678	2104	2517	3382
tomato	135	135	135	140	146	160
total	174	176	186	217	259	529

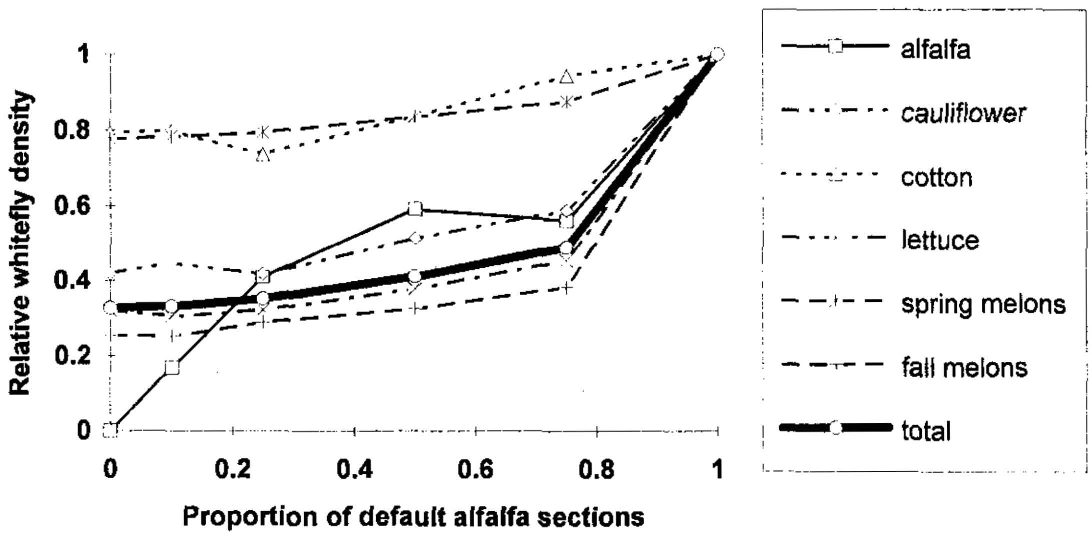


Figure 16. The ratio of the maximum whitefly density in simulation runs with different alfalfa acreages to the maximum density in the default run.

Table 13. Effect of varying weekly alfalfa harvest on the maximum adult whitefly density ($\times 10^{-3}$) per acre during the year and on the total region wide adult whitefly population ($\times 10^{-9}$). The values given in the top row of the table are the percentages of alfalfa harvested each week (A_h). Values marked with an * are the default values.

A_h	0	5	10	20	25*	50	75	100
alfalfa	9652	4688	2593	1158	889	391	239	193
asparagus	1617	1903	2042	2152	2175	2224	2263	2293
broccoli	4165	8976	9939	9380	8933	7721	7533	7607
cabbage	6539	9990	10902	10693	10397	9586	9533	9607
carrot	105	857	969	791	682	332	203	160
cauliflower	5242	12723	14265	13252	12493	10527	10351	10482
celery	23	435	452	315	257	108	61	45
cereal	134	170	192	219	228	257	271	277
corn	10	11	12	13	14	14	14	14
cotton	39870	45423	48347	49987	50204	50799	51137	51343
cucumber	3405	3973	4163	4256	4268	4335	4411	4468
garlic	88	178	185	156	142	107	98	96
lettuce	162	529	606	554	506	316	240	219
spring melon	2117	2371	2527	2683	2721	2791	2828	2866
fall melon	6731	20425	26115	27710	26309	16709	11651	10149
onion	657	983	1075	1058	1021	858	791	776
sorghum	137	235	276	304	306	297	287	280
squash	52	523	648	642	600	368	219	143
sugarbeet	1614	2768	3200	3391	3381	3222	3149	3134
tomato	135	147	154	159	160	160	160	160
total	1744	1127	837	592	529	372	320	307

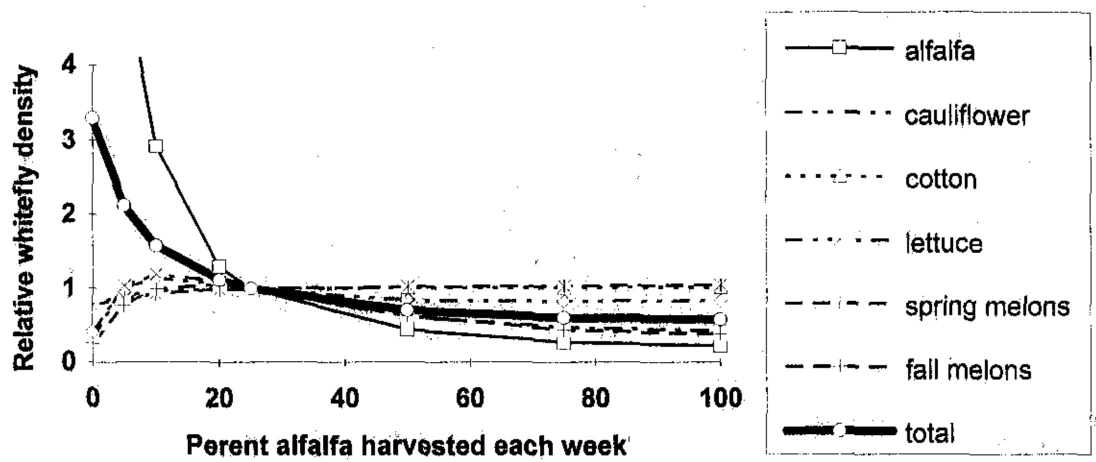


Figure 17. The ratio of the maximum whitefly density in simulation runs with different percentages of weekly alfalfa harvest to the maximum density in the default run.

0.8 and on most of the other crops was less than 0.5.

Percent alfalfa harvested each week. The percent of alfalfa harvested each week had little effect on most crops except on alfalfa (Table 13, Fig. 17). On many crops the density decreased when the harvest percent increased from 10% to 100%. For example, on cole crops relative density decreased from around 1.1 to 0.9, on fall melon from 1.0 to 0.4, on lettuce from 1.2 to 0.4, and on squash from 1.1 to 0.2. On cotton, spring melon, cucumber, and tomato there was almost no change. However, below 10% weekly alfalfa harvest, the whitefly densities were less on most crops. The density on alfalfa was the big exception. When there was no alfalfa harvest the relative density on alfalfa increased to 10.9.

Cotton acres planted. Increasing acreages planted to cotton resulted in an increase in regional whitefly population and an increase on most crops except for spring melon, cereal, corn, and tomato which are planted before cotton (Table 14, Fig. 18). When no cotton was planted the relative density on most crops was around 0.5 to 0.7. On cole crops, however, the relative density was around 0.2.

Cauliflower acres planted. Not planting cauliflower had very little effect on most other crops (Table 15, Fig. 19). The largest effect was on cotton and spring melon where relative densities were both 0.64. Interestingly, there were actually more whiteflies on cauliflower when only 10% of the default number of sections were planted to cauliflower.

Spring melon acres planted. Not planting spring melon had little effect on most all crops (Table 16, Fig. 20). However, it did reduce relative densities on fall melon to 0.28, cereal to 0.14, carrot to 0.35, and alfalfa to 0.36. Again, there were more whiteflies on the crop whose acres were being changed during intermediate number of acres.

Table 14. Effect of varying cotton acreage on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table are the proportion of the sections planted with cotton in the default scenario that are used in each simulation run (C_a). Values marked with an * are the default values.

C_a	0.0	0.1	0.3	0.5	0.8	1.0*
alfalfa	622	625	675	716	776	889
asparagus	1103	1104	1112	1245	1232	2175
broccoli	2688	2724	3779	4388	5013	8934
cabbage	2049	2054	4997	3789	6172	10397
carrot	425	426	450	476	496	682
cauliflower	3805	3859	4246	6406	4918	12494
celery	134	134	134	158	203	257
cereal	228	228	228	229	228	228
corn	14	14	14	14	14	14
cotton	0	2864	22018	27474	31692	50204
cucumber	269	348	496	791	2881	4268
garlic	40	40	41	68	62	142
lettuce	313	314	327	337	352	506
spring melon	2721	2721	2721	2721	2720	2721
fall melon	18436	18538	19404	20539	22226	26315
onion	318	324	360	566	821	1021
sorghum	158	158	207	194	227	306
squash	116	122	132	300	282	600
sugarbeet	1841	1872	2214	2548	3021	3382
tomato	160	160	160	160	160	160
total	285	288	325	364	401	529

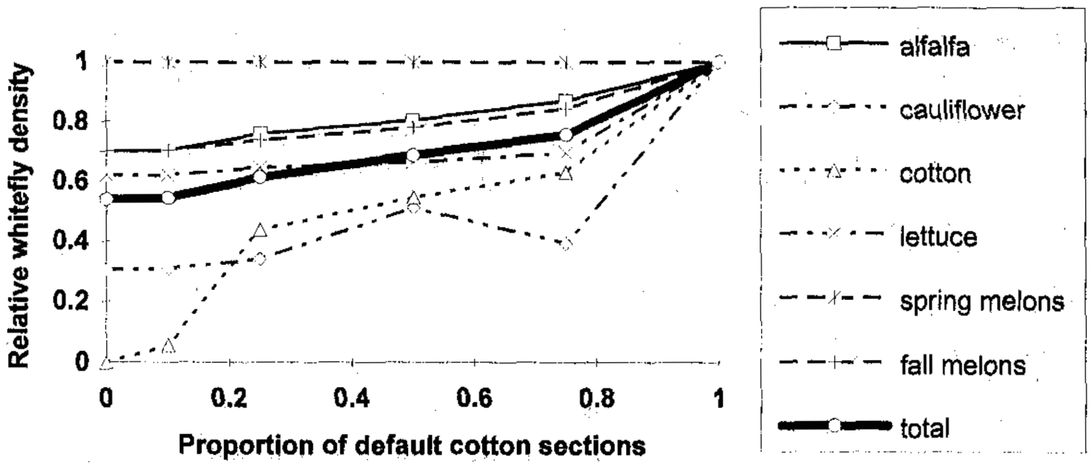


Figure 18. The ratio of the maximum whitefly density in simulation runs with different cotton acreages to the maximum density in the default run.

Table 15. Effect of varying cauliflower acreage on the maximum adult whitefly density ($\times 10^{-3}$) per acre during the year and on the total region wide adult whitefly population ($\times 10^{-9}$). The values given in the top row of the table are the proportion of the sections planted with cauliflower in the default scenario that are used in the simulation (cf_a). Values marked with an * are the default values.

cf_a	0.0	0.1	0.3	0.5	0.8	1.0*
alfalfa	610	626	676	749	793	889
asparagus	1921	1939	1963	2001	2098	2175
broccoli	7670	7726	7849	8330	8756	8934
cabbage	9139	9193	9384	9913	10254	10397
carrot	491	505	549	573	622	682
cauliflower	0	15350	8011	8795	11284	12494
celery	165	169	188	196	206	257
cereal	149	153	162	171	190	228
corn	8	8	8	11	14	14
cotton	32043	33012	33071	34863	41398	50204
cucumber	4071	4071	4073	4195	4230	4268
garlic	122	123	123	128	138	142
lettuce	369	378	409	435	468	506
spring melon	1752	1772	1952	2244	2479	2721
fall melon	19311	19662	21446	23087	24284	26315
onion	888	893	900	953	990	1021
sorghum	227	229	236	265	291	306
squash	445	448	452	506	567	600
sugarbeet	2250	2325	2364	2700	3008	3382
tomato	113	119	121	142	151	160
total	344	357	381	427	473	529

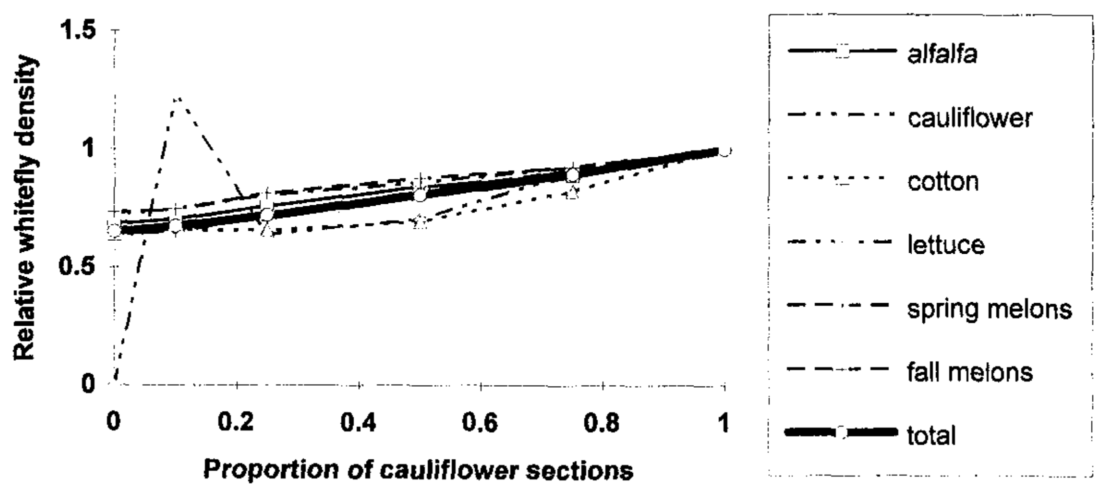


Figure 19. The ratio of the maximum whitefly density for simulation runs with different acreages of cauliflower to the maximum density for the default run.

Table 16. Effect of varying spring melon acreage on the maximum adult whitefly density ($\times 10^{-3}$) per acre during the year and on the total region wide adult whitefly population ($\times 10^{-9}$). The values given in the top row of the table are the proportion of the sections planted with spring melon in the default scenario that are used in the simulation (SM_a). Values marked with an * are the default values.

SM_a	0.0	0.1	0.3	0.5	0.8	1.0*
alfalfa	322	341	381	683	1028	889
asparagus	1225	1240	1456	1620	2340	2175
broccoli	5982	6068	6257	6989	8229	8934
cabbage	7662	7713	7852	8415	9345	10397
carrot	237	262	354	736	1187	682
cauliflower	8133	8247	8601	9327	10955	12494
celery	96	109	161	235	356	257
cereal	32	34	50	258	255	228
corn	14	14	14	14	14	14
cotton	36423	36493	39181	39034	44128	50204
cucumber	3910	3934	4300	4218	4576	4268
garlic	101	102	113	121	152	142
lettuce	180	192	228	378	531	506
spring melon	0	1396	3741	5155	7068	2721
fall melon	7386	8238	10273	22463	31454	26315
onion	694	703	716	865	1026	1021
sorghum	154	160	258	292	217	306
squash	405	411	474	555	633	600
sugarbeet	1421	1477	1734	2593	3262	3382
tomato	84	85	90	123	115	160
total	225	238	269	424	560	529

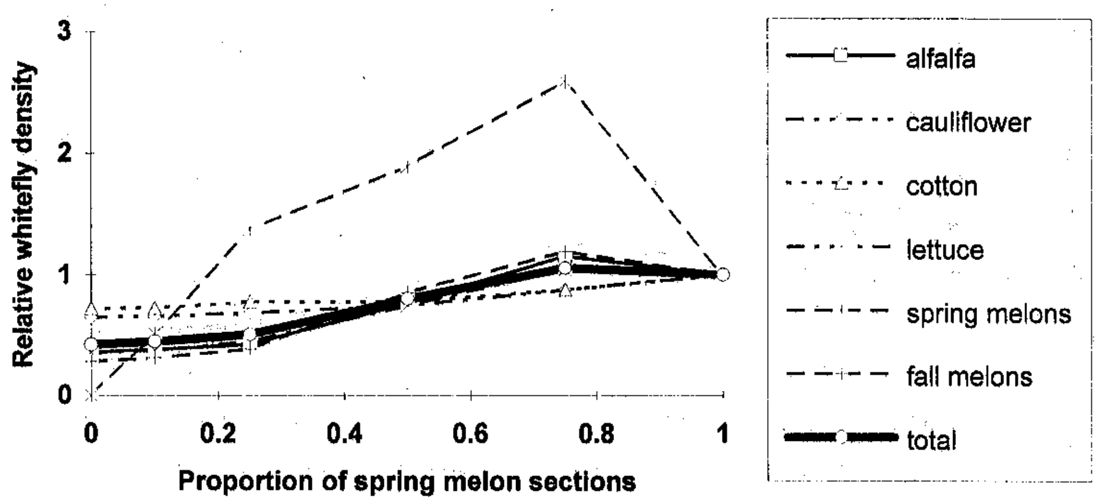


Figure 20. The ratio of the maximum whitefly density for simulation runs with different acreages of spring melon to the maximum density for the default run.

Table 17. Effect of varying fall melon acreage on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table are the proportion of the sections planted with fall melon in the default scenario that are used in the simulation (FM_a). Values marked with an * are the default values.

FM_a	0.0	0.1	0.3	0.5	0.8	1.0*
alfalfa	691	691	691	690	735	889
asparagus	2175	2175	2175	2174	2166	2175
broccoli	8053	8057	8331	8351	8528	8934
cabbage	9653	9656	9716	9773	10068	10397
carrot	314	322	351	452	673	682
cauliflower	11396	11410	11782	12085	11908	12494
celery	157	163	196	191	182	257
cereal	228	228	228	228	228	228
corn	14	14	14	14	14	14
cotton	50204	50204	50199	50202	50198	50204
cucumber	4268	4268	4268	4263	4268	4268
garlic	30	31	30	35	96	142
lettuce	40	47	127	279	560	506
spring melon	2721	2721	2721	2721	2709	2721
fall melon	0	10643	32845	28720	29681	26315
onion	259	259	259	267	840	1021
sorghum	306	306	306	306	306	306
squash	520	545	544	520	1060	600
sugarbeet	3051	3052	3089	3270	3381	3382
tomato	160	160	160	160	160	160
total	281	288	349	418	489	529

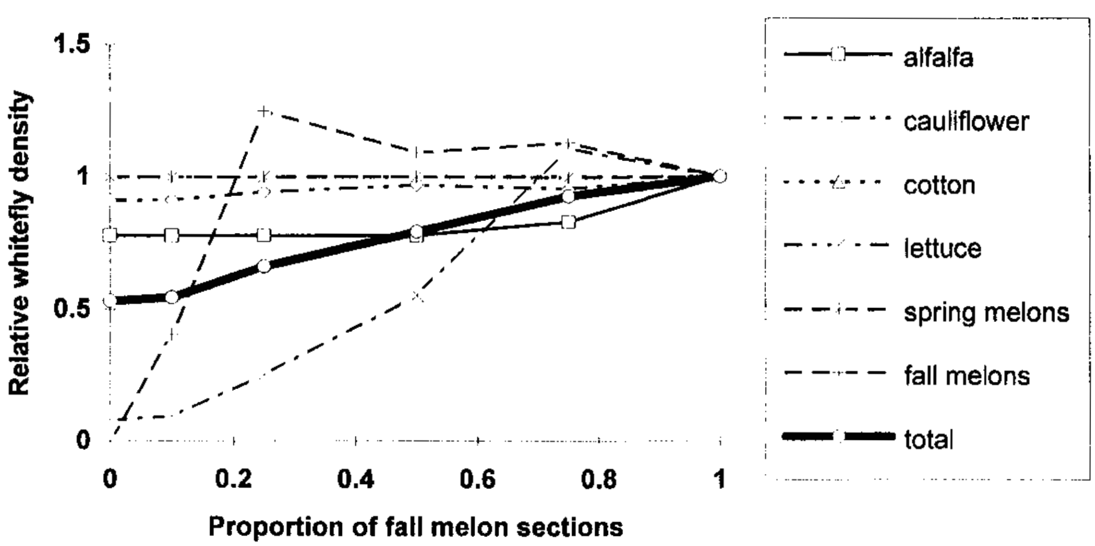


Figure 21. The ratio of the maximum whitefly density for simulation runs with different acreages of fall melon to the maximum density for the default run.

Fall melon acres planted. Not planting fall melon had even less effect on relative whitefly densities on other crops than not planting spring melon (Table 17, Fig. 21). The biggest effects were on lettuce (with relative density equal to 0.08) and garlic and onion (relative densities 0.20). Note, however, that all three of these crops were planted in the winter when the model is probably least accurate.

Change in dates planted and harvested

Alfalfa dry down. Alfalfa is a perennial and so does not have annual planting and harvesting dates. However, it is possible to stop irrigation during some period during the summer causing alfalfa dry down and this can act similarly to a harvest and planting cycle for annual crops. The alfalfa is still present and may still be able to maintain some whitefly populations, but for the simulations it was assumed that no whiteflies survived on dried alfalfa. The process modelled is actually closer to a top kill of the alfalfa. As in the other aspects of the model the reality of dry down are simplified.

The effects of drying down different proportions of the sections with alfalfa were examined (Table 18, Fig. 22). The alfalfa sections to be dried were chosen randomly by the program. A summer dry down period for alfalfa had a large effect on whitefly population. Drying down all sections of alfalfa from 23 July to 8 October, reduced the relative regional population to 0.42. The biggest effects were on celery (relative density 0.12) and squash (0.17). There was no effect on spring melon and tomato because they were planted and harvested before the alfalfa dry down and almost no effect on cotton. The effect of alfalfa dry down was as significant as having no alfalfa at all.

The high densities of whiteflies on alfalfa at 75% and 90% is probably due to the random effects

Table 18. Effect of alfalfa dry down on the maximum adult whitefly density ($\times 10^{-3}$) per acre during the year and on the total region wide adult whitefly population ($\times 10^{-9}$). The values given in the top row of the table are the proportion of the sections planted with alfalfa in the default scenario that are dried down (A_d). Values marked with an * are the default values.

A_d	0.0*	0.2	0.4	0.5	0.8	0.9	1.0
alfalfa	889	756	694	670	847	806	184
asparagus	2175	2114	2100	2000	1987	1964	1947
broccoli	8934	8098	7184	6761	5628	4754	4496
cabbage	10397	9537	8850	8465	8164	7508	6938
carrot	682	559	448	351	242	156	124
cauliflower	12494	11398	10775	10212	8402	6820	6001
celery	257	234	210	151	128	39	31
cereal	228	230	239	244	257	259	275
corn	14	14	14	14	14	14	14
cotton	50204	50207	50207	50217	50228	50235	50244
cucumber	4268	4238	4205	4179	4102	3586	3583
garlic	142	136	125	119	97	93	91
lettuce	506	456	425	368	271	206	189
spring melon	2721	2721	2721	2721	2721	2721	2721
fall melon	26315	23603	21739	18726	14360	9294	7935
onion	1021	969	946	896	827	755	707
sorghum	306	275	229	215	203	199	195
squash	600	363	324	277	252	99	99
sugarbeet	3382	3211	3109	2949	2764	2545	2470
tomato	160	160	160	160	160	160	160
total	529	476	440	394	323	247	220

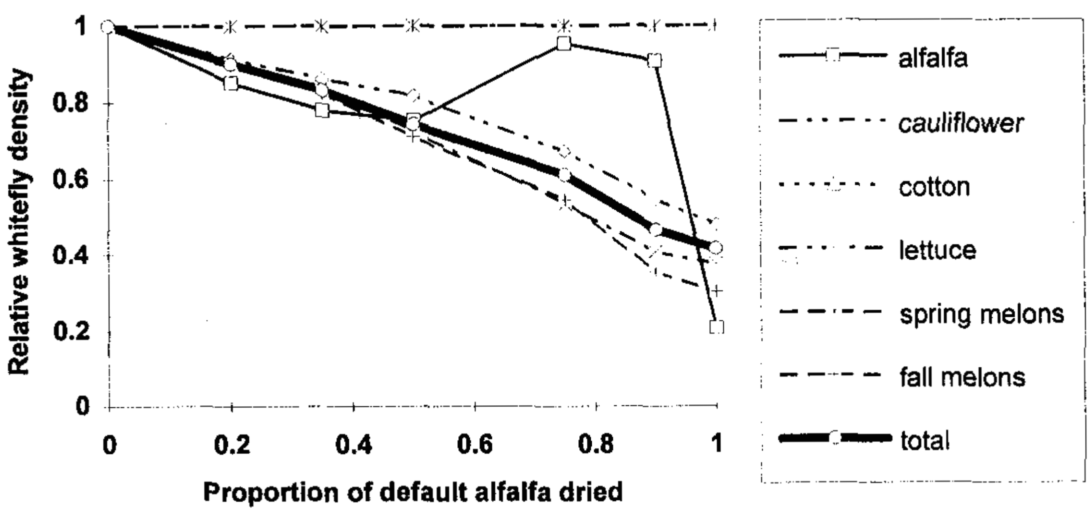


Figure 22. The ratio of the maximum whitefly density for simulation runs with different proportion of alfalfa dry down to the maximum density for the default run.

Table 19. Effect of selecting different sections for alfalfa dry down (but in all cases selecting 80% of the sections) on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table are not parameter values but represent different random seeds that are used to generate different random selections of sections (rs).

rs	1	2	3	4	5	6	7	8
alfalfa	678	727	578	446	429	557	528	581
asparagus	1971	1985	1967	1967	2005	2057	1966	2006
broccoli	5590	5367	5410	4893	4975	4896	4782	5289
cabbage	7137	7195	7086	7299	7799	7488	7583	8194
carrot	208	263	188	201	204	216	220	184
cauliflower	7644	8027	6751	6417	6665	6613	6570	6612
celery	41	101	43	82	39	40	73	87
cereal	268	266	269	270	273	269	262	268
corn	14	14	14	14	14	14	14	14
cotton	50241	50242	50240	50240	50242	50241	50242	50228
cucumber	3643	3605	3642	3621	3591	3606	3644	3597
garlic	107	104	105	94	98	114	97	95
lettuce	252	260	240	223	229	239	252	255
spring melon	2721	2721	2721	2721	2721	2721	2721	2721
fall melon	10233	11785	10379	10213	10665	10600	10845	11167
onion	753	770	753	751	744	734	741	781
sorghum	201	205	204	197	225	205	206	244
squash	114	99	253	99	99	117	103	99
sugarbeet	2625	2641	2641	2601	2635	2578	2602	2648
tomato	160	160	160	160	160	160	160	160
total	272	286	265	256	263	260	264	272

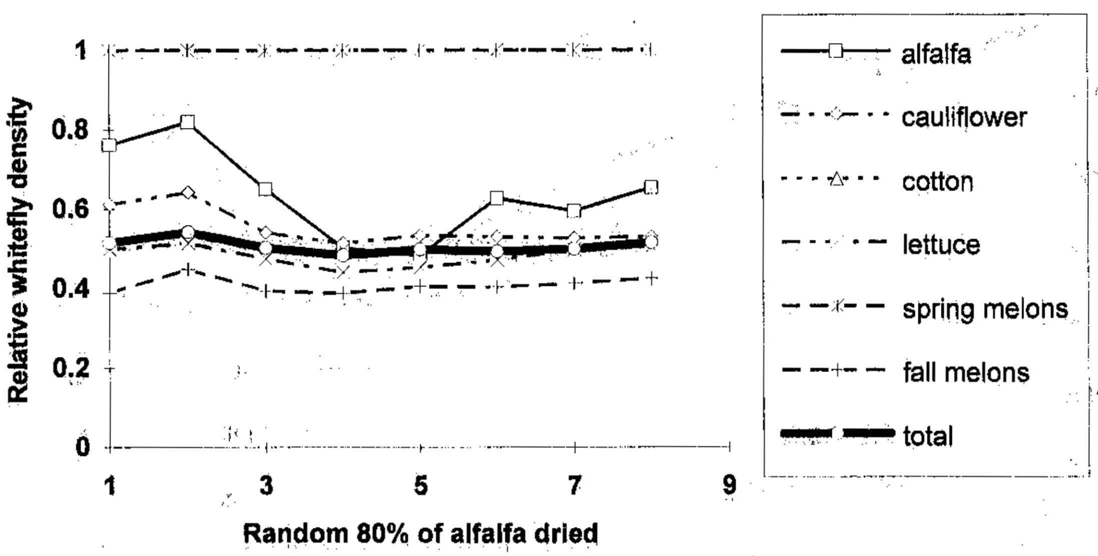


Figure 23. The ratio of the maximum whitefly density for simulation runs with different sections selected for alfalfa dry down to the maximum density for the default run.

Table 20. Effect of length of alfalfa dry down on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table are the number of weeks of alfalfa dry down centered at date Aug 27 (A_{dl}). Values marked with an * are the default values.

A_{dl}	0*	2	4	6	8	10	12	14
alfalfa	889	959	359	211	196	184	174	161
asparagus	2175	2143	2124	2062	1977	1947	1928	1907
broccoli	8934	7370	6350	4733	4577	4496	4489	4481
cabbage	10397	9472	8499	7147	7012	6938	6934	6931
carrot	682	669	265	188	158	124	122	117
cauliflower	12494	10180	8730	6370	6164	6001	5990	5985
celery	257	236	91	64	47	31	31	28
cereal	228	228	228	228	228	275	258	241
corn	14	14	14	14	14	14	14	14
cotton	50204	50218	50219	50221	50234	50244	50266	50289
cucumber	4268	4131	4123	4049	3994	3583	3542	3537
garlic	142	142	108	97	92	91	91	91
lettuce	506	621	215	202	196	189	189	189
spring melon	2721	2721	2721	2721	2721	2721	2721	2721
fall melon	26315	34470	8552	8132	8004	7935	7940	7943
onion	1021	1126	753	728	716	707	707	707
sorghum	306	271	257	221	205	195	190	183
squash	600	507	475	124	99	99	99	99
sugarbeet	3382	3302	2914	2708	2595	2470	2438	2400
tomato	160	160	160	160	160	160	160	160
total	529	546	293	247	231	220	219	217

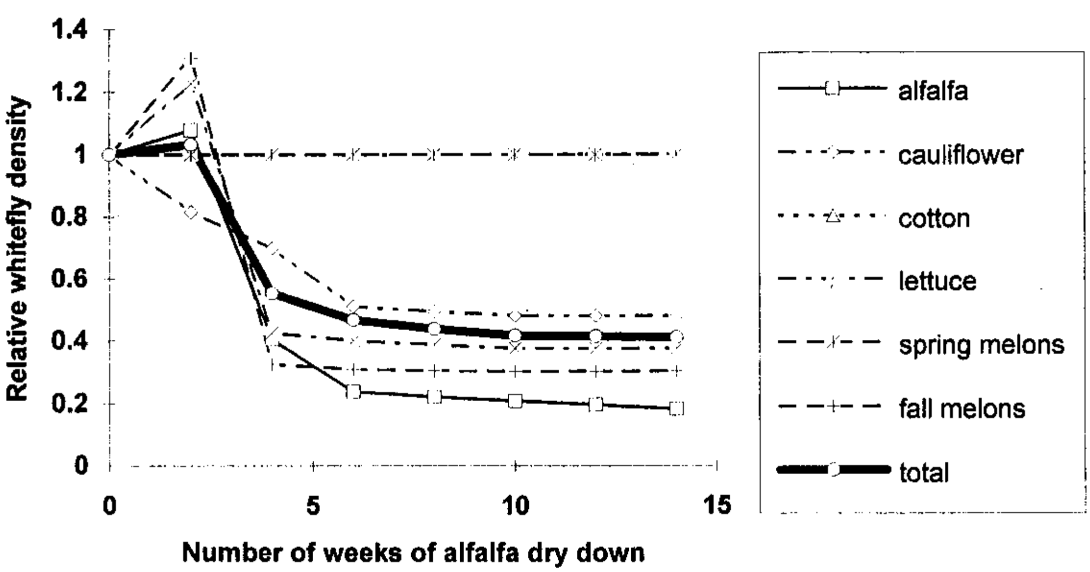


Figure 24. The ratio of the maximum whitefly density for simulation runs with different periods alfalfa dry down to the maximum density for the default run.

of choosing sections for dry down. Randomly choosing different sections for dry down gave variable results especially for densities on alfalfa (Table 19, Fig. 23). Relative densities on alfalfa, for example, varied from 0.48 to 0.82 and on squash densities varied from 0.16 to 0.42, depending on which sections were dried.

The length of the period of dry down also had an important influence on whitefly density (Table 20, Fig. 24). A dry down longer than six weeks did not appreciably reduce whiteflies density compared with densities at a dry down of six weeks. However, periods of dry down shorter than four weeks were not effective. In fact, a dry down of two weeks actually resulted in a greater whitefly population than growing alfalfa with no dry down.

Cotton planting and harvesting dates. To examine the effects of cotton planting time, simulations were run for planting dates from 12 March to 7 May, with the harvest dates correspondingly adjusted so that the time cotton remained in the ground remained constant. The effects of cotton defoliation were not included. No overall pattern emerged (Table 21, Fig. 25). The overall abundance of whiteflies in the region did not vary much (relative population varied from 0.89 to 1.0) with the highest population occurring at the intermediate cotton planting date of 9 April. The biggest effect of cotton planting date in these simulations was on cabbage where relative densities varied from 0.4 to 1.0, with higher densities occurring for intermediate planting dates. Planting cotton earlier did reduce populations on cole crops, squash, and cotton but resulted in higher densities on lettuce and fall melon. However, largest densities occurred on cotton when cotton was planted 26 March.

If cotton was planted at the same time but harvested later (so that the cotton is in the ground for longer periods of time) there was a regional increase of whitefly population and an increase on most crops except for fall melon and asparagus (Table 22, Fig. 26). In this case, peak densities

Table 21. Effect of cotton planting date on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region adult whitefly population (× 10^{-9}). The values given in the top row are week cotton was planted (C_{pd}). The cotton was in the ground for 21 weeks in every run. Values marked with an * are the default values.

C_{pd}	11	12	13	14	15*	16	17	18	19
alfalfa	930	928	946	911	880	840	841	848	855
asparagus	2576	2451	2349	2200	2055	1891	1803	1665	1138
broccoli	5937	5962	6221	8165	10977	12015	11684	9946	8598
cabbage	6261	6443	6622	13814	15653	16409	15049	12246	9854
carrot	704	683	728	704	685	660	673	903	1343
cauliflower	7520	7713	8109	10358	13507	13452	13754	12621	11029
celery	279	365	372	388	295	294	327	343	1480
cereal	227	227	227	228	228	228	228	228	228
corn	14	14	14	14	14	14	14	14	14
cotton	27715	36321	44000	43808	42580	37225	38095	40197	41763
cucumber	437	519	731	781	970	731	847	759	649
garlic	103	146	152	153	144	134	134	122	108
lettuce	617	554	592	532	498	468	463	450	431
spring melon	2715	2715	2721	2721	2721	2721	2721	2721	2721
fall melon	30756	29793	30233	27455	25723	24119	23618	23052	22116
onion	929	1156	1159	1086	1005	932	913	882	750
sorghum	332	356	362	393	324	309	289	247	171
squash	339	500	528	553	586	419	419	617	597
sugarbeet	2933	3198	3406	3416	3330	2963	2972	2987	3040
tomato	160	160	160	160	160	160	160	160	160
total	488	494	512	525	548	531	525	502	485

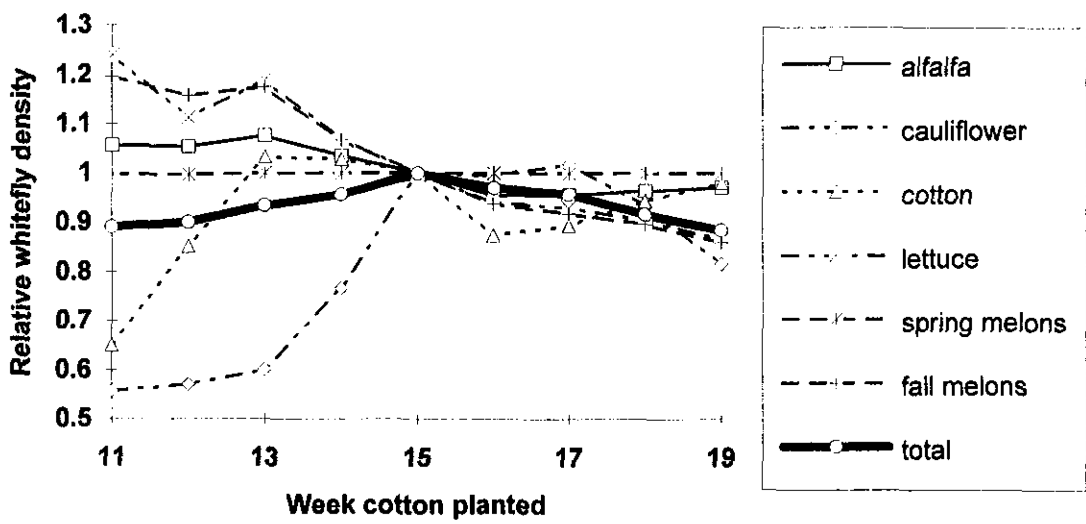


Figure 25. The ratio of the maximum whitefly density for simulation runs with different cotton planting dates to the maximum density for the default run.

Table 22. Effect of cotton harvest date on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row are the weeks cotton was harvested (C_{hd}). Cotton was planted on week 15 in every run. Values marked with an * are the default values.

C_{hd}	32	34	36*	38	40
alfalfa	871	883	880	915	980
asparagus	2075	2219	2055	1867	1178
broccoli	4966	5494	10977	13644	10347
cabbage	5028	5991	15653	17423	12493
carrot	623	663	685	726	1514
cauliflower	6408	7170	13507	16362	12443
celery	201	249	295	444	2873
cereal	228	228	228	228	228
corn	14	14	14	14	14
cotton	16938	29733	42580	51687	62335
cucumber	392	654	970	4402	2156
garlic	87	142	144	146	127
lettuce	581	544	498	479	450
spring melon	2721	2721	2721	2721	2721
fall melon	28965	28465	25723	24427	22767
onion	832	1087	1005	965	820
sorghum	244	276	324	323	184
squash	282	437	586	830	1320
sugarbeet	2565	2961	3330	3693	3899
tomato	160	160	160	160	160
total	442	465	548	598	562

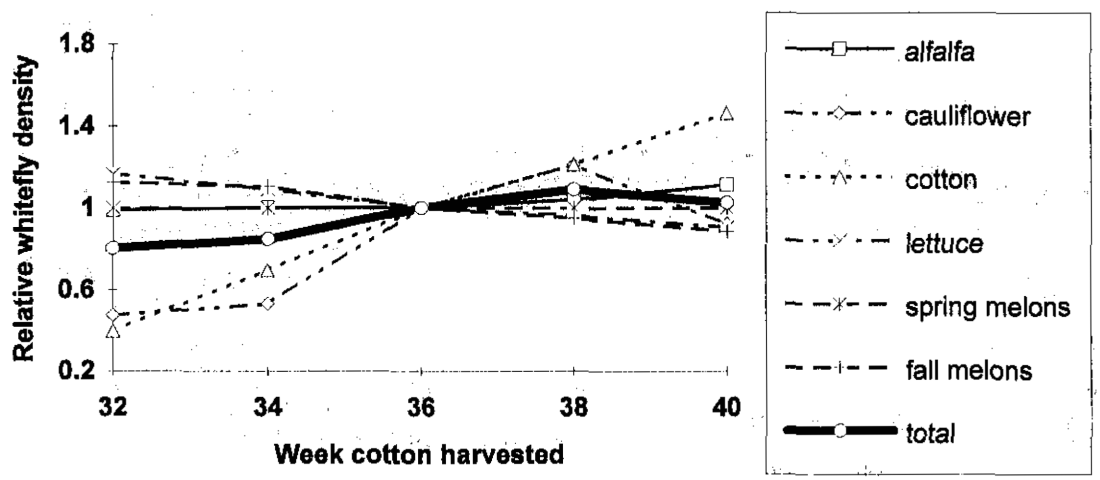


Figure 26. The ratio of the maximum whitefly density for simulation runs with different cotton harvest dates to the maximum density for the default run.

Table 23. Effect of fall melon planted on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table are the weeks fall melon were planted (FM_{pd}). Melon were harvested on week 52. Values marked with an * are the default values.

FM_{pd}	31	33	35*	37	39
alfalfa	1713	1331	889	691	691
asparagus	2175	2175	2175	2175	2175
broccoli	10045	9553	8933	8563	8263
cabbage	11016	10735	10397	10043	9802
carrot	1594	1026	682	524	405
cauliflower	13801	13420	12493	11961	11663
celery	399	333	257	213	184
cereal	228	228	228	228	228
corn	14	14	14	14	14
cotton	50204	50204	50204	50204	50204
cucumber	4268	4268	4268	4268	4268
garlic	188	166	142	130	53
lettuce	1405	829	506	341	191
spring melon	2721	2721	2721	2721	2721
fall melon	66817	44928	26309	15585	7955
onion	1756	1309	1021	872	631
sorghum	306	306	306	306	306
squash	698	621	600	593	564
sugarbeet	3905	3602	3381	3265	3167
tomato	160	160	160	160	160
total	913	706	529	427	355

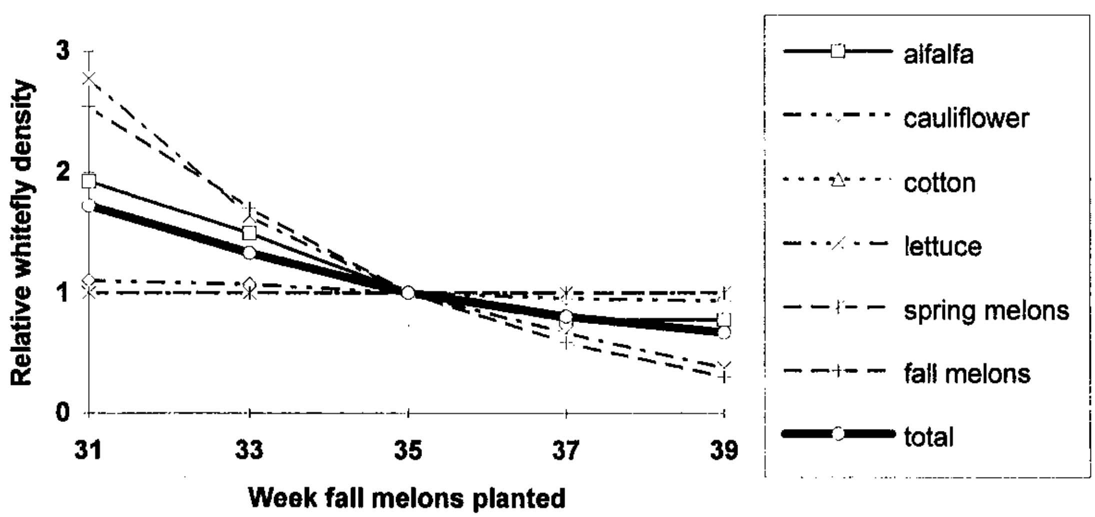


Figure 27. The ratio of the maximum whitefly density for simulation runs with different fall melon planting dates to the maximum density for the default run.

occurred on cole crops at an intermediate harvest date. The largest effect of harvest date occurred on cotton where relative densities increased from 0.4 to 1.5.

Fall melon planting date. Varying fall melon planting date (from 30 July to 24 September) while keeping the same harvesting date had a fairly substantial effect in lowering whitefly densities on nearly all crops, except, of course, for crops planted and harvested before fall melon were planted and for cotton (Table 23, Fig. 27). The biggest effect was on lettuce (relative densities from 2.8 to 0.38) and fall melon (2.5 to 0.30). However, the effect on lettuce occurred during the very end of the year, during which the model is less accurate. Regional population was reduced from 1.7 to 0.67 when the fall melon planting date was increased from 30 July to 24 September.

Cotton and fall melon dates. As cotton was harvested later (from 6 August to 1 October) and simultaneously fall melon was planted earlier (from 24 September to 30 July) resulted in increasing whitefly densities on most crops (Table 24, Fig. 28). For example, during this series of scenarios relative densities on cotton increased from 0.4 to 1.5, lettuce from 0.19 to 2.8, and fall melon from 0.16 to 2.5. However, densities were greatest on cole crops at the intermediate scenario when cotton was harvested on 17 September and fall melon were planted on 13 August.

Alfalfa, cotton, and fall melon dates. Including alfalfa dry down with the early harvested cotton or late planted fall melon or both had very dramatic effects in reducing whitefly densities (Table 25, Fig. 29). Alfalfa dry down combined with early cotton harvest resulted in fewer whiteflies than alfalfa dry down with late planted melon, though, on cole crops whiteflies were much lower. With alfalfa dry down and early cotton, the regional relative whitefly population was 0.24; relative whitefly density on alfalfa was 0.21; on cotton 0.35; on fall melon 0.30; and on cauliflower 0.05. For alfalfa dry down and late fall melon, the regional relative whitefly population was 0.33; relative density on alfalfa was 0.21; on cotton 1.18; on fall melon 0.15; and

Table 24. Effect of cotton harvest and fall melon planting dates on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table refer to different cotton harvest and fall melon planting dates as follows:
1: Cotton harvested week 32, melon planted week 39;
2: Cotton harvested week 34, melon planted week 37;
3: Cotton harvested week 36, melon planted week 35;
4: Cotton harvested week 38, melon planted week 33;
5: Cotton harvested week 40, melon planted week 31.
Values marked with an * are the default values.

	1	2	3*	4	5
alfalfa	451	552	880	1375	1843
asparagus	2075	2219	2055	1867	1178
broccoli	3883	4957	10977	14280	11563
cabbage	4098	5430	15653	17797	13240
carrot	277	473	685	1075	2452
cauliflower	5258	6586	13507	17314	13839
celery	109	197	295	520	3015
cereal	228	228	228	228	228
corn	14	14	14	14	14
cotton	16938	29733	42580	51687	62335
cucumber	392	654	970	4402	2156
garlic	26	64	144	171	206
lettuce	94	317	498	811	1393
spring melon	2721	2721	2721	2721	2721
fall melon	4097	13943	25723	43629	65519
onion	259	485	1005	1261	1723
sorghum	244	276	324	323	184
squash	240	383	586	851	1422
sugarbeet	2283	2759	3330	3921	4455
tomato	160	160	160	160	160
total	206	327	548	780	968

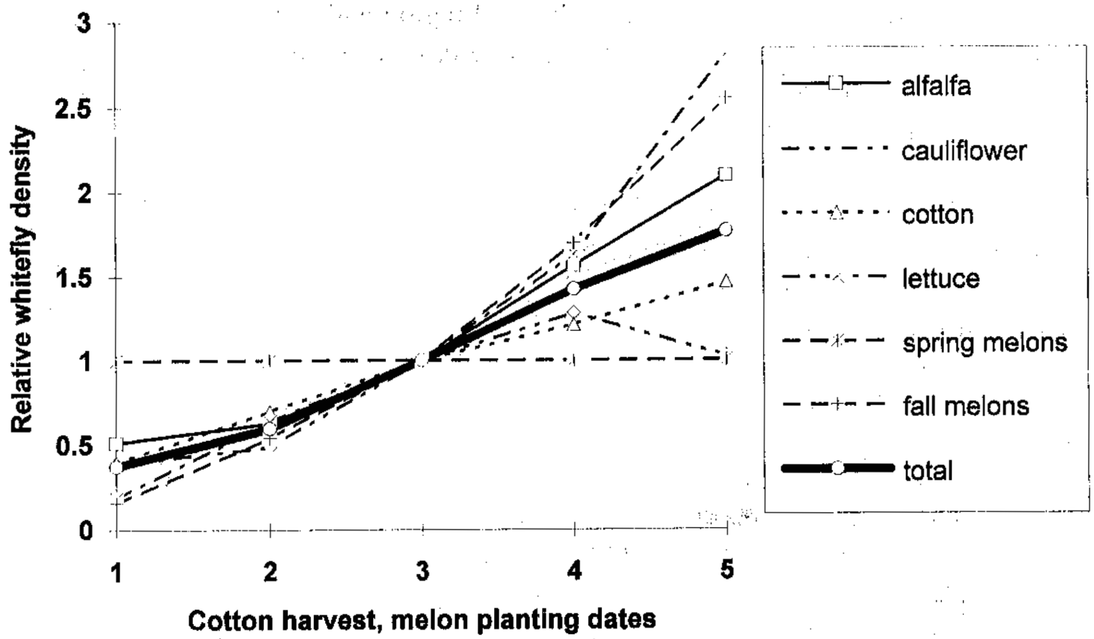


Figure 28. The ratio of the maximum whitefly density for simulation runs with different cotton harvest and fall melon planting dates to the maximum density for the default run.

Table 25. Effect of alfalfa dry down, early cotton harvest, and late melon planting on the maximum adult whitefly density (× 10^{-3}) per acre during the year and on the total region wide adult whitefly population (× 10^{-9}). The values given in the top row of the table refer to different combinations of 10 week alfalfa dry down (full compliance), early cotton harvest at week 32, and late fall melon planting at week 37 as follows:

1: Alfalfa dry down, cotton harvested week 32

2: Alfalfa dry down, fall melon planted week 37

3: Alfalfa dry down, cotton harvested week 32, melon planted week 37.

	1	2	3
alfalfa	184	184	184
asparagus	1850	1947	1850
broccoli	701	4334	335
cabbage	821	6819	578
carrot	52	59	35
cauliflower	671	5809	411
celery	16	19	7
cereal	275	275	275
corn	14	14	14
cotton	14756	50244	14756
cucumber	204	3583	204
garlic	13	8	5
lettuce	183	92	2
spring melon	2721	2721	2721
fall melon	7617	3953	72
onion	259	509	259
sorghum	203	195	203
squash	99	99	99
sugarbeet	1819	2414	1723
tomato	160	160	160
total	133	182	62

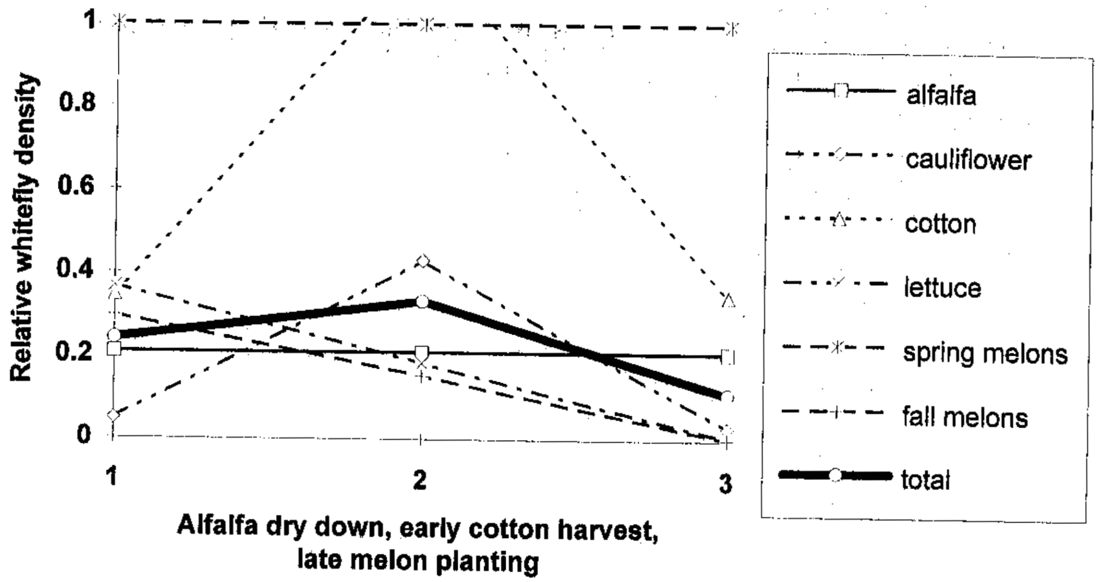


Figure 29. The ratio of the maximum whitefly density for simulation runs with different combinations of alfalfa dry down, early cotton harvest, and late melon planting to the maximum density for the default run.

on cauliflower 0.43. Both of these scenarios provided better relative reduction in whitefly density than the best cotton-melon scenario for all crops. Combining all three strategies was the best of all, with a regional relative whitefly population of 0.11; relative whitefly density on alfalfa 0.21; on cotton 0.35; on fall melon 0.003; and on cauliflower 0.03. Note that the most dramatic effect of the three strategies was the very low whitefly densities on fall melon.

Discussion

The model of regional whitefly population dynamics was developed primarily to explore the pest management potential of different cropping patterns, such as area planted to different crops and time of planting and harvest, in the Imperial Valley, California. It includes information from 19 crops, their spatial and temporal distributions within square mile sections, wind and temperature data, and the growth rates and dispersal characteristics of the whitefly, *Bemisia tabaci*.

Biological parameters of the whiteflies can not be manipulated in developing a pest management strategy. However, these parameters need to be examined because the actual values of many the parameters are either unknown or are variable. When the model was run for a wide range of parameter values, in most cases the results did not vary greatly. Thus, the fact that accurate measures of most parameters are not available will not effect the conclusions from the model. By far the most sensitive parameter was the growth rate of whiteflies on alfalfa. Unfortunately, the growth of whiteflies on alfalfa is very poorly studied. One of the primary purposes of this type of model is to suggest areas that need more research. This result of the model clearly suggests that experiments to determine the growth rate of whiteflies on alfalfa should be given high priority.

Because of several simplifications in the model and especially the uncertainties in the growth rate of whiteflies on alfalfa, the model cannot be relied on to make definitive answers to pest management questions. This is especially true for the results at the end of the season when the effects of rain and cold on whitefly survival were not included in the model. It can, however, be used to suggest promising alternatives that can then be followed up with further field research.

The most effective single pest management strategy suggested by the model was not planting alfalfa. No alfalfa in the simulation resulted in a regional population reduction of 67% and an even larger reduction on celery and squash. However, it had little effect on whitefly densities in tomato, spring melon, and cotton. Of course, this strategy is unrealistic since farmers are unlikely to stop growing alfalfa altogether and even if they did, they would certainly plant something else. No alfalfa for a period of 6 or more weeks was the second best strategy and gave results almost as good as no alfalfa. This strategy has been discussed as an alfalfa dry down but this description is a simplification and is actually closer to a top kill of the alfalfa plants. It also suggests that alfalfa must be absent for at least 6 weeks. If alfalfa is absent for 2 weeks, for example, the whitefly problem may even be aggravated probably because many of the whiteflies that leave alfalfa land on cotton where they can grow better than on alfalfa. The model also demonstrates the importance of getting fairly high compliance with dry down. The greater the number of farmers that cooperate, the greater the benefit. These results again suggests that alfalfa is the most critical crop in this system and deserves further research.

One of the effects of the silverleaf whitefly outbreak in the Imperial Valley was to discourage the planting of fall melon. The model predicted that the regional whitefly population would be reduced by half if no fall melon were planted. The same result would occur if no cotton were planted. However, an even larger regional population reduction would occur if no spring melon were planted. Reducing the number of acres planted rather than eliminating the crop was generally proportionally effective in reducing whiteflies. However, in many cases, reducing the number of acres of a crop actually resulted in an increase of whitefly density on that crop. For example, reducing the acres planted to spring melon from the default area to 75% resulted in a 2.6 increase in whitefly density on spring melon. This even lead to a slightly higher regional wide population of whiteflies.

Another strategy that is often suggested is to plant cotton as early as possible. If the length of the growing season for cotton was not affected by planting date, the model suggested that there was no clear advantage to planting cotton earlier or later. Actually, the worse time for planting was an intermediate date near 4/9. In any case the effect was small on the regional wide whitefly population. Harvesting cotton early did have a fairly large effect of reducing whitefly populations, especially on cotton. However, an even greater effect in reducing region wide populations resulted when fall melon were planted late. On the other hand, if alfalfa dry down was adopted, then early cotton harvest was more effective than late fall melon planting.

The model was also designed to explore the spatial arrangements of crops which may have significant effects on whitefly population dynamics. For example, choosing different sections for alfalfa dry down had almost as important effect as changing the number of sections that were dried down. However, the model has not been run using other spatial cropping arrangements.

Clearly, these conclusions are not definitive since the model needs extensive validation against field data. Also, SLWF is being intensively studied and the new data need to be incorporated into the model. Some of this information can be immediately incorporated into the model simply by changing the parameter values, such as the growth rates on different crops. However, it may be necessary to extend or modify the model to make it more realistic. For, example, in the model whiteflies only disperse from a crop when it is harvested while in reality they may leave a plant for other reasons (Bellows et al. 1988) and the whiteflies appear to exist in two different dispersal forms (Byrne and Houck 1990). Also, the model does not include the effect of winter conditions on whitefly survival and dispersal. How important these factors are will have to wait for experimental results. Similarly, it may prove necessary to include some natural enemies or whitefly age structure to more realistically model the whitefly population dynamics.

In conclusion, although data on demographic variables associated with SLWF in the Imperial Valley are limited, the SLWF simulation model has demonstrated utility in exploring the consequences of regional pest management strategies. It suggests that alfalfa may be the most critical crop in understanding the regional dynamics of whitefly. It will be especially important to know how well whiteflies multiply on alfalfa. Simulations have also indicated that the best strategy for whitefly management may be a summer alfalfa dry down (or top kill) and that the effect on whiteflies will be especially dramatic if combined with early cotton harvest and late fall melon.

References

Bartlett, A. C. and N. J. Gawel. 1993. Determining whitefly species. Science 261: 1334-1335.

Baumgärtner, J. and E. Yano. 1990. Whitefly population dynamics and modelling. In: D. Gerling (ed.). Whiteflies: their bionomics, pest status and management. pp. 123- 146. Andover, Hants, UK. Intercept., Ltd.

Bellows, T. S., Jr., and K. Arakawa. 1988. Dynamics of preimaginal populations of *Bemisia tabaci* (Homoptera: Aleyrodidae) and *Eretmocerus* sp. (Hymenoptera: Aphelinidae) in southern California cotton. Environmental Entomology 17: 483-487.

Bellows, T. S., Jr., T. M. Perring, K. Arakawa, and C. A. Farrar. 1988. Patterns in diel flight activity of *Bemisia tabaci* (Homoptera: Aleyrodidae) in cropping systems in southern California. Environmental Entomology 17: 225-228.

Butler, G. D., T. J. Henneberry, and W. D. Hutchison. 1986. Biology, sampling and population dynamics of *Bemisia tabaci*. Agricultural Zoology Reviews 1: 167-195.

Byrne, D. N. 1991. Whitefly biology. Annual Review of Entomology 36: 431-457.

Byrne, D. N. and M. A. Houck. 1990. Morphometric identification of wing polymorphism in *Bemisia tabaci* (Homoptera: Aleyrodidae). Annals of the Entomological Society of America 83: 487-493.

Campbell, B. C., J. E. Duffus, and P. Baumann. 1993. Determining whitefly species. Science 261: 1333.

Cohen, S., J. E. Duffus, and H. Y. Liu. 1992. A new *Bemisia tabaci* biotype in the Southwestern United States and its role in silverleaf of squash and transmission of lettuce infectious yellow virus. Phytopathology 82: 86-90.

Gill, R. J. 1992. A review of the sweetpotato whitefly in southern California. Pan-Pacific Entomologist 68: 144-152

Gonzalez, R. A., G. E. Goldman, E. T. Natwick, H. R. Rosenberg, J. I. Grieshop, S. R. Sutter, T. Funakoshi, S. Davila-Garcia. 1992. Whitefly invasion in Imperial Valley costs growers, workers millions in losses. California Agriculture 46: 7-8.

Gruenhagen, N. M., T. M. Perring, L. G. Bezark, D. M. Daoud, T. F. Leigh. 1993. Silverleaf whitefly present in the San Joaquin Valley. California Agriculture 47: 4-6.

Henneberry, T. J., G. D. Butler, Jr. 1992. Whiteflies as a factor in cotton production with specific reference to *Bemisia tabaci* (Gennadius). 1992 Proceedings - Beltwide Cotton Production Research Conferences. v. 2, p. 674-683. Memphis, Tenn.: National Cotton Council of America.

Horowitz, A. R., H. Podoler, and D. Gerling. 1984. Life table analysis of the tobacco whitefly *Bemisia tabaci* (Gennadius) in cotton fields in Israel. Acta Oecologica/Oecologia Applicata 5: 221-233.

Leslie, P. H. 1945. On the use of matrices in certain population mathematics. Biometrika 33: 183-212.

Meyerdirk, D. E., D. L. Coudriet, and N. Prabhaker. 1986. Population dynamics and control strategy for *Bemisia tabaci* in the Imperial Valley, California. Agriculture, Ecosystems and Environment 17: 61-67.

Perring, T. M., A. Cooper, D. J. Kazmer, C. Shields, and J. Shields. 1991. New strain of sweetpotato whitefly invades California vegetables. California Agriculture 45: 10-12.

Perring, T. M., A. D. Cooper, R. J. Rodriguez, C. A. Farrar, T. S. Bellows, Jr. 1993a. Identification of a whitefly species by genomic and behavioral studies. Science 259: 74-77.

Perring, T. M., C. A. Farrar, A. D. Cooper, T. S. Bellows, Jr., R. J. Rodriguez. 1993b. Determining whitefly species: response. Science 261: 1334-1335.

Powell, D. A. and T. S. Bellows, Jr. 1992. Adult longevity, fertility and population growth rates for *Bemisia tabaci* (Genn.) (Hom., Aleyrodidae) on the host plant species. Journal of Applied Entomology 113: 68-78.

Riley, D. and D. Wolfenbarger. 1993. Cultivated hosts and population dynamics of sweetpotato whitefly in the lower Rio Grande Valley of TX. 1993 Proceedings - Beltwide Cotton Production Research Conferences. v. 2, p. 667-670. Memphis, Tenn.: National Cotton Council of America.

University of California, Riverside. 1993. The silverleaf whitefly. Published in cooperation with the Imperial County Whitefly Management Committee.

von Arx, R. J. Baumgärtner, and V. Delucchi. 1983. A model to simulate the population dynamics of *Bemisia tabaci* Genn. (Stern., Aleyrodidae) on cotton in the Sudan Gezira. Zeitschrift für Angewandte Entomologia 96: 341-363

Watson, T. F., J. C. Silvertooth, A. Tellez, and L. Lastra. 1992. Seasonal dynamics of sweetpotato whitefly in Arizona. Southwestern Entomologist 17: 149-167.

Zalom, F. G., E. T. Natwick, and N. C. Toscano. 1985. Temperature regulation of *Bemisia tabaci* (Homoptera: Aleyrodidae) populations in Imperial Valley cotton. Journal of Economic Entomology 78: 61-64.

Appendix 1

Model Implementation

The simulation program was written in C++ using the Borland® compiler for the MS-DOS® operating system. The complete source code is given below. The code exists in four primary files (WF.CPP, CROP.CPP, SIM.CPP, WFBAT.CPP) and three header files (WF.H, CROP.H, AND SIM.H). Also given in this appendix are the two general parameter files, one of the crop files (CAB.CRP, for cabbage), and one of the files of dispersion factors (ARIA.01, for week1).

Crucial to understanding the program are the **Crop** class (defined in CROP.H) and the **Whitefly** class (defined in SIM.H). These are data structures used to store the parameter values and state variables for crops and whiteflies and the functions for operating on them. The **Crop** class maintains a field that along with the number of acres of each crop in each section gives the whitefly population size on each crop in each section and the carrying capacity of that crop for whiteflies. All the other whitefly parameters and properties are stored in the **Whitefly** class which contains the intrinsic rate of growth, the growth equation, the generation time, dispersal parameters, and the routines for mortality, dispersion, population growth, and writing output results to files.

The program starts in the function **main()**, which provides a summary of the operation of the program. The first three lines of code perform various initialization functions. First, an instance of the **Whitefly** class is created which automatically calls the **Whitefly** constructor, **Whitefly::Whitefly()**, given in SIM.CPP. This constructor reads the general whitefly parameter file and initializes the **Whitefly** class. Next, the function **ReadCropParameters()** is called which reads the general crop parameter file and stores the parameter values in global variables. The final initialization function, **CropInit()**, creates and array of **Crop** objects. The number of **Crop** objects is determined by the number of crops listed in the general crop parameter file. The creation of the **Crop** array calls the **Crop** constructor (**Crop::Crop** in CROP.CPP) for each crop.

The program then enters the simulation loop which calls two functions for each week of the simulation. The first function, **SetUpCrops()**, determines which crops should be planted in which sections for the current week and calculates the number of whiteflies leaving each section. The second function, **WfSim()**, carries out the whitefly simulation, which involves functions for calculating dispersal from each section to every other section, population growth on each crop in each section, any mortality acting on whiteflies, and writes the results to output files.

After the simulation is completed, the final function, **WriteParamFile()**, writes the parameter values to another output file.

```
/*   wf.cpp contains the main() function
 *   for a model of whitefly population in Imperial Valley on
 *    19 different crops.  The Imperial Valley is conceptially partioned
 *    into a 42 row by 36 column grid of 1 square mile 'sections'.
 *
 *    The program gets parameter values from files:
 *    croppar.dat and simpar.dat.
 *
 *    It also reads data from files in the directory defDir:
 *    aria.?? and a set of crop files for each crop used in the model.
 *
 *    Output is written to three files in dirctory outDir:
 *    outFile.out for whitefly densities,
 *    outFilec.out for acres of each crop during each week, and
 *    outFilep.out for all parameter values used by the simulation.
 *    outFiles.out for numbers in each section every secOutFreq week.
 *
 *    9/11/93 Larry Wihoit DPR
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "wf.h"
#include "crop.h"
#include "sim.h"

char defDir[25] = "data\\";
char outDir[25] = "runs\\";
char outFile[8];

char cropParFile[30] = "croppar.dat";
char simParFile[30] = "simpar.dat";

char gCropName[maxNumCrops][20];
char gFileName[maxNumCrops][20];
int gChange[maxNumCrops];
int gWeekPlantStarts[maxNumCrops];
int gWeekPlantEnds[maxNumCrops];
int gWeekHarvestStarts[maxNumCrops];
int gWeekHarvestEnds[maxNumCrops];
float gPropAreaChanged[maxNumCrops];
float gPropSecUsed[maxNumCrops];
float gPropHarvested[maxNumCrops];
float gPropWeekHarv[maxNumCrops];
float gPropSurvHarv[maxNumCrops];
float gWfCapacity[maxNumCrops];
float gWfGrowthRate[maxNumCrops];
int gSecOut[maxNumCrops];

Whitefly *wfly;
Crop *crop;
int numCrops = 0;

void main(void)
{
  int week;

  wfly = new Whitefly;
  ReadCropParameters();
  CropInit();
  for(week=1; week<=wfly->NumWeeks(); week++) {
    SetupCrops(week);
    WfSim(week);
  }
  WriteParamFile();
}
```

```
/**********************************************************************

   Read in parameters from files for crop characteristics

**********************************************************************/

void ReadCropParameters(void)
{
  char sdum[30], nextCrop[30];
  int include, loc;
  FILE *fpParamFile;

  if ((fpParamFile = fopen(cropParFile,"r")) == NULL)
    Error( "Cannot open file", cropParFile, "");

  printf("Reading crop parameters...\n");

  loc = fscanf(fpParamFile,"%s", nextCrop);
  while( loc != EOF && numCrops<maxNumCrops ) {
    fscanf(fpParamFile, "%d", &include);
    if( include ) {
      numCrops++;
      fscanf(fpParamFile,"%s %s", sdum, gCropName[numCrops-1]);
      fscanf(fpParamFile,"%s %s", sdum, gFileName[numCrops-1]);
      fscanf(fpParamFile,"%s %d", sdum, &gChange[numCrops-1]);
      fscanf(fpParamFile,"%s %d", sdum, &gWeekPlantStarts[numCrops-1]);
      fscanf(fpParamFile,"%s %d", sdum, &gWeekPlantEnds[numCrops-1]);
      fscanf(fpParamFile,"%s %d", sdum, &gWeekHarvestStarts[numCrops-1]);
      fscanf(fpParamFile,"%s %d", sdum, &gWeekHarvestEnds[numCrops-1]);
      fscanf(fpParamFile,"%s %f", sdum, &gPropAreaChanged[numCrops-1]);
      fscanf(fpParamFile,"%s %f", sdum, &gPropSecUsed[numCrops-1]);
      fscanf(fpParamFile,"%s %f", sdum, &gPropHarvested[numCrops-1]);
      fscanf(fpParamFile,"%s %f", sdum, &gPropWeekHarv[numCrops-1]);
      fscanf(fpParamFile,"%s %f", sdum, &gPropSurvHarv[numCrops-1]);
      fscanf(fpParamFile,"%s %f", sdum, &gWfCapacity[numCrops-1]);
      fscanf(fpParamFile,"%s %f", sdum, &gWfGrowthRate[numCrops-1]);
      fscanf(fpParamFile,"%s %d", sdum, &gSecOut[numCrops-1]);
    } else {
      for(int i=1; i<=15; i++) {
        fscanf(fpParamFile,"%s %s", sdum, sdum);
      }
    }
    loc = fscanf(fpParamFile,"%s", nextCrop);
  };

  fclose(fpParamFile);
}
```

```
/************************************************************************

  Allocate memory and set up arrays for crop data

************************************************************************/

void CropInit(void)
{
  if( wfly->doSim )
    printf("Starting whitefly simulation...\n");
  else
    printf("Setting up crop data...\n");

  crop = new Crop[numCrops];   // make 1 offset-- overload new?
  printf("Fixing sections...\n");
  wfly->SetInitWf();
  wfly->FixSections();
  wfly->WriteOutputFiles(0);
}

/************************************************************************

  Determine which crops are planted for current week

************************************************************************/

void SetupCrops(int week)
{
  int c;

  printf(" Week %d  ", week);
  if( wfly->doSim )
    wfly->WeeklyInit();
  for(c=0; c<numCrops; c++)
    crop[c].Setup(week);
}

/************************************************************************

   The driver program MAIN calls the function WFSIM on a weekly
basis.

************************************************************************/

void WfSim(int week)
{
  if( wfly->doSim ) {
    wfly->Disperse(week);
    wfly->PopGrow(week);
    wfly->Mortality(week);
  }
  wfly->WriteOutputFiles(week);
}

/************************************************************************

    Write parameters to file

************************************************************************/

void WriteParamFile(void)
{
  wfly->WriteParameterFile();
  printf("Finished!\n");
}
```

```
/************************************************************************

  Function to handle run-time errors.

************************************************************************/

void Error(char *str1, char *str2, char *str3)
{
  fprintf(stderr, "Run-time error...\n");
  fprintf(stderr, "%s %s %s\n", str1, str2, str3);
  exit(1);
}


void *operator new(size_t size)
{
  void *p;

  if( !(p = malloc(size)) )
    Error("Out of memory", "", "");

  return p;
}
```

```
/*   crop.cpp modifies the crops files for the spwf model
 *    Contains functions for the Crop class
 *    3/11/93 Larry Wihoit DPR
 */


#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
#include "wf.h"
#include "crop.h"
#include "sim.h"

int Crop::cnum = -1;

/************************************************************************

    Read crop parameter values from files

************************************************************************/

Crop::Crop(void)
{
  int cs;
  char cropFile[50];
  FILE  *fpCropFile;

  cnum++;
  strcpy(name, gCropName[cnum]);
  strcpy(fname, gFileName[cnum]);
  change = gChange[cnum];
  weekPlantStarts = gWeekPlantStarts[cnum];
  weekPlantEnds = gWeekPlantEnds[cnum];
  weekHarvestStarts = gWeekHarvestStarts[cnum];
  weekHarvestEnds = gWeekHarvestEnds[cnum];
  propAreaChanged = gPropAreaChanged[cnum];
  propSecUsed = gPropSecUsed[cnum];
  propHarvested = gPropHarvested[cnum];
  propWeekHarv = gPropWeekHarv[cnum];
  propSurvHarv = gPropSurvHarv[cnum];
  wfCapacity = gWfCapacity[cnum];
  wfGrowthRate = gWfGrowthRate[cnum];
  secOut = gSecOut[cnum];

  gv = wfly->GrowthFunc(wfGrowthRate);

  totalAcres = 0.0;
  totalWf = 0.0;

  sprintf(cropFile,"%s%s.crp", defDir, fname);

  /* open crop file */
  if ((fpCropFile = fopen(cropFile,"r")) == NULL)
    Error("Cannot open crop file", cropFile, "");

  int d;
  float fd;
  for(numSections = 0;
      fscanf(fpCropFile,"%d %d %f %d %d %d",&d,&d,&fd,&d,&d,&d) != EOF;
      numSections++ );

  fseek(fpCropFile, 0L, SEEK_SET);

  if( numSections == 0 ) Error("Could not read data in", "cropFile", "");
  row = new int[numSections];
  col = new int[numSections];
```

```
  section = new int[numSections];
  acres = new float[numSections];
  weekPlant = new int[numSections];
  weekHarvest = new int[numSections];
  numWf = new float[numSections];
  planted = new int[numSections];

  for(cs=0; cs<numSections; cs++ ) {

  fscanf(fpCropFile,"%d%d%f%d%d%f",&row[cs],&col[cs],&acres[cs],&weekPlant[cs],&weekHarvest[cs],&
numWf[cs]);
    section[cs] = row[cs]*HCOL + col[cs];
    if( section[cs] > allSections )
      Error("The row and column value is too out of bound in Crop::Crop", "", "");
    planted[cs] = (weekPlant[cs] < weekHarvest[cs]) ? 0 : 1;
    if( !planted[cs] )
      numWf[cs] = 0.0;
    else
      numWf[cs] *= wfly->initWfProp;
  }

  fclose(fpCropFile);

  // If crop characteristics are to be changed from default
  if( change ) {

    // Calculate which of the default sections are to be used
    if( propSecUsed > 0.0 && propSecUsed < 1.0 ) {
      int numNewSections = propSecUsed * numSections;
      for( ; numSections > numNewSections; numSections--) {
        for( cs=random(numSections); cs<numSections-1; cs++) {
          row[cs] = row[cs+1];
          col[cs] = col[cs+1];
          acres[cs] = acres[cs+1];
          weekPlant[cs] = weekPlant[cs+1];
          weekHarvest[cs] = weekHarvest[cs+1];
          numWf[cs] = numWf[cs+1];
        }
      }
    } else if( propSecUsed <= 0.0 ) numSections = 0;


    // Calculate new area of crop in each section
    if( propAreaChanged > 0.0 && (propAreaChanged < 1.0 || propAreaChanged > 1.0) ) {
      for( cs=0; cs<numSections; cs++ ) {
        acres[cs] *= propAreaChanged;
      }
    }

    // Set the new planting and harvesting dates in each section
    int numPWks = weekPlantEnds - weekPlantStarts + 1;
    int numHWks = weekHarvestEnds - weekHarvestStarts + 1;
    for( cs=0; cs<numSections; cs++ ) {
      weekPlant[cs] = weekPlantStarts + random(numPWks);
      weekHarvest[cs] = weekHarvestStarts + random(numHWks);
    }

    // Determine which sections will be harvested
    if( propHarvested >= 0.0 && propHarvested < 1.0 ) {
      int numSecNotHarv = (1.0 - propHarvested) * numSections;
      if( numSecNotHarv == numSections ) {
        for( cs=0; cs<numSections; cs++ ) {
          weekPlant[cs] = 0;
          weekHarvest[cs] = 0;
        }
      } else {
        int secNotHarv = 0;
```

```
        int numTries = 0;
      do {
        cs = random(numSections);
        if( weekPlant[cs] != 0 ) {
          weekPlant[cs] = 0;
          weekHarvest[cs] = 0;
          secNotHarv++;
        }
        if( ++numTries == 32766 )
          Error("Couldn't get all sections harvested in crop initialization", "", "");
      } while( secNotHarv < numSecNotHarv );
    }
  }

} else { // if the default crop characteristics are to be used
  weekPlantStarts = weekPlantEnds = weekPlant[0];
  weekHarvestStarts = weekHarvestEnds = weekHarvest[0];

  for( cs=1; cs<numSections; cs++ ) {
    weekPlantStarts = min(weekPlant[cs], weekPlantStarts);
    weekPlantEnds = max(weekPlant[cs], weekPlantEnds);
    weekHarvestStarts = min(weekHarvest[cs], weekHarvestStarts);
    weekHarvestEnds = max(weekHarvest[cs], weekHarvestEnds);
  }
}

summerCrop = ( weekPlantStarts < weekHarvestEnds ) ? 1 : 0;
}
```

```
/***********************************************************************

  Determine which sections should be planted and calculate number of
  whiteflies leaving in sections where crop is harvested.

***********************************************************************/

void Crop::Setup(int week)
{
  int cs, newState;
  int (Crop::*newPlant)(int, int, int);

  newPlant = summerCrop ? &(Crop::SummerPlant) : &(Crop::WinterPlant);

  for(cs=0; cs<numSections; cs++) {
    newState = (this->*newPlant)(week, weekPlant[cs], weekHarvest[cs]);
    if( wfly->doSim ) {
      if( newState == 0 && planted[cs] == 1 ) {
        wfly->AddLeaving(row[cs], col[cs], propSurvHarv*numWf[cs]);
        numWf[cs] = 0.0;
      } else if( propWeekHarv > 0.0 && propWeekHarv <= 1.0 )  {
        wfly->AddLeaving(row[cs], col[cs], propWeekHarv*propSurvHarv*numWf[cs]);
        numWf[cs] *= 1.0 - propWeekHarv;
      }
    }
    planted[cs] = newState;
  }
}

int Crop::SummerPlant(int week, int plant, int harvest)
{
  return ( week>=plant && week<=harvest ) ? 1 : 0;
}

int Crop::WinterPlant(int week, int plant, int harvest)
{
  return ( week>=plant || week<=harvest ) ? 1 : 0;
}

/***********************************************************************

  Determine the index in the list of sections in which the crop is
  planted (the "crop section") given the global section number.  If the
  crop is not planted in this global section (either because it never
  is or because it is not planted during the current week), then
  return -1.

***********************************************************************/


int Crop::FindSec(int s)
{
  int mid;
  int lower = -1;
  int upper = numSections;

  while( upper - lower > 1) {
    mid = (upper + lower) >> 1;
    if( s == section[mid] ) {
      if( !planted[mid] )
        return -1;
      return mid;
    }
    else if( s > section[mid] )
      lower = mid;
    else
```

```
      upper = mid;
  }
  return -1;
}

/************************************************************************

  Determine the index in the list of sections in which the crop is
  planted (the "crop section") given the global section number.  If the
  crop is never planted in this global section, then return -1.

************************************************************************/


int Crop::FindAllSec(int s)
{
  int mid;
  int lower = -1;
  int upper = numSections;

  while( upper - lower > 1) {
    mid = (upper + lower) >> 1;
    if( s == section[mid] )
      return mid;
    else if( s > section[mid] )
      lower = mid;
    else
      upper = mid;
  }
  return -1;
}

/************************************************************************

  Determine whether or not the crop is growing during week week in section cs.

************************************************************************/


int Crop::Planted(int week, int cs)
{
  if( (summerCrop && ( week>=weekPlant[cs] && week<=weekHarvest[cs] )) ||
      (!summerCrop && ( week<weekHarvest[cs] || week>weekPlant[cs] )) )
    return 1;
  return 0;
}
```

```
/*    sim.cpp contains functions for the Whitefly class
 */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "wf.h"
#include "sim.h"
#include "crop.h"

float Mort(float a, float b) { return a*(1.0-b) + b; }

/************************************************************************

    Read simulation parameter values from files

************************************************************************/

Whitefly::Whitefly(void)
{
  char sdum[50];
  int s, n;
  FILE *fpParamFile, *fpDD, *fpSD;

  if ((fpParamFile = fopen(simParFile,"r")) == NULL)
    Error("Cannot open file", simParFile, "");

  fscanf(fpParamFile,"%s %s", sdum, outFile);
  fscanf(fpParamFile,"%s %d", sdum, &doSim);
  fscanf(fpParamFile,"%s %d", sdum, &totalWeeks);
  fscanf(fpParamFile,"%s %f", sdum, &roi);
  fscanf(fpParamFile,"%s %f", sdum, &growthFactor);
  fscanf(fpParamFile,"%s %f", sdum, &genTime);
  fscanf(fpParamFile,"%s %f", sdum, &dispersalProp);
  fscanf(fpParamFile, "%s %f", sdum, &initWfProp);
  fscanf(fpParamFile,"%s %d", sdum, &wkWfSecOut);
  fscanf(fpParamFile,"%s %d", sdum, &secOutFreq);
  fscanf(fpParamFile, "%s %d", sdum, &initNumSec);
  fscanf(fpParamFile, "%s", sdum);

  if( initNumSec > 0 ) {
    initRow = new int[initNumSec];
    initCol = new int[initNumSec];
    initNum = new float[initNumSec];

    for(n=0; n<initNumSec; n++)  {
      fscanf(fpParamFile, "%s %d %d %f", sdum, &initRow[n], &initCol[n], &initNum[n]);
      if( strcmp(sdum, "Random_num_seed") == 0 )
        Error("Not enough rows and cols listed for initial number of sections in ",
         simParFile, "");
    }
  }

  do {
    if( fscanf(fpParamFile, "%s", sdum) == EOF )
      Error( "Did not find Random_num_seed in", simParFile, "");
  } while ( strcmp( sdum, "Random_num_seed") != 0 );

  fscanf(fpParamFile,"%u", &rseed);
  srand(rseed);

  /* Read in insecticide information */
  fscanf(fpParamFile,"%s %d",sdum, &numSprayedCrops);
  fscanf(fpParamFile,"%s %f",sdum, &icWFM);
  fscanf(fpParamFile,"%s %d",sdum, &icResidTime);
  fscanf(fpParamFile,"%s %f",sdum, &residZero);
```

```
  fscanf(fpParamFile,"%s %d",sdum, &icTime);
  fscanf(fpParamFile,"%s %d",sdum, &icNextTime);
  fscanf(fpParamFile,"%s %d",sdum, &icLastTime);

  fscanf(fpParamFile,"%s",sdum);

  if( numSprayedCrops ) {
    for (s=0; s<numSprayedCrops; s++) {
      sprayedCrop[s] = new char[20];
      fscanf(fpParamFile,"%s", sprayedCrop[s]);
      if( strcmp(sprayedCrop[s], "Degree_days") == 0 )
        Error("Not enough crops to be sprayed listed in", simParFile, "");
    }
  }

  do {
    if( fscanf(fpParamFile, "%s", sdum) == EOF )
      Error( "Did not find Degree_days in", simParFile, "");
  } while ( strcmp( sdum, "Degree_days") != 0 );

  /* Read in degree day values */
  degd = new float[totalWeeks];
  for (s=0; s<totalWeeks; s++)  {
    if( fscanf(fpParamFile,"%f\n", &degd[s]) == EOF )
      Error("Not enough degree day values in", simParFile, "");
  }

  fclose(fpParamFile);
}

/***********************************************************************

    REASSIGN DISPERSERS TO NEW SECTIONS

***********************************************************************/
void Whitefly::Disperse(int week)
{
  int c, s, cs;
  int b, rr, cc, k;
  int rw, cl, brw, bcl;
  int minrr, maxrr, mincc, maxcc;
  char  grid_fname[25];
  FILE *fpGrid;

  sprintf(grid_fname,"%saria.%02d", defDir, week);

  if ((fpGrid = fopen(grid_fname,"r")) == NULL)
    Error("Cannot open input file", grid_fname, "");

  // read in weekly dispersal distribution grid

  b = (HZ-1)/2;

  for (rr=0; rr<HZ; rr++) {
    for (cc=0; cc<HZ; cc++) {
      fscanf(fpGrid,"%e",&grid[rr][cc]);
    }
    fscanf(fpGrid,"\n");
  }
  fclose(fpGrid);

  for( s=0, rw=0; rw<HROW; rw++ ) {
    for( cl=0; cl<HCOL; cl++, s++) {
      if( leaving[s] > 0.0 ) {
        brw = b - rw;
        bcl = b - cl;
        minrr = max( brw, 0 );
```

```
        maxrr = min( HZ, brw+HROW );
        mincc = max( bcl, 0);
        maxcc = min( HZ, bcl+HCOL );
        for( rr=minrr; rr<maxrr; rr++ ) {
          k = (rr - brw)*HCOL + mincc - bcl;
          for( cc=mincc; cc<maxcc; cc++, k++ ) {
            // if( k<0 || k>=allSections) Error( "k out of bounds", "", "");
            arriving[k] += dispersalProp * leaving[s] * grid[rr][cc];
          }
        }
        arriving[s] += (1.0 - dispersalProp) * leaving[s];
      }
    }
  }

  /* partition recent arrivals and movers to new crops  */

  for( c=0; c<numCrops; c++ ) {
    for( cs=0; cs<crop[c].numSections; cs++ ) {
      s = crop[c].section[cs];
      if( arriving[s] > 0.0 && crop[c].planted[cs])
        crop[c].numWf[cs] += crop[c].acres[cs]/640.0 * arriving[s];
    }
  }
}

/*************************************************************************

    ALLOW POPULATION GROWTH TO OCCUR

    Exponential growth equation and value for parameter roi comes from
    Zalom et al. 1985 (J. Econ. Entomol 76: 61-64).  Note: these data
    are from the A strain. The values of gv are calculated from
    estimates of the rate of growth
    of the whitefly on different crops.  This was done by asking
    people what the relative population size of whitefly would be
    after one generation on different crops assuming they start at
    the same population density.  Thus:

    exp(gf[x]*genTime*roi) = growth_vec[x]*exp(genTime*roi),  and
    solving,

    gf[x] = 1 + log(growth_vec[x])/(genTime*roi).

    Finally,
    gv[x] = roi* gf[x].

    This array is calculated once, in function ReadParameters().

*************************************************************************/

void Whitefly::PopGrow(int week)
{
  int c, cs;
  float max;

  for( c=0; c<numCrops; c++) {
    for( cs=0; cs<crop[c].numSections; cs++ ) {
      if( crop[c].planted[cs] ) {
        crop[c].numWf[cs] *=  exp(crop[c].gv*degd[week-1]);
        max = crop[c].wfCapacity*crop[c].acres[cs];
        if( crop[c].numWf[cs] > max )
          crop[c].numWf[cs] = max;
       }
    }
  }
}
```

```
/**********************************************************************

    IMPOSE ANY SPECIAL MORTALITY FACTORS SUCH AS PLOWDOWNS,
    PESTICIDE APPLICATIONS OR CATOSTROPHIC WEATHER

    Impose any special mortality factors such as early
    plowdowns, pesticide applications, or catostrophic
    weather.
    This code can be customized to investigate events which
    lower the population of whiteflies on a crop without causing
    dispersal to adjacent crops.

**********************************************************************/

void Whitefly::Mortality(int week)
{
  static float resid;
  int i, j, k;

  if( week == 1 )
    iicTime = icTime;

  if (numSprayedCrops) {
    if(week == icTime) {
      /* Calculate mortality from insecticide.
       * The macro function MORT is used since there may be residual
       * mortality from a previous spray.
       */
      wfMortIc = Mort(wfMortIc,icWFM);

      if( icResidTime > 0 )
        resid = pow( residZero, 1.0/icResidTime );
      else
        resid = 0.0;

      iicTime = icTime;
      icTime += icNextTime;
      if(icTime > icLastTime)
        icTime = 0;
    }

    /* Calculate the reduction in mortality from insecticide decay  */
    if(week > iicTime)
      wfMortIc *= resid;

    /* Calculate number of wf surviving in each section on each crop */
    if(week >= iicTime && wfMortIc > 0.0) {
      for( i=0; i<numSprayedCrops; i++ ) {
        for(j=0; j<numCrops; j++ ) {
          if(strcmp(sprayedCrop[i], crop[j].name) == 0 ) {
            for( k=0; k<crop[j].numSections; k++ ) {
              crop[j].numWf[k] *= (1.0 - wfMortIc);
            }
            break;
          }
        }
      }
    }
  }
}
```

```
/**********************************************************************

  Initialize whitefly dispersal arrays each week.

**********************************************************************/

void Whitefly::WeeklyInit(void)
{
  for(int s=0; s<allSections; s++) {
    leaving[s] = 0.0;
    arriving[s] = 0.0;
  }
}

/**********************************************************************

  Sets the initial number of sections with whiteflies present according
  to the request of the user

**********************************************************************/

void Whitefly::SetInitWf(void)
{
  int c, cs, s, is;
  float acresPlanted;

  if( !initNumSec )
    return;

  for( c=0; c<numCrops; c++ ) {
    for( cs=0; cs<crop[c].numSections; cs++) {
        crop[c].numWf[cs] = 0.0;
    }
  }

  for( is=0; is<initNumSec; is++ ) {
    s = initRow[is]*HCOL + initCol[is];
    for( acresPlanted=0.0, c=0; c<numCrops; c++ ) {
      if( (cs = crop[c].FindSec(s)) >= 0 )
        acresPlanted += crop[c].acres[cs];
    }
    if( acresPlanted > 0.0 ) {
      for( c=0; c<numCrops; c++ ) {
        if( (cs = crop[c].FindSec(s)) >= 0 )
          crop[c].numWf[cs] = initNum[is]*crop[c].acres[cs]/acresPlanted;
      }
    }
  }
}

/**********************************************************************

  If the total proportion of crops grown in any section for any week
  is greater than 1.0, change proportions of all crops so sum is 1

**********************************************************************/

void Whitefly::FixSections(void)
{
  int week;
  int c, cs, s;
  int adjust = 0, cropChanged = 0;
  float sum;
  FILE  *fpFixSecFile;

  if ((fpFixSecFile = fopen("fixsec.out","w")) == NULL)
    Error("Cannont open fsec.out", "", "");
```

```
  for(c=0; c<numCrops; c++ ) {
    if( crop[c].change ) {
      cropChanged = 1;
      break;
    }
  }

  if( cropChanged ) {
    // Iterate through each section
    for( s=0; s<allSections; s++ ) {

      // Check if total acres of all crop is more than 640, the num of acres per sq mile section
      for(sum=0, c=0; c<numCrops; c++ ) {
        if( (cs = crop[c].FindAllSec(s)) >= 0 )
          sum += crop[c].acres[cs];
      }

      // If the acreage is more, go through each week to see if the total acres is > 640 during
that week
      if( sum > 640.0 ) {
        for( week=1; week<=totalWeeks; week++ ) {

          for( sum=0, c=0; c<numCrops; c++ ) {
            if( (cs = crop[c].FindAllSec(s)) >= 0 && crop[c].Planted(week, cs) )
              sum += crop[c].acres[cs];
          } // for(sum, c...


          // If total acres is above 640 for this week, reduce acres of all crops in that section
          if( sum > 640.0 ) {
            adjust = 1;
            fprintf(fpFixSecFile, "Total acres = %6.0f at r = %3d, c = %3d for week = %3d\n", sum,
s/HCOL, s%HCOL, week);
            for( c=0; c<numCrops; c++ ) {
              if( (cs = crop[c].FindAllSec(s)) >= 0 )
                crop[c].acres[cs] *= 640.0/sum;
            }   // for( c...
          }   // if( sum ...

        }   // for(week...
      }   // if(sum...

    } // for(s...
  } // if(cropChanged)

  if( !adjust )
    fprintf(fpFixSecFile, "No adjustments of crops needed.");
}

/************************************************************************

    Weekly print out of number of whiteflies and total acres per crop.

************************************************************************/

void Whitefly::WriteOutputFiles(int week)
{
  int c, s, cs;
  int rw, cl;
  float totalNumWf=0.0, totalNumAcres=0.0, wfSec, cropSec;
  char  cropOutFile[30], wfOutFile[30], wfSecOutFile[30], cropSecOutFile[30];
  FILE  *fpCropOutFile, *fpWfOutFile, *fpWfSecOutFile, *fpCropSecOutFile;
  static int previousOpenCrop = 0;
  static int previousOpenWf = 0;
  static int previousOpenWfSec = 0;
  static int previousOpenCropSec = 0;
```

```
// Print out crop acres planted to file cropOutFile
sprintf(cropOutFile,"%s%sc.out", outDir, outFile);
if( previousOpenCrop ) {
  if ((fpCropOutFile = fopen(cropOutFile,"a")) == NULL)
    Error("Cannot open crop output file", cropOutFile, "");
} else {
  if ((fpCropOutFile = fopen(cropOutFile,"w")) == NULL)
    Error("Cannot open crop output file", cropOutFile, "");
  previousOpenCrop = 1;
}

fprintf(fpCropOutFile,"%3d\t", week);
for( c=0; c<numCrops; c++ ) {
  crop[c].totalAcres = 0.0;
  for( cs=0; cs< crop[c].numSections; cs++ ) {
    if( crop[c].planted[cs]) {
      crop[c].totalAcres += crop[c].acres[cs];
    }
  }
  totalNumAcres += crop[c].totalAcres;
  fprintf( fpCropOutFile, "%8.0f\t", crop[c].totalAcres);
}
fprintf(fpCropOutFile,"%8.0f\n", totalNumAcres/1000.0);
fclose(fpCropOutFile);

// If simulation run, print out num wf per week per crop in wfOutFile
if( doSim ) {
  sprintf(wfOutFile,"%s%s.out", outDir, outFile);
  if( previousOpenWf ) {
    if ((fpWfOutFile = fopen(wfOutFile,"a")) == NULL)
      Error("Cannot open wf output file", wfOutFile, "");
  } else {
    if ((fpWfOutFile = fopen(wfOutFile,"w")) == NULL)
      Error("Cannot open wf output file", wfOutFile, "");
    previousOpenWf = 1;
  }

  fprintf(fpWfOutFile,"%3d\t", week);
  for( c=0; c<numCrops; c++ ) {
    crop[c].totalWf = 0.0;
    for( cs=0; cs< crop[c].numSections; cs++ ) {
      if( crop[c].planted[cs]) {
        crop[c].totalWf += crop[c].numWf[cs];
      }
    }
    totalNumWf += crop[c].totalWf;
    if( crop[c].totalAcres > 0.0 )
      fprintf( fpWfOutFile, "%8.2f\t", crop[c].totalWf/crop[c].totalAcres);
    else
      fprintf( fpWfOutFile, "    0.00\t");
  }
  fprintf(fpWfOutFile,"%8.2f\n", totalNumWf/1000000.0);
  fclose(fpWfOutFile);
}

// If user wants to see wf per section per week, print it out in wfSecOutFile
if( week%secOutFreq == 0 ) {
  if( wkWfSecOut ) {
    sprintf(wfSecOutFile,"%s%ss.out", outDir, outFile);
    if( previousOpenWfSec ) {
      if ((fpWfSecOutFile = fopen(wfSecOutFile,"a")) == NULL)
        Error("Cannot open wf output file", wfSecOutFile, "");
    } else {
      if ((fpWfSecOutFile = fopen(wfSecOutFile,"w")) == NULL)
        Error("Cannot open wf output file", wfSecOutFile, "");
      previousOpenWfSec = 1;
```

```
      }

      fprintf( fpWfSecOutFile, "week = %3d\n", week);
      for(rw=0; rw<HROW; rw++) {
        for(cl=0; cl<HCOL; cl++) {
          s = rw*HCOL + cl;
          wfSec=0.0;
          for( c=0; c<numCrops; c++ ) {
            if( (cs = crop[c].FindSec(s)) >= 0 )
              wfSec += crop[c].numWf[cs];
          }
          fprintf( fpWfSecOutFile, "%8.2f\t", wfSec);
        }
        fprintf( fpWfSecOutFile, "\n");
      }
      fprintf( fpWfSecOutFile, "\n");
      fclose(fpWfSecOutFile);
    }

  // If user wants to see crop acres per section per week, print it out in cropSecOutFile
    for(c=0; c<numCrops; c++) {
      if( crop[c].secOut ) {
        sprintf(cropSecOutFile,"%s%sc%02d.out", outDir, outFile, c);
        if( previousOpenCropSec ) {
          if ((fpCropSecOutFile = fopen(cropSecOutFile,"a")) == NULL)
            Error("Cannot open crop output file", cropSecOutFile, "");
        } else {
          if ((fpCropSecOutFile = fopen(cropSecOutFile,"w")) == NULL)
            Error("Cannot open crop output file", cropSecOutFile, "");
          previousOpenCropSec = 1;
        }

        fprintf( fpCropSecOutFile, "week = %3d\n", week);
        for(rw=0; rw<HROW; rw++) {
          for(cl=0; cl<HCOL; cl++) {
            s = rw*HCOL + cl;
            if( (cs = crop[c].FindSec(s)) >= 0 )
              cropSec = crop[c].acres[cs];
            else
              cropSec = 0.0;
            fprintf( fpCropSecOutFile, "%8.2f\t", cropSec);
          }
          fprintf( fpCropSecOutFile, "\n");
        }
        fprintf( fpCropSecOutFile, "\n");
        fclose(fpCropSecOutFile);
      }
    }
    }
}
```

```
/************************************************************************

   Create file of all parameter values used in simulation

************************************************************************/

void Whitefly::WriteParameterFile(void)
{
  int c, is, n, noCropsChanged =1;
  char cropname[30], paramFile[30];
  char  cropOutFile[30], cropSecOutFile[30], wfOutFile[30], wfSecOutFile[30];
  FILE  *fpParamFile;

  sprintf(paramFile,"%s%sp.out", outDir, outFile);

  if ((fpParamFile = fopen(paramFile,"w")) == NULL)
    Error("Cannont open parameter file", paramFile, "");

  printf("\nCreating parameter output file %s\n", paramFile);

  if( doSim ) {
    sprintf(wfOutFile,"%s%s.out", outDir, outFile);
    printf("Creating simulation output file %s\n", wfOutFile);
  }

  if( wkWfSecOut ) {
    sprintf(wfSecOutFile,"%s%ss.out", outDir, outFile);
    printf("Creating section output file %s\n", wfSecOutFile);
  }

  sprintf(cropOutFile,"%s%sc.out", outDir, outFile);
  printf("Creating crop output file %s\n", cropOutFile);

  for(c=0; c<numCrops; c++) {
    if( crop[c].secOut ) {
      sprintf(cropSecOutFile,"%s%sc%02d.out", outDir, outFile, c);
      printf("Creating crop section output file %s\n", cropSecOutFile);
    }
  }

  fprintf(fpParamFile, "Name base of output files: %s\n", outFile);
  if( doSim ) {
    fprintf(fpParamFile, "Number of weeks of simulation: %d\n", totalWeeks);
    fprintf(fpParamFile,"\nRate of increase: %8.5f\n", roi);
    fprintf(fpParamFile,"Rate of growth factor: %5.2f\n", growthFactor);
    fprintf(fpParamFile,"Generation time: %6.1f\n", genTime);
    fprintf(fpParamFile,"Prop dispers leaving: %5.2f\n", dispersalProp);
    fprintf(fpParamFile, "Initial proportion of whiteflies: %5.2f\n", initWfProp);

    if( initNumSec > 0 ) {
      fprintf(fpParamFile, "Initial number of sections: %3d\n", initNumSec);
    }
    else {
      fprintf(fpParamFile, "Initial number of sections: all\n");
    }

    if( initNumSec > 0 ) {
      fprintf(fpParamFile, "Rows and columns of initial whiteflies:\n");
      for(is=0; is<initNumSec; is++)
        fprintf(fpParamFile, "  %3d %3d\n", initRow[is], initCol[is]);
    }
  }

  fprintf(fpParamFile, "\nCrops changed from default:\n");
  for(c=0; c<numCrops; c++)  {
    if( crop[c].change ) {
      fprintf( fpParamFile, "  %s\n", crop[c].name);
```

```
      noCropsChanged = 0;
    }
  }

  if( noCropsChanged )
    fprintf( fpParamFile, "   None\n");

  fprintf(fpParamFile,"\nParameters for each crop:\n");

  for(c=0; c<numCrops; c++)  {
    fprintf(fpParamFile,"  \n%s\n", crop[c].name);
    fprintf(fpParamFile,"  Week planting starts: %3d\n", crop[c].weekPlantStarts);
    fprintf(fpParamFile,"  Week planting ends: %3d\n", crop[c].weekPlantEnds);
    fprintf(fpParamFile,"  Week harvest starts: %3d\n", crop[c].weekHarvestStarts);
    fprintf(fpParamFile,"  Week harvest ends: %3d\n", crop[c].weekHarvestEnds);
    fprintf(fpParamFile,"  Prop area changed: %5.2f\n", crop[c].propAreaChanged);
    fprintf(fpParamFile,"  Prop sections used %5.2f\n", crop[c].propSecUsed);
    fprintf(fpParamFile,"  Prop harvested during harv: %5.2f\n", crop[c].propHarvested);
    fprintf(fpParamFile,"  Prop harvested each week: %5.2f\n", crop[c].propWeekHarv);
    fprintf(fpParamFile,"  Prop wf surviving harvest: %5.2f\n", crop[c].propSurvHarv);
    fprintf(fpParamFile,"  Wf carrying capacity: %5.2f\n", crop[c].wfCapacity);
    fprintf(fpParamFile,"  Wf growth rate %6.3f\n", crop[c].wfGrowthRate);
  }

  fprintf(fpParamFile, "\nCrop files used:\n");
  for(c=0; c<numCrops; c++)
    fprintf( fpParamFile, "  %s.crp\n", crop[c].fname);

  fprintf(fpParamFile,"\nRandom num seed: %3u\n", rseed);

  if( doSim ) {
    if( numSprayedCrops ) {
      fprintf(fpParamFile,"\nInsectide used.\n");
      fprintf(fpParamFile,"Whitefly mortality from insecticide: %5.2f\n", &icWFM);
      fprintf(fpParamFile,"Residual time: %3d\n", icResidTime);
      fprintf(fpParamFile,"Residual zero: %5.2f\n", residZero);
      fprintf(fpParamFile,"Week spraying starts: %3d\n", icTime);
      fprintf(fpParamFile,"Weeks between spraying %3d\n", icNextTime);
      fprintf(fpParamFile,"Week spraying ends: %3d\n", icLastTime);
      fprintf(fpParamFile,"Crops that were sprayed: ");

      for (n=0; n<numSprayedCrops; n++) {
        fprintf(fpParamFile,"%s ", sprayedCrop[n]);
      }
    }
    else
      fprintf(fpParamFile,"\nInsectide not used\n");

    int n0, n1, m;
    fprintf(fpParamFile, "\nDegree days for each week:\n");
    for( n0=0, n1=0, m=1; n1<totalWeeks; m++) {
      n0 = m*8 - 8;
      n1 = (m*8 < totalWeeks) ? m*8 : totalWeeks;
      for (n=n0; n<n1; n++)
        fprintf(fpParamFile,"%5.2f\t", degd[n]);
      fprintf(fpParamFile, "\n");
    }
  }

  fclose(fpParamFile);
}

/*
inline float Whitefly::GrowthFunc(float cropGr)
{
  return roi * log(growthFactor*cropGr*(exp(genTime*roi) -1) +1)/(genTime*roi);
}
```

```
inline void Whitefly::AddLeaving(int sec, int num)
{
  leaving[sec] += num;
}
*/
```

```
/*  WFBAT is a program that is used to consecutively run wf.exe several
 *    times for different parameter values.
 *    It is called with one argument for the number of runs.
 *    The program will read in parameters for each run from a series of
 *    parameter files, numbered as sim1.dat, sim2.dat, simr3.dat, etc.
 •    and/or crop1.dat, crop2.dat, etc.
 *    The program renames sim?.dat files to simpar.dat and crop?.dat file
 *    to croppar.dat which are read by wf.exe.
 */

#include <stdio.h>
#include <io.h>
#include <stdlib.h>
#include <process.h>
#include <string.h>
#include <dos.h>

char **SVector(unsigned long nrl, unsigned long nrh, int ncl, int nch);
void FreeSVector(char **m, unsigned long nrl, unsigned long nrh, int ncl);
void PError(int result, char *string);
void Error(char *str1);

void main(int argc, char *argv[])
{
  int i, result, simFileExist, cropFileExist;
  char numr[10];
  int numRuns = 1;
  char simParFile[] = "simpar.dat";
  char cropParFile[] = "croppar.dat";
  char **simFileRun;
  char **cropFileRun;

  if( argc == 2 ) {
    strcpy(numr, argv[1]);
    numRuns = atoi(numr);
  }

  simFileRun = SVector(1, numRuns, 0, 15);
  cropFileRun = SVector(1, numRuns, 0, 15);

  for(i=1; i<=numRuns; i++) {
    sprintf(simFileRun[i], "sim%d.dat", i);
    if( access(simFileRun[i], 0) == 0 ) {
      simFileExist = 1;
      result = remove(simParFile);
      PError(result, simParFile);
      result = rename(simFileRun[i], simParFile);
      PError(result, simFileRun[i]);
    } else simFileExist = 0;

    sprintf(cropFileRun[i], "crop%d.dat", i);
    if( access(cropFileRun[i], 0) == 0 ) {
      cropFileExist = 1;
      result = remove(cropParFile);
      PError(result, cropParFile);
      result = rename(cropFileRun[i], cropParFile);
      PError(result, cropFileRun[i]);
    } else cropFileExist = 0;

    if( simFileExist || cropFileExist ) {
      result = spawnl(P_WAIT, "wf.exe", NULL);
      PError(result, "Error from spawn.");
    } else {
      printf("Did not find parameter files for run %d\n", i);
    }
  }
```

```
  sound(440);
  delay(1000);
  nosound();

  FreeSVector(simFileRun, 1, numRuns, 0);
  FreeSVector(cropFileRun, 1, numRuns, 0);
}

char **SVector(unsigned long nrl, unsigned long nrh, int ncl, int nch)
{
  unsigned long i;
  int j;
  char **m;

  m = (char **)malloc((unsigned)(nrh-nrl+1)*sizeof(char *));
  if(!m) Error("Allocation failure in SVector()");
  m -= nrl;

  for(i=nrl; i<=nrh; i++) {
    m[i] = (char *)malloc((unsigned)(nch-ncl+1)*sizeof(char));
    if(!m[i]) Error("Allocation failure in SVector()");
    m[i] -= ncl;
  }

  return m;
}

void FreeSVector(char **m, unsigned long nrl, unsigned long nrh, int ncl)
{
  unsigned long i;

  for(i=nrh; i>=nrl; i--)
    free((char*) (m[i] + ncl));

  free((char*)(m + nrl));
}

void PError(int result, char *string)
{
  if(result == -1 ) {
    perror(string);
    exit(1);
  }
}

void Error(char *str)
{
  printf("Run-time error...\n");
  printf("%s\n", str);
  exit(1);
}
```

```
/* File wf.h containes declaration for the Whitefly
 * simulation model, wf.cpp
 */

#define max(a,b)     (((a) > (b)) ? (a) : (b))
#define min(a,b)     (((a) < (b)) ? (a) : (b))

void ReadCropParameters(void);
void CropInit(void);
void SetupCrops(int week);
void WfSim(int week);
void WriteParamFile(void);

void Error(char *str1, char *str2, char *str3);
void *operator new(size_t size);

class Whitefly;
class Crop;

const int maxNumCrops = 25;

extern char defDir[];
extern char outDir[];
extern char outFile[];

extern char cropParFile[30];
extern char simParFile[30];

extern char gCropName[maxNumCrops][20];
extern char gFileName[maxNumCrops][20];
extern int gChange[maxNumCrops];
extern int gWeekPlantStarts[maxNumCrops];
extern int gWeekPlantEnds[maxNumCrops];
extern int gWeekHarvestStarts[maxNumCrops];
extern int gWeekHarvestEnds[maxNumCrops];
extern float gPropAreaChanged[maxNumCrops];
extern float gPropSecUsed[maxNumCrops];
extern float gPropHarvested[maxNumCrops];
extern float gPropWeekHarv[maxNumCrops];
extern float gPropSurvHarv[maxNumCrops];
extern float gWfCapacity[maxNumCrops];
extern float gWfGrowthRate[maxNumCrops];
extern int gSecOut[maxNumCrops];


extern Whitefly *wfly;
extern Crop *crop;
extern int numCrops;
```

```
/* File crop.h containes declaration of
 * Crop class
 */

class Whitefly;

class Crop {
public:
  Crop();
  void Setup(int week);
  int FindSec(int s);
  int FindAllSec(int s);
  int Planted(int week, int cs);
  int SummerPlant(int week, int plant, int harvest);
  int WinterPlant(int week, int plant, int harvest);
  friend class Whitefly;

private:
  static int cnum;
  char name[20];          // Name of crop
  char fname[9];          // Name of crop file
  int change;             // Use default planting or harvesting data or ones listed below
  int weekPlantStarts;
  int weekPlantEnds;
  int weekHarvestStarts;
  int weekHarvestEnds;
  float propAreaChanged;  // Proportion of default acreage of crop in each section
  float propSecUsed;      // Proportion of default number of sections with crop
  float propHarvested;    // Proportion of crop that is harvested
  float propWeekHarv;     // Proportion of crop that is harvested each week (makes sense probably
only for alfalfa)
  float propSurvHarv;     // Proportion of whiteflies that survive crop harvest
  float wfCapacity;       // Maximum wf density crop can support
  float wfGrowthRate;     // Rate of wf growth on crop relative to cotton
  int secOut;           // Print out the acres per section each week?

  int summerCrop;
  float gv;
  float totalAcres;
  float totalWf;

  int numSections;
  int *row;
  int *col;
  float *acres;
  int *weekPlant;
  int *weekHarvest;
  float *numWf;

  int *section;
  int *planted;
};
```

```
/* File sim.h containes declaration of
 * Whitefly class
 */
#include <math.h>

const int HZ = 35;
const int HCOL = 36;
const int HROW = 42;
const int allSections = HCOL*HROW;


class Whitefly {
public:
  Whitefly();
  void Disperse(int week);
  void PopGrow(int week);
  void Mortality(int week);
  void WeeklyInit(void);
  void FixSections(void);
  void WriteOutputFiles(int week);
  void WriteParameterFile(void);
  void SetInitWf(void);
  int NumWeeks(void) { return totalWeeks; }
  float GrowthFunc(float cropGr){ return roi * log(growthFactor*cropGr*(exp(genTime*roi) -1)
+1)/(genTime*roi); }
  void AddLeaving(int r, int c, float num){ leaving[r*HCOL+c] += num; }

  int doSim;        // Run simulation (if don't program will calulate crop characteristics
  float initWfProp; // Proportion of default initial number of whiteflies used

private:
  char outFile[30]; // Base of name for simulation output (name.out), crops (namec.out), and
parameters (namep.out)
  int totalWeeks;    // Total number of weeks simulation run
  float roi;         // Rate of whitefly population increase
  float growthFactor;// Scaling factor for roi
  float genTime;     // Generation time of whiteflies in degree days
  float dispersalProp; // Proportion of wf dispersing from a section that leave the section
  int wkWfSecOut;     // Write weekly num wf per section to output files?
  int secOutFreq;     // Weekly frequency at which section output (if any) is printed to file
  int initNumSec;    // Number of sections in which whiteflies are initially present--use 0 for
default settings
  int *initRow;     // List of rows with initial wfs
  int *initCol;      // List of columns with initial wfs
  float *initNum;     // List of initial num wfs in each of the initial sections
  unsigned rseed;

  float leaving[allSections];   // Array of number of wf leaving each section
  float arriving[allSections];  // Array of number of wf arriving to each
  float grid[HZ][HZ];           // Array of dispersal proportions read from aria files

  int insecticide;  // Boolean: insecticide present = 1, absent = 0
  float icWFM;     // Proportion of whitefly killed by insecticide
  float wfMortIc;    // Mortality of wf each week due to insecticide
  int icResidTime; // Residual time in weeks till insecticide effectiveness equals a value of
residZero
  float residZero;
  int icTime;       // First week larvicide is used
  int iicTime;
  int icNextTime;    // Number of weeks larvicide is used after the previous time used.
  int icLastTime;  // Last week larvicide is used
  char *sprayedCrop[maxNumCrops]; // List of crops that are to be sprayed
  int numSprayedCrops;   // Number of sprayed crops
  float *degd;      // Array of degree days for each week of simulation
};
```

```
/**********************************************************************
 SIMPAR.DAT
**********************************************************************/
Output_files                     def
Do_simulation                    1
Num_weeks_run_simulation         52
Rate_of_increase                 0.00175
Rate_of_growth_factor            2.5
Generation_time-in_deg_days      325.0
Prop-dispersers-leaving-sec      0.25
Prop_change_init_wf              1.0
Weekly_wf_per_sec_output         1
Freq_write_section_output        10
Num_sec_init_wf_(0=all)          0
Rows_Cols_with_init_wf-prefix_with_rc:
rc 16 17 1000
rc 27 19 2000
rc 33 24 500

Random_num_seed                  100

Num_crops_to_spray               0
Mortality_due_insecticide        0.90
ResidTime                        4
ResidZero                        0.01
First_week_sprayed               10
Weeks_between_spraying           10
Last_week_sprayed                40
List_of_crops_to_be_sprayed
cotton
melon

Degree_days
            23,24
    41.36
    19.06
    32.63
    26.75
    31.39
    24.75
    43.77
    66.87
    41.49
    42.32
    76.99
    59.49
    70.71
    85.95
    74.84
    80.84
    88.06
    86.66
    85.90
   104.54
```

90.07
118.45
103.63
121.47
135.69
134.23
142.10
141.74
126.61
133.55
142.43
127.61
115.22
134.34
133.20
135.26
104.68
108.74
108.92
85.14
88.77
78.63
64.94
55.72
67.43
48.96
29.66
33.71
28.78
12.36
12.63
12.63

```
/***********************************************************************
 CROPPAR.DAT
***********************************************************************/

Include_this_crop              1
Name_of_crop                   alfalfa
File_name                      alf
Change_from_default            0
Week_planting_starts           41
Week_planting_ends             41
Week_harvest_starts            30
Week_harvest_ends              30
Prop_area_change               1.0
Prop_sections_used             1.0
Prop_alfalfa_dry               1.0
Prop_harvested_each_week       0.25
Prop_wf_surv_from_harvest      0.8
Whitefly_carrying_capacity     100000.0
Whitefly_growth_rate           0.385
Write_section_acres            0

Include_this_crop              1
Name_of_crop                   asparagus
File_name                      asp
Change_from_default            0
Week_planting_starts           19
Week_planting_ends             19
Week_harvest_starts            40
Week_harvest_ends              43
Prop_area_change               1.0
Prop_sections_used             1.0
Prop_harvested                 1.0
Prop_harvested_each_week       0.0
Prop_wf_surv_from_harvest      1.0
Whitefly_carrying_capacity     100000.0
Whitefly_growth_rate           0.15
Write_section_acres            0

Include_this_crop              1
Name_of_crop                   broccoli
File_name                      broc
Change_from_default            0
Week_planting_starts           39
Week_planting_ends             39
Week_harvest_starts            12
Week_harvest_ends              15
Prop_area_change               1.0
Prop_sections_used             1.0
Prop_harvested                 1.0
Prop_harvested_each_week       0.0
Prop_wf_surv_from_harvest      1.0
Whitefly_carrying_capacity     100000.0
Whitefly_growth_rate           1.08
Write_section_acres            0
```

```
Include_this_crop             1
Name_of_crop                  cabbage
File_name                     cab
Change_from_default           0
Week_planting_starts          39
Week_planting_ends            39
Week_harvest_starts           12
Week_harvest_ends             16
Prop_area_change              1.0
Prop_sections_used            1.0
Prop_harvested                1.0
Prop_harvested_each_week      0.0
Prop_wf_surv_from_harvest     1.0
Whitefly_carrying_capacity    100000.0
Whitefly_growth_rate          1.08
Write_section_acres           0

Include_this_crop             1
Name_of_crop                  carrot
File_name                     car
Change_from_default           0
Week_planting_starts          43
Week_planting_ends            43
Week_harvest_starts           16
Week_harvest_ends             19
Prop_area_change              1.0
Prop_sections_used            1.0
Prop_harvested                1.0
Prop_harvested_each_week      0.0
Prop_wf_surv_from_harvest     1.0
Whitefly_carrying_capacity    100000.0
Whitefly_growth_rate          0.2
Write_section_acres           0

Include_this_crop             1
Name_of_crop                  cauliflower
File_name                     caul
Change_from_default           0
Week_planting_starts          39
Week_planting_ends            39
Week_harvest_starts           12
Week_harvest_ends             15
Prop_area_change              1.0
Prop_sections_used            1.0
Prop_harvested                1.0
Prop_harvested_each_week      0.0
Prop_wf_surv_from_harvest     1.0
Whitefly_carrying_capacity    100000.0
Whitefly_growth_rate          1.08
Write_section_acres           0

Include_this_crop             1
Name_of_crop                  celery
```

```
File_name                    cel
Change_from_default          0
Week_planting_starts         43
Week_planting_ends           43
Week_harvest_starts          9
Week_harvest_ends            11
Prop_area_change             1.0
Prop_sections_used           1.0
Prop_harvested               1.0
Prop_harvested_each_week     0.0
Prop_wf_surv_from_harvest    1.0
Whitefly_carrying_capacity   100000.0
Whitefly_growth_rate         0.015
Write_section_acres          0

Include_this_crop            1
Name_of_crop                 cereal
File_name                    cer
Change_from_default          0
Week_planting_starts         7
Week_planting_ends           7
Week_harvest_starts          28
Week_harvest_ends            31
Prop_area_change             1.0
Prop_sections_used           1.0
Prop_harvested               1.0
Prop_harvested_each_week     0.0
Prop_wf_surv_from_harvest    1.0
Whitefly_carrying_capacity   100000.0
Whitefly_growth_rate         0.015
Write_section_acres          0

Include_this_crop            1
Name_of_crop                 corn
File_name                    corn
Change_from_default          0
Week_planting_starts         11
Week_planting_ends           11
Week_harvest_starts          20
Week_harvest_ends            22
Prop_area_change             1.0
Prop_sections_used           1.0
Prop_harvested               1.0
Prop_harvested_each_week     0.0
Prop_wf_surv_from_harvest    1.0
Whitefly_carrying_capacity   100000.0
Whitefly_growth_rate         0.015
Write_section_acres          0

Include_this_crop            1
Name_of_crop                 cotton
File_name                    cot
Change_from_default          0
Week_planting_starts         15
```

```
Week_planting_ends           15
Week_harvest_starts          36
Week_harvest_ends            39
Prop_area_change             1.0
Prop_sections_used           1.0
Prop_harvested               1.0
Prop_harvested_each_week     0.0
Prop_wf_surv_from_harvest    1.0
Whitefly_carrying_capacity   100000.0
Whitefly_growth_rate         1.0
Write_section_acres          0

Include_this_crop            1
Name_of_crop                 cucumber
File_name                    cuc
Change_from_default          0
Week_planting_starts         39
Week_planting_ends           39
Week_harvest_starts          49
Week_harvest_ends            50
Prop_area_change             1.0
Prop_sections_used           1.0
Prop_harvested               1.0
Prop_harvested_each_week     0.0
Prop_wf_surv_from_harvest    1.0
Whitefly_carrying_capacity   100000.0
Whitefly_growth_rate         1.0
Write_section_acres          0

Include_this_crop            1
Name_of_crop                 garlic
File_name                    gar
Change_from_default          0
Week_planting_starts         51
Week_planting_ends           51
Week_harvest_starts          22
Week_harvest_ends            23
Prop_area_change             1.0
Prop_sections_used           1.0
Prop_harvested               1.0
Prop_harvested_each_week     0.0
Prop_wf_surv_from_harvest    1.0
Whitefly_carrying_capacity   100000.0
Whitefly_growth_rate         0.015
Write_section_acres          0

Include_this_crop            1
Name_of_crop                 lettuce
File_name                    let
Change_from_default          0
Week_planting_starts         51
Week_planting_ends           51
Week_harvest_starts          8
Week_harvest_ends            11
```

```
Prop_area_change            1.0
Prop_sections_used          1.0
Prop_harvested              1.0
Prop_harvested_each_week    0.0
Prop_wf_surv_from_harvest   1.0
Whitefly_carrying_capacity  100000.0
Whitefly_growth_rate        0.2
Write_section_acres         0

Include_this_crop           1
Name_of_crop                spring_melon
File_name                   smel
Change_from_default         0
Week_plant_starts           7
Week_plant_ends             7
Week_harvest_starts         26
Week_harvest_endss          28
Prop_area_change            1.0
Prop_sections_used          1.0
Prop_harvested              1.0
Prop_harvested_each_week    0.0
Prop_wf_surv_from_harvest   1.0
Whitefly_carrying_capacity  100000.0
Whitefly_growth_rate        1.54
Write_section_acres         0

Include_this_crop           1
Name_of_crop                fall_melon
File_name                   fmel
Change_from_default         0
Week_plant_starts           35
Week_plant_ends             35
Week_harvest_starts         51
Week_harvest_endss          51
Prop_area_change            1.0
Prop_sections_used          1.0
Prop_harvested              1.0
Prop_harvested_each_week    0.0
Prop_wf_surv_from_harvest   1.0
Whitefly_carrying_capacity  100000.0
Whitefly_growth_rate        1.54
Write_section_acres         0

Include_this_crop           1
Name_of_crop                onion
File_name                   onion
Change_from_default         0
Week_planting_starts        51
Week_planting_ends          51
Week_harvest_starts         24
Week_harvest_ends           27
Prop_area_change            1.0
Prop_sections_used          1.0
Prop_harvested              1.0
```

```
Prop_harvested_each_week      0.0
Prop_wf_surv_from_harvest     1.0
Whitefly_carrying_capacity    100000.0
Whitefly_growth_rate          0.015
Write_section_acres           0

Include_this_crop             1
Name_of_crop                  sorghum
File_name                     sor
Change_from_default           0
Week_planting_starts          27
Week_planting_ends            27
Week_harvest_starts           41
Week_harvest_ends             43
Prop_area_change              1.0
Prop_sections_used            1.0
Prop_harvested                1.0
Prop_harvested_each_week      0.0
Prop_wf_surv_from_harvest     1.0
Whitefly_carrying_capacity    100000.0
Whitefly_growth_rate          0.015
Write_section_acres           0

Include_this_crop             1
Name_of_crop                  squash
File_name                     squ
Change_from_default           0
Week_planting_starts          43
Week_planting_ends            43
Week_harvest_starts           20
Week_harvest_ends             23
Prop_area_change              1.0
Prop_sections_used            1.0
Prop_harvested                1.0
Prop_harvested_each_week      0.0
Prop_wf_surv_from_harvest     1.0
Whitefly_carrying_capacity    100000.0
Whitefly_growth_rate          1.0
Write_section_acres           0

Include_this_crop             1
Name_of_crop                  sugarbeet
File_name                     sug
Change_from_default           0
Week_planting_starts          0
Week_planting_ends            0
Week_harvest_starts           0
Week_harvest_ends             0
Prop_area_change              1.0
Prop_sections_used            1.0
Prop_harvested                1.0
Prop_harvested_each_week      0.0
Prop_wf_surv_from_harvest     1.0
Whitefly_carrying_capacity    100000.0
```

Whitefly_growth_rate	0.2
Write_section_acres	0

Include_this_crop	1
Name_of_crop	tomato
File_name	tom
Change_from_default	0
Week_planting_starts	7
Week_planting_ends	7
Week_harvest_starts	27
Week_harvest_ends	27
Prop_area_change	1.0
Prop_sections_used	1.0
Prop_harvested	1.0
Prop_harvested_each_week	0.0
Prop_wf_surv_from_harvest	1.0
Whitefly_carrying_capacity	100000.0
Whitefly_growth_rate	0.39
Write_section_acres	0

```
/*  cab.crp: the cabbage crop file
    */

 3  22   71  39  12   1760
 4  22    7  39  14    160
 4  24   13  39  14    320
 8  17    7  39  16    160
 8  18    7  39  13    160
 8  28    7  39  15    160
 9  13    7  39  14    160
 9  16    7  39  13    160
10  16   13  39  14   1120
12  17   13  39  14    960
12  26   13  39  14    640
13   9   26  39  12    640
14  15   13  39  14    320
14  18   19  39  14    480
14  23    7  39  14    160
15  17   19  39  13    480
15  19   19  39  14    640
16  26   13  39  13    480
16  31   26  39  13    640
17  22   26  39  13    640
18  25   13  39  13    320
21  16    7  39  13    160
21  32    7  39  14    160
21  33   13  39  13    320
23  24    7  39  13    160
24  18    7  39  15    160
24  27   13  39  14    320
25  17   26  39  14    640
25  19   19  39  15    480
25  20    7  39  13    160
25  21   39  39  15    960
25  25   13  39  15    640
26  24    7  39  14    160
26  26   32  39  13    800
27  22    7  39  13    160
27  24    7  39  13    160
29  15   32  39  13    800
31  22   13  39  14    320
32  15    7  39  13    160
33  22   51  39  13   1280
33  24    7  39  14    160
33  25   26  39  13    640
34  18   13  39  14    320
34  23   39  39  12    960
35   8    7  39  14    160
35  17   13  39  12    320
35  25   13  39  14    320
35  33    7  39  13    160
36  25    7  39  13    160
```

```
/* aria.01: file of dispersion factors for week 1
 */
 .55E-06 .71E-06 .00E+00 .61E-06 .78E-06 .00E+00 .43E-06 .51E-06 .00E+00 .20E-
06 .20E-06 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .40E-06 .40E-06 .00E+00 .10E-05 .86E-
06 .00E+00 .16E-05 .12E-05 .00E+00 .14E-05 .11E-05
 .71E-06 .64E-06 .67E-06 .72E-06 .74E-06 .90E-06 .65E-06 .57E-06 .41E-06 .34E-
06 .28E-06 .13E-06 .84E-07 .21E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .41E-07 .17E-06 .26E-06 .57E-06 .69E-06 .83E-06 .11E-05 .13E-
05 .18E-05 .15E-05 .14E-05 .13E-05 .13E-05 .14E-05
 .00E+00 .67E-06 .74E-06 .81E-06 .83E-06 .10E-05 .81E-06 .73E-06 .56E-06 .44E-
06 .40E-06 .20E-06 .12E-06 .39E-07 .18E-07 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .36E-07 .78E-07 .25E-06 .40E-06 .80E-06 .88E-06 .11E-05 .15E-05 .16E-
05 .20E-05 .17E-05 .16E-05 .15E-05 .13E-05 .00E+00
 .61E-06 .72E-06 .81E-06 .99E-06 .10E-05 .14E-05 .12E-05 .11E-05 .94E-06 .78E-
06 .75E-06 .37E-06 .28E-06 .11E-06 .69E-07 .14E-07 .00E+00 .00E+00 .00E+00
.28E-07 .14E-06 .21E-06 .56E-06 .75E-06 .15E-05 .16E-05 .19E-05 .22E-05 .23E-
05 .28E-05 .21E-05 .20E-05 .16E-05 .14E-05 .12E-05
 .78E-06 .74E-06 .83E-06 .10E-05 .12E-05 .16E-05 .13E-05 .14E-05 .12E-05 .11E-
05 .94E-06 .48E-06 .41E-06 .18E-06 .97E-07 .28E-07 .14E-07 .00E+00 .28E-07
.57E-07 .19E-06 .35E-06 .81E-06 .96E-06 .19E-05 .21E-05 .25E-05 .27E-05 .27E-
05 .32E-05 .23E-05 .21E-05 .17E-05 .15E-05 .16E-05
 .00E+00 .90E-06 .10E-05 .14E-05 .16E-05 .24E-05 .21E-05 .22E-05 .21E-05 .20E-
05 .20E-05 .11E-05 .95E-06 .39E-06 .24E-06 .43E-07 .14E-07 .00E+00 .28E-07
.85E-07 .47E-06 .77E-06 .19E-05 .23E-05 .39E-05 .40E-05 .42E-05 .44E-05 .42E-
05 .48E-05 .32E-05 .28E-05 .20E-05 .18E-05 .00E+00
 .43E-06 .65E-06 .81E-06 .12E-05 .13E-05 .21E-05 .20E-05 .21E-05 .23E-05 .23E-
05 .27E-05 .16E-05 .13E-05 .45E-06 .32E-06 .92E-07 .14E-07 .00E+00 .28E-07
.18E-06 .64E-06 .90E-06 .25E-05 .33E-05 .53E-05 .46E-05 .46E-05 .42E-05 .40E-
05 .42E-05 .27E-05 .23E-05 .16E-05 .13E-05 .86E-06
 .51E-06 .57E-06 .73E-06 .11E-05 .14E-05 .22E-05 .21E-05 .25E-05 .28E-05 .28E-
05 .33E-05 .24E-05 .20E-05 .87E-06 .42E-06 .14E-06 .63E-07 .00E+00 .13E-06
.28E-06 .84E-06 .17E-05 .41E-05 .47E-05 .66E-05 .57E-05 .57E-05 .49E-05 .42E-
05 .44E-05 .27E-05 .22E-05 .15E-05 .11E-05 .10E-05
 .00E+00 .41E-06 .56E-06 .94E-06 .12E-05 .21E-05 .23E-05 .28E-05 .40E-05 .43E-
05 .55E-05 .43E-05 .47E-05 .26E-05 .17E-05 .31E-06 .19E-06 .25E-06 .38E-06
.63E-06 .33E-05 .53E-05 .93E-05 .87E-05 .11E-04 .86E-05 .80E-05 .57E-05 .46E-
05 .42E-05 .25E-05 .19E-05 .11E-05 .83E-06 .00E+00
 .20E-06 .41E-06 .50E-06 .84E-06 .11E-05 .21E-05 .23E-05 .29E-05 .43E-05 .52E-
05 .75E-05 .60E-05 .63E-05 .41E-05 .31E-05 .12E-05 .28E-06 .43E-06 .51E-06
.24E-05 .62E-05 .82E-05 .13E-04 .12E-04 .15E-04 .10E-04 .86E-05 .57E-05 .46E-
05 .40E-05 .21E-05 .16E-05 .88E-06 .69E-06 .40E-06
 .40E-06 .41E-06 .53E-06 .86E-06 .11E-05 .21E-05 .28E-05 .34E-05 .55E-05 .75E-
05 .12E-04 .11E-04 .14E-04 .10E-04 .92E-05 .43E-05 .25E-05 .44E-05 .48E-05
.85E-05 .18E-04 .21E-04 .29E-04 .23E-04 .25E-04 .15E-04 .11E-04 .66E-05 .53E-
05 .39E-05 .19E-05 .15E-05 .80E-06 .57E-06 .40E-06
 .20E-06 .46E-06 .54E-06 .71E-06 .82E-06 .15E-05 .19E-05 .26E-05 .46E-05 .61E-
05 .11E-04 .14E-04 .20E-04 .15E-04 .13E-04 .66E-05 .49E-05 .79E-05 .96E-05
.13E-04 .25E-04 .31E-04 .40E-04 .27E-04 .23E-04 .12E-04 .87E-05 .47E-05 .33E-
05 .23E-05 .96E-06 .75E-06 .40E-06 .26E-06 .00E+00
 .59E-06 .56E-06 .61E-06 .78E-06 .91E-06 .14E-05 .17E-05 .25E-05 .51E-05 .67E-
05 .15E-04 .20E-04 .40E-04 .43E-04 .48E-04 .32E-04 .24E-04 .22E-04 .41E-04
.61E-04 .95E-04 .86E-04 .80E-04 .40E-04 .29E-04 .13E-04 .93E-05 .41E-05 .25E-
05 .19E-05 .81E-06 .56E-06 .25E-06 .17E-06 .00E+00
```

.59E-06 .11E-05 .12E-05 .15E-05 .16E-05 .21E-05 .23E-05 .31E-05 .50E-05 .69E-05 .13E-04 .19E-04 .47E-04 .62E-04 .67E-04 .11E-03 .97E-04 .23E-03 .18E-03 .22E-03 .13E-03 .12E-03 .86E-04 .31E-04 .21E-04 .82E-05 .53E-05 .17E-05 .90E-06 .77E-06 .35E-06 .21E-06 .78E-07 .41E-07 .00E+00
.19E-05 .16E-05 .18E-05 .22E-05 .25E-05 .32E-05 .36E-05 .45E-05 .63E-05 .87E-05 .15E-04 .20E-04 .55E-04 .71E-04 .15E-03 .26E-03 .24E-03 .44E-03 .46E-03 .50E-03 .28E-03 .13E-03 .95E-04 .25E-04 .18E-04 .62E-05 .33E-05 .84E-06 .64E-06 .47E-06 .19E-06 .14E-06 .36E-07 .00E+00 .00E+00
.20E-05 .24E-05 .27E-05 .34E-05 .40E-05 .52E-05 .61E-05 .83E-05 .10E-04 .15E-04 .22E-04 .36E-04 .69E-04 .22E-03 .42E-03 .12E-01 .23E-01 .32E-01 .21E-01 .94E-02 .50E-03 .22E-03 .60E-04 .13E-04 .85E-05 .24E-05 .63E-06 .28E-06 .18E-06 .85E-07 .57E-07 .28E-07 .00E+00 .00E+00 .00E+00
.26E-05 .28E-05 .32E-05 .41E-05 .48E-05 .64E-05 .77E-05 .11E-04 .14E-04 .21E-04 .29E-04 .55E-04 .91E-04 .31E-03 .31E-02 .23E-01 .60E-01 .74E-01 .57E-01 .20E-01 .30E-02 .18E-03 .39E-04 .96E-05 .48E-05 .49E-06 .38E-06 .13E-06 .28E-07 .28E-07 .28E-07 .00E+00 .00E+00 .00E+00 .00E+00
.27E-05 .30E-05 .35E-05 .45E-05 .54E-05 .74E-05 .90E-05 .13E-04 .17E-04 .26E-04 .36E-04 .71E-04 .11E-03 .40E-03 .58E-02 .34E-01 .76E-01 .00E+00 .72E-01 .30E-01 .54E-02 .16E-03 .14E-04 .53E-05 .29E-05 .28E-06 .17E-06 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.26E-05 .28E-05 .32E-05 .41E-05 .48E-05 .64E-05 .77E-05 .11E-04 .14E-04 .21E-04 .27E-04 .50E-04 .75E-04 .23E-03 .80E-02 .27E-01 .60E-01 .71E-01 .60E-01 .27E-01 .76E-02 .68E-05 .46E-05 .11E-06 .11E-06 .36E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.20E-05 .24E-05 .27E-05 .34E-05 .39E-05 .51E-05 .60E-05 .81E-05 .98E-05 .14E-04 .17E-04 .30E-04 .42E-04 .11E-03 .53E-02 .16E-01 .24E-01 .30E-01 .24E-01 .16E-01 .51E-02 .53E-05 .43E-05 .10E-05 .36E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.19E-05 .16E-05 .18E-05 .21E-05 .24E-05 .29E-05 .33E-05 .41E-05 .46E-05 .56E-05 .62E-05 .94E-05 .12E-04 .14E-04 .27E-02 .53E-02 .80E-02 .80E-02 .80E-02 .53E-02 .27E-02 .61E-05 .41E-05 .20E-05 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.59E-06 .11E-05 .12E-05 .13E-05 .15E-05 .17E-05 .18E-05 .22E-05 .24E-05 .28E-05 .30E-05 .65E-05 .98E-05 .13E-04 .20E-03 .39E-03 .57E-03 .57E-03 .57E-03 .38E-03 .20E-03 .91E-05 .61E-05 .30E-05 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.59E-06 .47E-06 .49E-06 .50E-06 .50E-06 .49E-06 .48E-06 .44E-06 .43E-06 .37E-06 .34E-06 .34E-05 .65E-05 .95E-05 .14E-03 .28E-03 .41E-03 .41E-03 .41E-03 .28E-03 .14E-03 .94E-05 .63E-05 .32E-05 .48E-07 .32E-07 .16E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.20E-06 ,33E-06 .34E-06 .34E-06 .34E-06 .32E-06 .31E-06 .26E-06 .26E-06 .23E-06 .24E-06 .34E-05 .66E-05 .98E-05 .89E-04 .17E-03 .25E-03 .25E-03 .25E-03 .17E-03 .89E-04 .97E-05 .65E-05 .33E-05 .97E-07 .64E-07 .32E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.20E-06 .13E-06 .13E-06 .12E-06 .11E-06 .10E-06 .93E-07 .57E-07 .77E-07 .97E-07 .15E-06 .34E-05 .67E-05 .10E-04 .36E-04 .62E-04 .88E-04 .88E-04 .88E-04 .62E-04 .36E-04 .10E-04 .67E-05 .34E-05 .15E-06 .97E-07 .48E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .65E-07 .63E-07 .60E-07 .57E-07 .50E-07 .47E-07 .28E-07 .83E-07 .14E-06 .21E-06 .32E-05 .62E-05 .91E-05 .29E-04 .48E-04 .68E-04 .68E-04 .68E-04 .48E-04 .29E-04 .91E-05 .62E-05 .32E-05 .21E-06 .14E-06 .69E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .90E-07 .18E-06 .27E-06 .29E-05 .56E-05 .83E-05 .21E-04 .35E-04 .48E-04 .48E-04 .48E-04

.35E-04 .21E-04 .83E-05 .56E-05 .29E-05 .27E-06 .18E-06 .90E-07 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .11E-06 .22E-06 .33E-06 .27E-05 .50E-05 .74E-05 .14E-04 .21E-04 .28E-04 .28E-04 .28E-04
.21E-04 .14E-04 .74E-05 .50E-05 .27E-05 .33E-06 .22E-06 .11E-06 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .13E-06 .26E-06 .39E-06 .25E-05 .45E-05 .66E-05 .12E-04 .17E-04 .23E-04 .23E-04 .23E-04
.17E-04 .12E-04 .66E-05 .45E-05 .25E-05 .39E-06 .26E-06 .13E-06 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .15E-06 .30E-06 .45E-06 .23E-05 .41E-05 .59E-05 .98E-05 .14E-04 .18E-04 .18E-04 .18E-04
.14E-04 .98E-05 .59E-05 .41E-05 .23E-05 .45E-06 .30E-06 .15E-06 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .17E-06 .34E-06 .51E-06 .21E-05 .36E-05 .52E-05 .76E-05 .10E-04 .12E-04 .12E-04 .12E-04
.10E-04 .76E-05 .52E-05 .36E-05 .21E-05 .51E-06 .34E-06 .17E-06 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .13E-07 .25E-07 .38E-07 .20E-06 .36E-06 .53E-06 .19E-05 .33E-05 .46E-05 .66E-05 .85E-05 .10E-04 .10E-04 .10E-04
.85E-05 .66E-05 .46E-05 .33E-05 .19E-05 .53E-06 .36E-06 .20E-06 .38E-07 .25E-07 .13E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .25E-07 .51E-07 .76E-07 .23E-06 .39E-06 .55E-06 .17E-05 .29E-05 .41E-05 .55E-05 .70E-05 .85E-05 .85E-05 .85E-05
.70E-05 .55E-05 .41E-05 .29E-05 .17E-05 .55E-06 .39E-06 .23E-06 .76E-07 .51E-07 .25E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .38E-07 .76E-07 .11E-06 .27E-06 .42E-06 .57E-06 .16E-05 .25E-05 .35E-05 .45E-05 .55E-05 .65E-05 .65E-05 .65E-05
.55E-05 .45E-05 .35E-05 .25E-05 .16E-05 .57E-06 .42E-06 .27E-06 .11E-06 .76E-07 .38E-07 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00
.00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .11E-06 .11E-06 .11E-06 .57E-06 .57E-06 .57E-06 .35E-05 .35E-05 .35E-05 .65E-05 .65E-05 .65E-05 .65E-05
.65E-05 .35E-05 .35E-05 .35E-05 .57E-06 .57E-06 .57E-06 .11E-06 .11E-06 .11E-06 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00 .00E+00

Appendix 2

Dispersion Factors for the Whitefly Model

Basis

Previous work conducted for the Pesticide Regulatory Display Model ('Telone model') had produced sets of tabulated values, which when multiplied by pounds applied and divided by wind speed, produced an air concentration index for a 2 meter height (Johnson 1991ab). These indices were devised at the scale of a section (1 square mile) and were based on the gaussian plume model. They provided a framework for setting dispersal factors for the silverleaf whitefly (SLWP) given that SLWP have been reported as moving with the wind, at wind speed (Mike Pitcairn, personal communication) and that deposition of particulate matter may be proportional to air concentration (Zanetti 1990). In the absence of any other information, the gaussian plume model, with the 2 meter height index was choosen as a basis for SLWP dispersal.

Technical Implementation

To minimize on-line computational effort during the running of the model and to modularize this section of the model, 52 tables of dispersion factors were produced. Each table provides dispersion factors which summarize 1 week of 1990 weather data from Brawley, CA (Imperial County). The wind data provided consisted of a daily summary of average wind direction and average speed (Table 26). The wind direction is the direction from which the wind originates. The wind speed is in miles per hour. This data is from station Brawley.A (CAIMAAC2).

In the original work with Telone to develop section factors, a series of 18 11x21 tables were produced. Each table provided factors for an 11x21 mile grid, assuming that the source section

Table 26. Daily wind summary for station CAIMAAC2, located in Brawley, California. Records marked with an x were interpolated from surrounding values. Successive days go down the columns. Direction is direction wind is blowing from.

SE 4	SW 3	W 6	S 6	NE 5	SE 3	SE 3
SW 5	W 3	SW 4	W 7	NE 5	E 3	E 3
N 6	SE 4	SW 4	W 5	NE 5	SE 4	E 3
E 4	E 3	S 5	SW 6	NE 5	SW 5	SE 3
N 6	SE 4	SW 4	W 9	S 6	SW 4	SE 4
SW 4	E 3	SW 5	W 9	SE 5	SE 4	W 7
E 3	E 3	W 10	W 9	SE 4	E 6	NW 3
SE 4	SE 3	W 12	SW 4	SE 4	NW 6	N 4
NW 4	W 5	W 6	SE 4	SE 5	E 3	SE 4
E 3	W 8	W 6	S 4	SE 8	SE 3	E 3
E 3	W 14	W 6 X	SW 4	NE 5	E 3	S 4
E 4	NW 5	W 6 X	SW 4	S 6	SE 3	W 13
SW 6	E 4	W 9 X	S 4	SE 7	W 5	NE 4
SE 5	E 3	SW 9 X	SE 4	SE 5	NW 4	NW 4
E 3	SE 4	W 9	SE 5	SE 4	N 7	W 4
SW 6	SW 8	SW 5	SE 4	SE 6	NE 4	SE 3
W 5	W 9	S 4	W 5	SE 5	SE 3	E 3
W 5	W 9	W 5	NW 5	W 5	E 3	SE 3
W 5 X	W 5	NE 5	W 5	W 4	E 3	SW 3
W 5 X	NW 4	S 5	SE 6	SW 3	SE 4	W 3
SW 4 X	NE 5	SE 8	SE 6	S 4	SE 4	E 3
N 6 X	N 4	SW 6	SE 5	SW 4	SE 5	NE 5
SW 4	W 4	S 4	W 6	SW 6	SE 5	S 3
N 6	W 4	W 13	S 7	W 5	SE 3	SW 3
W 3	SW 4	W 8	SE 5	SE 4	SE 4	E 3
SE 3	SE 3	S 4	SE 4	E 4	W 9	E 3
N 7	S 4	SE 5	SE 4	SE 4	N 6	E 3
W 4	SE 5	W 6	SE 8	SW 4	NE 3	SE 3
E 4	SE 4	W 7	SE 9	SE 4	E 2	NW 4
SW 5	SE 5	SW 4	S 7	SE 5	SE 2	W 4
W 6	SE 4	W 7	S 5	SE 8	E 2	SE 3
W 9	SE 3	W 7	SE 5	SE 5	E 2 X	W 5
NW 7	W 9	W 13	SE 5	SE 8	E 3	E 4
E 3	W 13	W 5	S 5	NW 4	SE 3	SE 4
W 5	W 7	SW 4	SE 6	SE 5	SE 2	SW 9
SE 4	SE 5	NW 6	SE 7	SW 3	E 2	W 13
E 3	SE 4	W 8	SE 8	W 3	SE 4	N 7
SE 3	SE 4	W 6	SE 5	S 3	SE 4	N 6
NW 4	S 3	W 5	S 4	SE 4	W 7	NE 7
S 3	SW 4	W 7	SW 4	SE 3	W 9	W 4
SW 3	NW 5	W 10	SE 7	SE 4	N 8	E 4
SW 3	W 4	W 14	SE12	SE 7	NW 7	W 4
W 5	SW 3	W 5	SE 6	SE 6	SW 4	NW 4
W 12	S 5	W 7	SE 5	SE 6	SE 4	SE 5
W 11	W 10	W 8	SE10	SE 7	N 7	NW 6
N 4	SW 4	W 6	SE10	SE 5	N 7	NW 6
S 6	W 4	W 4	SW 6	W 4	SE 4 X	
SE 6	SW 3	SW 4	SW 4	W 6	N 7 X	
W 10	S 3	S 4	SW 4	W 4	N 7 X	
W 7	SE 5	SE 7	SE 4	SE 5	SE 3 X	
E 4	SE 4	SE 5	SE 4	SE 6	SE 3 X	
NW 5	W 5	E 4	SE 6	NW 3	E 3 X	
NE 5	W 9	S 4	SE 6	SE 4	SE 3	

was in the bottom row, center (middle, south). There were 6 stability conditions (A-F) and 3 methods of summarizing the factors within a section (arithmetic average of positives, geometric average of positives, maximum value). For the current work, the table utilizing stability condition A, most unstable, and the arithmetic method of summarizing, were chosen. The most unstable condition was chosen because it provided the widest dispersal, which seemed appropriate given that the wind speed and direction were based on the mean over an entire day. To reflect wind direction, the 11x21 factor table was rotated to reflect the daily average direction from which the wind originated. If the wind originated from the south, no rotation was necessary. If the wind originated from the southeast, the table was rotated by 45 degrees, counterclockwise.

To reflect wind speed, the original table, which was calculated based on 1 meter/second (about 2 miles/hour) was expanded, if necessary. The daily average windspeed was divided by 2 and this result truncated to an integer. Next, the 11x21 table was lengthened and widened by this factor, duplicating the values in the original table. However, the tabulated values were each divided by the square of the expansion factor, since according to the gaussian plume model, air concentrations decrease inversely with increased air speed in the single dimension downwind direction.

After expansion and rotation, the resulting table of factors was added to the accumulating weekly factor table. These calculations were performed utilizing a 51 x 51 size array. After adding 7 successive days of factors and because some indices do not line up well after a rotation by 45 degrees, a smoothing operater, consisting of averaging 9 adjacent points to estimate the center point, was applied. Following smoothing, the center 35 x 35 portion of the 51 x 51 array was extracted. Finally, the source section was set to 0 and the entire table normalized to sum up to 1.

Verification

Numerical checking was performed on rotation, smoothing, addition, and normalization. Qualitative checking was performed by examination of tables with respect to wind direction.

Programs

Program WKDSP is the main program. Array TAB(21,11) contains the original 11x21 table of factors. Array XPAND(51,51) is the work space to rotate and expand the table of factors for each day. Array SPACE(51,51) is the work space to accumulate the daily factors for a weekly total. Array FINAL(35,35) extracts the center portion of SPACE(51,51) for normalization and printout.

Lines 112-114 limit the expansion factor to an integer between 1 and 5. Rotation is performed in lines 131-153 with a call to subroutine ROTE and adjustments to the indices. Smoothing occurs with a call to subroutine SMOOTH at line 192. Normalization occurs at line 203 with a call to subroutine AVGNOR. The output file is written with the top being north and left is west. The source section is at the center at 18,18. The filenames in this version are of the form TESTARIA.nn, where nn=01,02,03,...,52. There is 1 file for each of 52 weeks. The last day, 365 is not used.

References

Johnson, Bruce R. 1991a. A technique based on the gaussian dispersion model for obtaining crude estimates for air concentrations from Telone applications. Memo to Ronald J. Oshima, August 6, 1991. State of California, Department of Pesticide Regulation. @1470

Johnson, Bruce R. 1991b. Updated flux factor and section tables for Telone to be used in the Pesticide Regulatory Display Model and correction of error. Memorandum to Ron Oshima, Branch Chief, Environmenal Monitoring and Pest Management, Department of Pesticide Regulation, October 2, 1991. @1471 1,3-D gaussian plume model

Zannetti, P. (Paolo). 1990. Air pollution modeling : theories, computational methods, and available software. Southampton ; Boston : Computational Mechanics Publications ; New York : Van Nostrand Reinhold, [c1990].

Table 27. Factors in ARIAVG.A. Each factor corresponds to a 1 square mile section. These factors based on gaussian plume model represent arithmetic average of 21x21 point estimates for that section.

.00E+00	.00E+00	.00E+00	.00E+00	.10E-09	.11E-09	.10E-09	.00E+00	.00E+00	.00E+00	.00E+00
.00E+00	.00E+00	.00E+00	.10E-09	.11E-09	.12E-09	.11E-09	.10E-09	.00E+00	.00E+00	.00E+00
.00E+00	.00E+00	.00E+00	.11E-09	.13E-09	.14E-09	.13E-09	.11E-09	.00E+00	.00E+00	.00E+00
.00E+00	.00E+00	.10E-09	.12E-09	.16E-09	.17E-09	.16E-09	.12E-09	.10E-09	.00E+00	.00E+00
.00E+00	.00E+00	.11E-09	.14E-09	.18E-09	.20E-09	.18E-09	.14E-09	.11E-09	.00E+00	.00E+00
.00E+00	.00E+00	.12E-09	.17E-09	.22E-09	.24E-09	.22E-09	.17E-09	.12E-09	.00E+00	.00E+00
.00E+00	.00E+00	.13E-09	.19E-09	.27E-09	.30E-09	.27E-09	.19E-09	.13E-09	.00E+00	.00E+00
.00E+00	.00E+00	.13E-09	.23E-09	.33E-09	.37E-09	.33E-09	.23E-09	.13E-09	.00E+00	.00E+00
.00E+00	.00E+00	.14E-09	.27E-09	.41E-09	.47E-09	.41E-09	.27E-09	.14E-09	.00E+00	.00E+00
.00E+00	.00E+00	.15E-09	.32E-09	.52E-09	.61E-09	.52E-09	.32E-09	.15E-09	.00E+00	.00E+00
.00E+00	.00E+00	.16E-09	.37E-09	.67E-09	.81E-09	.67E-09	.37E-09	.16E-09	.00E+00	.00E+00
.00E+00	.00E+00	.17E-09	.44E-09	.88E-09	.11E-08	.88E-09	.44E-09	.17E-09	.00E+00	.00E+00
.00E+00	.00E+00	.17E-09	.51E-09	.12E-08	.16E-08	.12E-08	.51E-09	.17E-09	.00E+00	.00E+00
.00E+00	.00E+00	.16E-09	.56E-09	.16E-08	.23E-08	.16E-08	.56E-09	.16E-09	.00E+00	.00E+00
.00E+00	.00E+00	.14E-09	.58E-09	.23E-08	.37E-08	.23E-08	.58E-09	.14E-09	.00E+00	.00E+00
.00E+00	.00E+00	.11E-09	.55E-09	.34E-08	.63E-08	.34E-08	.55E-09	.11E-09	.00E+00	.00E+00
.00E+00	.00E+00	.00E+00	.49E-09	.50E-08	.12E-07	.50E-08	.49E-09	.00E+00	.00E+00	.00E+00
.00E+00	.00E+00	.00E+00	.32E-09	.71E-08	.27E-07	.71E-08	.32E-09	.00E+00	.00E+00	.00E+00
.00E+00	.00E+00	.00E+00	.14E-09	.97E-08	.85E-07	.97E-08	.14E-09	.00E+00	.00E+00	.00E+00
.00E+00	.00E+00	.00E+00	.00E+00	.88E-08	.55E-06	.88E-08	.00E+00	.00E+00	.00E+00	.00E+00
.00E+00	.00E+00	.00E+00	.00E+00	.00E+00	.22E-04	.00E+00	.00E+00	.00E+00	.00E+00	.00E+00

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 4.10

```
      PROGRAM WKDSP
C$debug:
C$LARGE:SPACE,XPAND,TAB,FINAL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C THIS PROGRAM READS A FILE CONTAINING DAILY AVERAGE WIND
C SPEED AND DIRECTION, AND USING THE WIND DIRECTION INFORMATION
C IT ADDS TOGETHER A TABLE OF FACTORS (THE FACTORS TABLES FROM
C TELONE MODELING) FOR ONE WEEK, THEN NORMALIZES THE FACTORS
C SO THAT THEY ADD UP TO ONE.  IT MAKES THE TABLE POINT IN
C THE DIRECTION OF THE WIND.  IE. THIS PROGRAM A S S U M E S
C THAT 'N' MEANS THE WIND IS BLOWING FROM THE NORTH.
C                              FROM
C                              FROM
C THIS PROGRAM TAKES CENTER 35X35 SQUARE OUT FROM 51X51 SQUARE
C AND NORMALIZES THAT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT INTEGER(A-Z)
      CHARACTER*2 CDI(8),DIRECT,WEEKC
      REAL DEGS(8),ANGLE,X,Y,WSPEED,XFACT,XF2
      DOUBLE PRECISION TAB(21,11)
      DOUBLE PRECISION SPACE(51,51),XPAND(51,51),FINAL(35,35)
      DOUBLE PRECISION TOTAL
    LOGICAL FLAG1
      CHARACTER*12 TABNAM
      EQUIVALENCE (TABNAM(10:11),WEEKC(1:2))
      DATA CDI/'N ','NE','E ','SE','S ','SW','W ','NW'/
C      DATA DEGS/0.,45.,90.,135.,180.,225.,270.,315./
      DATA DEGS/180.,225.,270.,315.,0.,45.,90.,135./


C GET TABLE, NOTE THAT NORTHERNMOST PORTION OF TABLE IS AT K=1
C NOTE THAT K IS E-W DIRECTION, I IS N-S DIRECTION

   write(0,4000)
4000        format(1x,'opening and reading ariavg.a')

      OPEN (UNIT=1,STATUS='OLD',FILE='ARIAVG.A')
      DO 10 I=1,21
       READ(1,100)(TAB(I,K),K=1,11)
100    FORMAT(1X,11D8.0)
10     CONTINUE
      CLOSE(1)

C COMMENT OUT FOLLOWING OPERATION FOR NOW
C SET ORIGIN =0

C           TAB(21,6)=0.

C OPEN WEATHER FILE

   write(0,4005)
4005        format(1x,'opening and reading test.out ')

      OPEN(UNIT=1,STATUS='OLD',FILE='TEST.OUT')

C BEGIN READ AND PROCESSING LOOP, EACH 7 DAYS RESULTS IN NEW
```

```
C CREATE NAME OF FORM AIC2ARIA.01 WWWWMMMS.KK
C WWWW LAST 4 CHARACTERS OF WEATHER STATION NAME
C MMM  METHOD IN TELONE MODEL USED TO SUMMARIZE SECTION FACTORS C
C  (ARI=ARITHMETIC, GEO=GEOMETRIC, MAX=MAXIMUM)
C S    STABILITY CLASS (A-F)
C KK   WEEK OF INTEREST (01,02,...52)

      TABNAM(1:4)='TEST'
      TABNAM(5:7)='ARI'
      TABNAM(8:9)='A.'
    TABNAM(12:12)=' '

C FIRST ZERO OUT SPACE AND XPAND

      DO 35 I=1,51
      DO 35 J=1,51
      XPAND(I,J)=0.
35     SPACE(I,J)=0.

C START MAIN LOOP BELOW
C..............................................................
      COUNT=0
1     CONTINUE
   FLAG1=.TRUE.
      READ(1,200,END=600)DIRECT,WSPEED
200   FORMAT(A2,F2.0)
   write(0,4010)direct
4010        format(1x,'direct= ',a2)
   FLAG1=.FALSE.
      COUNT=COUNT+1
      IF(COUNT.GT.365)GOTO1000

C CONVERT DIRECTION TO ANGLE IN RADIANS

      DO 20 J=1,8
       IF (DIRECT.EQ.CDI(J))THEN
        ANGLE=2.*3.1416*DEGS(J)/360.
        GOTO 21
       ENDIF
20     CONTINUE

C UH OH, DIDN'T MATCH, BETTER ISSUE WARNING

      WRITE(0,5001)DIRECT
5001  FORMAT(1X,'WKDSP ERROR: FAILED TO MATCH DIRECTION= ',A2)
      PAUSE

21     CONTINUE

C  COMPUTE FACTOR TO EXPAND TABLE BY, ONLY USE REAL INTEGER FROM 1 TO 5
C  DON'T USE ANYTHING ELSE LIKE FRACTIONS, OR HIGHER OR LOWER
C  AND NOTE THAT 2 MPH IS APPROXIMATELY 1 M/S

   XFACT=FLOAT(INT(WSPEED/2.))
   XFACT=MIN(XFACT,5.)
   XFACT=MAX(XFACT,1.)

C AND XFACT SQUARED
```

```

  XF2=XFACT*XFACT

C NOW EXPAND FACTOR TABLE INTO XPAND

  CALL XP(XFACT,XPAND,51,51,TAB,21,11)

C NOW USE DIRECTION TO START ROTATION OF FIRST DAY OF WEEK

      DO 40 I=1,51
      DO 40 K=1,51

C MAKE CENTER OF TABLE, BOTTOM ROW, THE ORIGIN

       TI=26-I
       TK=K-26

C NOW ROTATE

       X=FLOAT(TI)
       Y=FLOAT(TK)
C          write(0,4020)
C4020     format(1x,'calling rote ')
       CALL ROTE(ANGLE,X,Y,1)

C NOW TRANSLATE BACK TO ORIGINAL SQUARE COORDINATES

       IX=26-NINT(X)
       IY=NINT(Y)+26

C NOW TRANSLATE INTO SPACE COORDINATES, IE MOVE THE SMALL SQUARE
C SO THAT SMALL SQUARE WILL IDENTIFY THE POINT (1,6) (NS,EW) WITH THE
C BIG SQUARE POINT (26,26), X IS DOWNWIND DISTANCE NS, Y IS CROSSWIND
C DISTANCE EW
C
C        IX=IX+5
C        IY=IY+20
C
C BOUNDS CHECK ON INDICES, TO AVOID ACCIDENTAL SYSTEM CRASH

       IF(IX.LT.1.OR.IX.GT.51.OR.IY.LT.1.OR.IY.GT.51)THEN
C         WRITE(0,5005)IX,IY,COUNT
C5005    FORMAT(1X,'INDICES OUT OF BOUNDS IX,IY,COUNT ',3I4)
C         STOP
   CONTINUE
       ELSE

C BOUNDS OK, NOW ADD TO VALUE IN SPACE

        SPACE(IX,IY)=XPAND(I,K)/XF2+SPACE(IX,IY)

       ENDIF

40    CONTINUE

C ROTATION AND ADDITION TO SPACE IS NOW DONE
C CHECK TO FIND OUT IF WE ARE AT END OF A WEEK

```

```
600    WEEK=7
       IF (MOD(COUNT,WEEK).EQ.0.OR.FLAG1)THEN

C TAKE CARE OF SITUATION WHERE LAST LINE IS MULTIPLE OF 7

  IF(MOD(COUNT,WEEK).EQ.0.AND.FLAG1)GOTO1000

C TRANSFER CENTER 35X35 PORTION OF MATRIX FROM 51X51 MATRIX
C TO FINAL

  DO 670 I=1,35
  DO 670 J=1,35
670          FINAL(I,J)=SPACE(8+I,8+J)
C NEED TO SMOOTH IT OUT
  write(0,4030)
4030         format(1x,'calling smooth')
C       CALL SMOOTH(SPACE,51,51)
  CALL SMOOTH(FINAL,35,35)

C RESET ORIGIN TO 0

C            SPACE(26,26)=0.
  FINAL(18,18)=0.

C NORMALIZE POSITIVE VALUES IN SPACE
  write(0,4050)
4050         format(1x,'calling avgnor')
C       CALL AVGNOR(SPACE,51,51)
  CALL AVGNOR(FINAL,35,35)

C NOW PRINT IT OUT

C FIRST GET WEEK NUMBER CODED INTO FILENAME

       WEEKID=COUNT/7
       IF(WEEKID.LT.10)THEN
        WRITE(WEEKC,201)WEEKID
201      FORMAT(I2)
        WEEKC(1:1)='0'
       ELSE
        WRITE(WEEKC,201)WEEKID
       ENDIF

C NOW OPEN FILE AND GET IT OUT

  write(0,3434)tabnam
3434         format(1x,'tabnam= ',a12)


      OPEN(UNIT=2,FILE=TABNAM,STATUS='NEW')

C WRITE IT OUT WITH NORTH AT THE TOP

       DO 300 I=1,35
C       WRITE(2,320)(SPACE(I,J),J=1,51)
C320     FORMAT(51E8.2)
   WRITE(2,320)(FINAL(I,J),J=1,35)
320          FORMAT(35E8.2)
```

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 4.10

```
300    CONTINUE

C CLOSE FILE
  CLOSE(2)

C ZERO OUT SPACE FOR NEXT WEEK

        DO 36 I=1,51
        DO 36 J=1,51
        XPAND(I,J)=0.
 36      SPACE(I,J)=0.

       ENDIF
       IF(.NOT.FLAG1)GOTO1
1000  STOP
       END
```

main Local Symbols

Name	Class	Type	Size	Offset
WSPEED	local	REAL*4	4	0002
COUNT	local	INTEGER*4	4	0006
I	local	INTEGER*4	4	000a
J	local	INTEGER*4	4	000e
K	local	INTEGER*4	4	0012
FLAG1	local	LOGICAL*4	4	0016
XF2	local	REAL*4	4	001a
CDI	local	CHAR*2	16	001c
TAB	local	REAL*8	1848	001e
DEGS	local	REAL*4	32	002c
X	local	REAL*4	4	0756
Y	local	REAL*4	4	075a
TI	local	INTEGER*4	4	075e
TK	local	INTEGER*4	4	0762
IX	local	INTEGER*4	4	0766
IY	local	INTEGER*4	4	076a
ANGLE	local	REAL*4	4	076e
FINAL	local	REAL*8	9800	0772
WEEK	local	INTEGER*4	4	2dba
SPACE	local	REAL*8	20808	2dbe
TABNAM	local	CHAR*12	12	7f06
XFACT	local	REAL*4	4	7f12
WEEKID	local	INTEGER*4	4	7f16
XPAND	local	REAL*8	20808	7f1a
DIRECT	local	CHAR*2	2	d062
WEEKC	local	CHAR*2	2	7f0f

Global Symbols

Name	Class	Type	Size	Offset
AVGNOR	extern	***	***	***
ROTE	extern	***	***	***
SMOOTH	extern	***	***	***
XP	extern	***	***	***
main	FSUBRT	***	***	0000

Microsoft FORTRAN Optimizing Compiler Version 4.10

Code size = 06e2 (1762)
Data size = 011f (287)
Bss size = d064 (53348)

No errors detected

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 4.10

```
      SUBROUTINE AVGNOR(A,M,N)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C NORMALIZES POSITIVE VALUES
C A IS DOUBLE PRECISION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT INTEGER(A-Z)
      DOUBLE PRECISION A(M,N)
      DOUBLE PRECISION TOTAL
C
      TOTAL=0.
      COUNT=0
      DO 10 I=1,M
      DO 10 J=1,N
      IF(A(I,J).GT.0.)THEN
       TOTAL=TOTAL+A(I,J)
       COUNT=COUNT+1
      ENDIF
10    CONTINUE

C CHECK TOTAL

      IF(TOTAL.LE.0.)THEN
       WRITE(0,100)TOTAL
100    FORMAT(1X,'AVGNOR BAD TOTAL: = ',E20.10)
       PAUSE
      ENDIF

C TOTAL OK, PROCEED

      DO 20 I=1,M
      DO 20 J=1,N
20    A(I,J)=A(I,J)/TOTAL

      RETURN
      END
```

AVGNOR Local Symbols

Name	Class	Type	Size	Offset
N	param			0006
M	param			000a
A	param			000e
__V18	param			fffa
__V19	param			fffc
__V20	param			fffe
TOTAL	local	REAL*8	8	0002
I	local	INTEGER*4	4	000a
COUNT	local	INTEGER*4	4	000e
J	local	INTEGER*4	4	0012

Microsoft FORTRAN Optimizing Compiler Version 4.10

Global Symbols

Name	Class	Type	Size	Offset
AVGNOR.	FSUBRT	***	***	0000

Code size = 0297 (663)
Data size = 0015 (21)
Bss size = 0016 (22)

No errors detected

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 4.10

```
      SUBROUTINE SMOOTH(A,M,N)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C TAKES LOCAL AVERAGE OF 9 SQUARES AS ESTIMATE
C FOR CENTER SQUARE.  NOTE A IS DOUBLE PRECISION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT INTEGER(A-Z)
      REAL*8 A(M,N)
      REAL RA(60,60),NEWA(60,60)

      DO 10 I=1,M
      DO 10 J=1,N
      NEWA(I,J)=0.
 10           RA(I,J)=A(I,J)

      DO 20 I=2,M-1
      DO 20 J=2,N-1
       DO 25 K=-1,1
       DO 25 L=-1,1
 25            NEWA(I,J)=NEWA(I,J)+RA(I+K,J+L)
       NEWA(I,J)=NEWA(I,J)/9.
 20           CONTINUE

      DO 40 I=2,M-1
      DO 40 J=2,N-1
 40           A(I,J)=NEWA(I,J)

      RETURN
      END
```

SMOOTH Local Symbols

Name	Class	Type	Size	Offset
N	param			0006
M	param			000a
A	param			000e
__V19	param			fffa
__V20	param			fffc
__V21	param			fffe
I	local	INTEGER*4	4	0000
J	local	INTEGER*4	4	0004
K	local	INTEGER*4	4	0008
L	local	INTEGER*4	4	000c
RA.	local	REAL*4	14400	0010
NEWA.	local	REAL*4	14400	3850

Global Symbols

Name	Class	Type	Size	Offset
SMOOTH.	FSUBRT	***	***	0000

Code size = 041d (1053)
Data size = 0004 (4)
Bss size = 7090 (28816)

Microsoft FORTRAN Optimizing Compiler Version 4.10

No errors detected

```
      SUBROUTINE XP(XFACT,XPAND,NS,EW,TAB,TNS,TEW)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C THIS SUBROUTINE EXPANDS FACTOR TABLE IN NS AND EW
C DIRECTIONS ACCORDING TO FACTOR XFACT, IT ASSUMES
C THAT THE ORIGIN OF TAB IS (21,6) (CENTER OF BOTTOM
C ROW) AND THE ORIGIN OF XPAND IS (26,26). THESE 2
C ORIGINS ARE IDENTIFIED TOGETHER
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT INTEGER(A-Z)
      DOUBLE PRECISION XPAND(NS,EW),TAB(TNS,TEW)
      REAL XFACT
C
      DO 10 I=1,51
      DO 10 J=1,51
10          XPAND(I,J)=0.

      DO 20 I=1,51

C GET EAST WEST COORDINATES STRAIGHTENED OUT

       EB1=I
       EB2=I-26
       EA2=EB2/XFACT
       EA1=EA2+6

C DON'T GO OUT OF BOUNDS

       IF(.NOT.(EA1.LT.1.OR.EA1.GT.11))THEN

C GET NS COORDINATES STRAIGHTENED OUT

        DO 30 J=1,51
         NB1=J
         NB2=27-NB1
         NA2=1+(NB2-1)/XFACT
         NA1=22-NA2

C DON'T GO OUT OF BOUNDS

         IF(.NOT.(NA1.LT.1.OR.NA1.GT.21.OR.NB2.LT.1.OR.NB2.GT.26))
     1         THEN

C AND NOW, FOR THE MOMENT YOU'VE BEEN WAITING FOR

           XPAND(J,I)=TAB(NA1,EA1)
          ENDIF
30          CONTINUE
         ENDIF
20          CONTINUE
      RETURN
      END
```

Microsoft FORTRAN Optimizing Compiler Version 4.10

XP Local Symbols

Name	Class	Type	Size	Offset
TEW	param			0006
TNS	param			000a
TAB	param			000e
EW.	param			0012
NS.	param			0016
XPAND	param			001a
XFACT	param			001e
__V21	param			fff4
__V22	param			fff6
__V23	param			fff8
__V24	param			fffa
__V25	param			fffc
__V26	param			fffe
NA1	local	INTEGER*4	4	0000
NA2	local	INTEGER*4	4	0004
NB1	local	INTEGER*4	4	0008
NB2	local	INTEGER*4	4	000c
I	local	INTEGER*4	4	0010
J	local	INTEGER*4	4	0014
EA1	local	INTEGER*4	4	0018
EA2	local	INTEGER*4	4	001c
EB1	local	INTEGER*4	4	0020
EB2	local	INTEGER*4	4	0024

Global Symbols

Name	Class	Type	Size	Offset
XP.	FSUBRT	***	***	0000

Code size = 028f (655)
Data size = 0004 (4)
Bss size = 0028 (40)

No errors detected

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 4.10

```
      SUBROUTINE ROTE(ANGLE,X,Y,N)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ROTATES SET OF POINTS IN X,Y OF DIMENSION N BY ANGLE
C CONTENTST OF X,Y, DESTROYED, REPLACED BY ROTATED POINTS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT INTEGER(A-Z)
      REAL X(N),Y(N),ANGLE,TX,TY

      DO 10 I=1,N
       TX=X(I)*COS(ANGLE)-Y(I)*SIN(ANGLE)
       TY=X(I)*SIN(ANGLE)+Y(I)*COS(ANGLE)
       X(I)=TX
       Y(I)=TY
10            CONTINUE
      RETURN
      END
```

ROTE Local Symbols

Name	Class	Type	Size	Offset
N	param			0006
Y	param			000a
X	param			000e
ANGLE	param			0012
__V20	param			fffc
__V21	param			fffe
I	local	INTEGER*4	4	0000
TX.	local	REAL*4	4	0004
TY.	local	REAL*4	4	0008

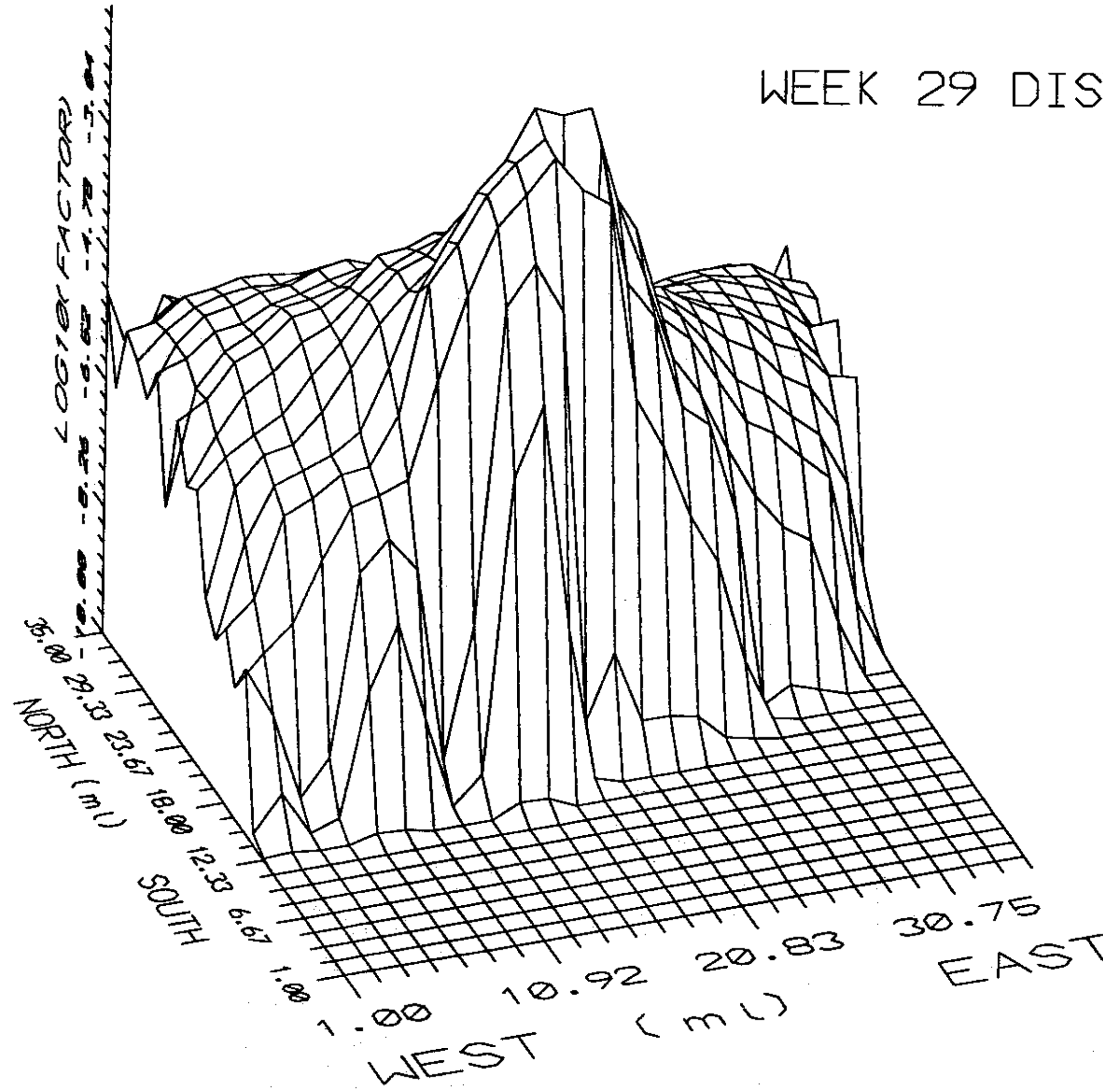
Global Symbols

Name	Class	Type	Size	Offset
ROTE.	FSUBRT	***	***	0000

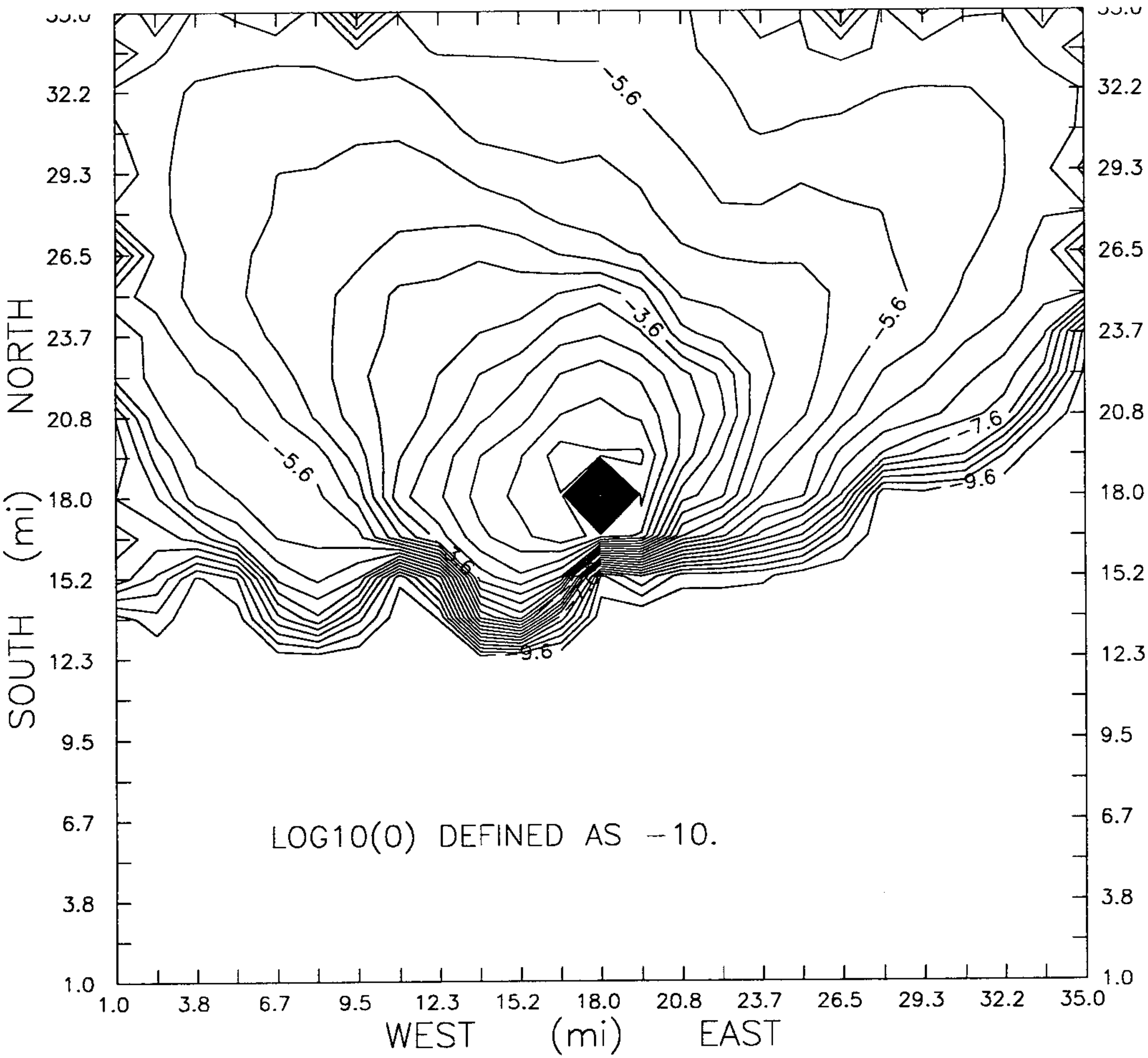
Code size = 0145 (325)
Data size = 0000 (0)
Bss size = 000c (12)

No errors detected

WEEK 29 DISPERSION FACTORS

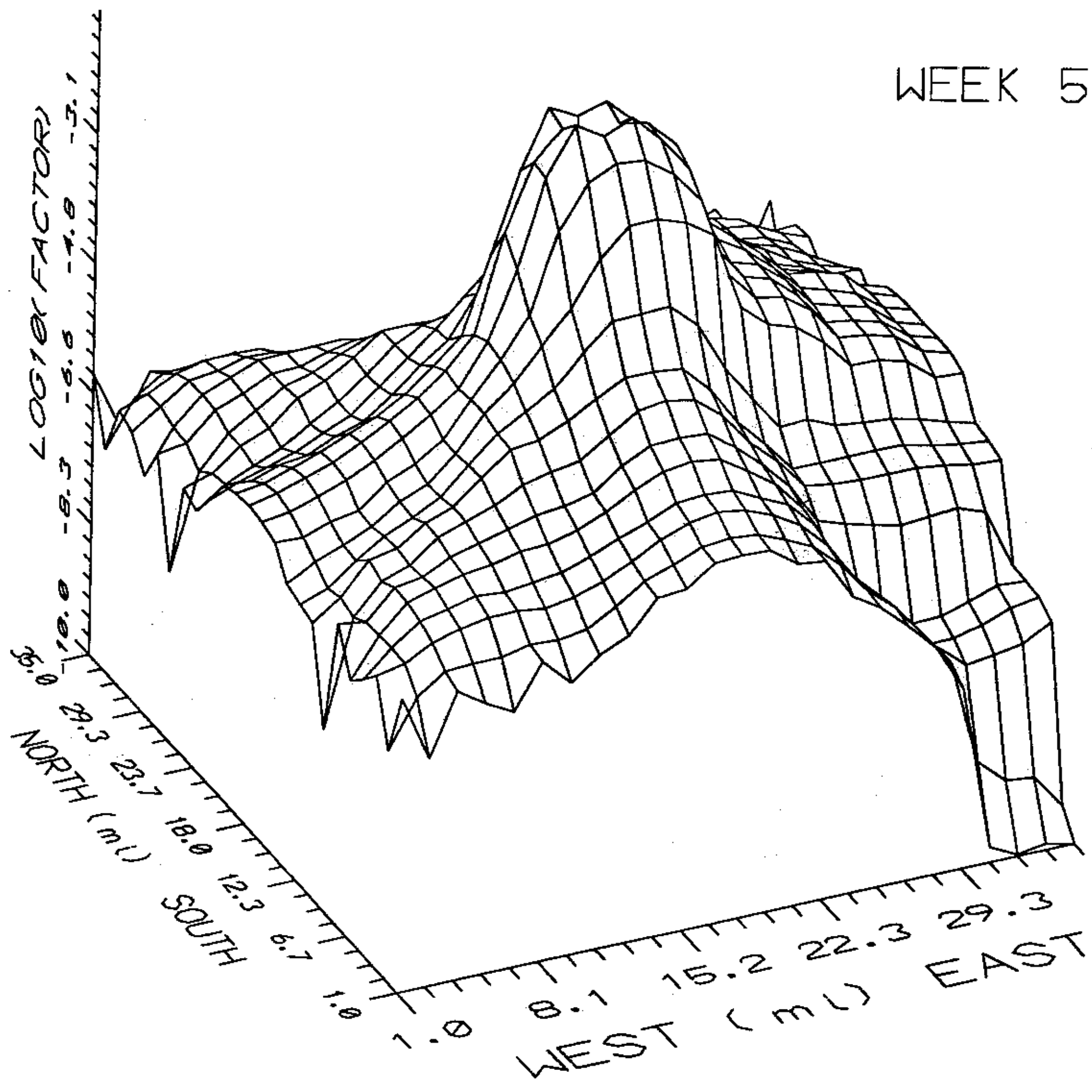


WIND DATA		
DAY	DIR	SPD
197	SE	5
198	S	4
199	SW	4
200	SE	7
201	SE	12
202	SE	6
203	SE	5



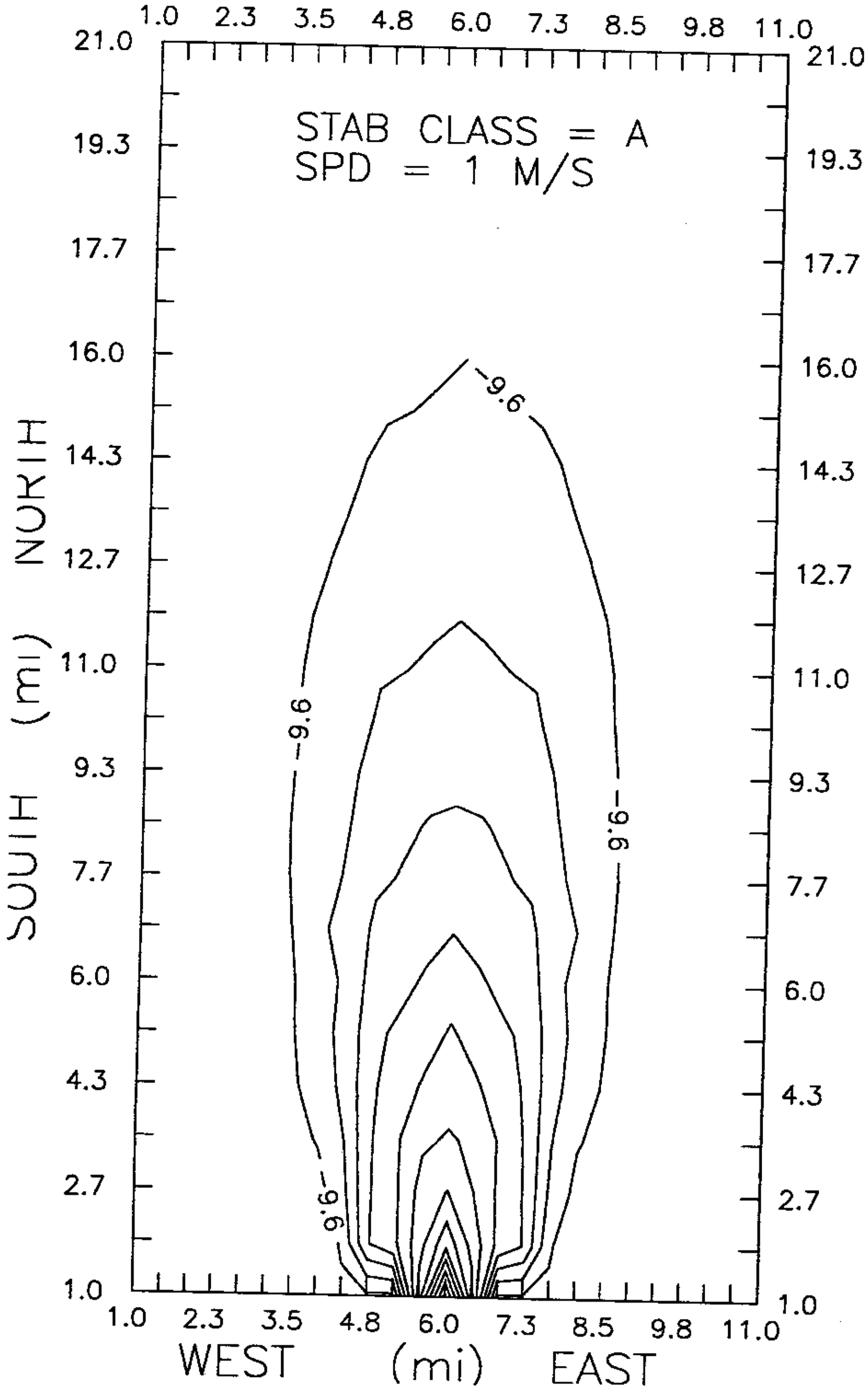
-5.6
-3.6
-7.6
-9.6
LOG10(0) DEFINED AS -10.
NORTH (mi) SOUTH
WEST (mi) EAST
1.0
3.8
6.7
9.5
12.3
15.2
18.0
20.8
23.7
26.5
29.3
32.2
35.0

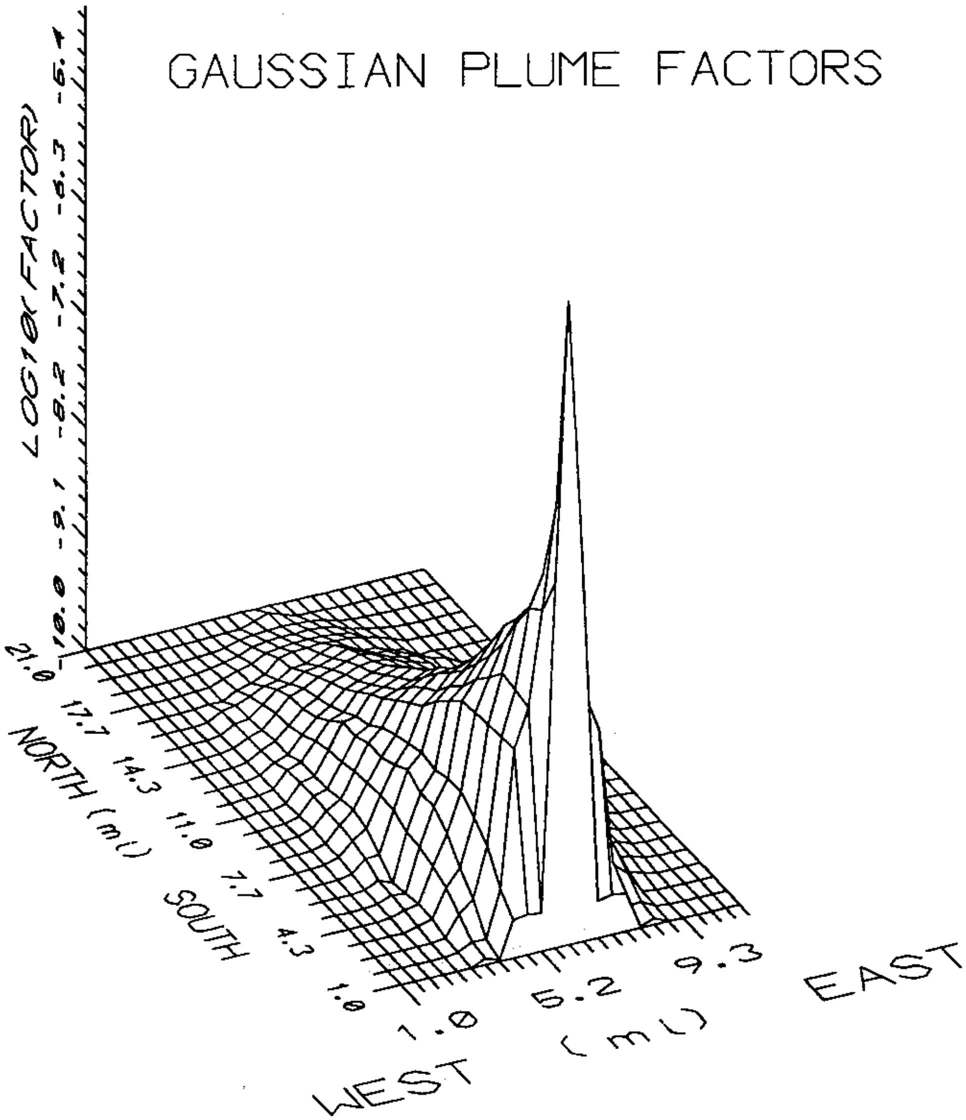
WEEK 51 DISPERSION FACTORS

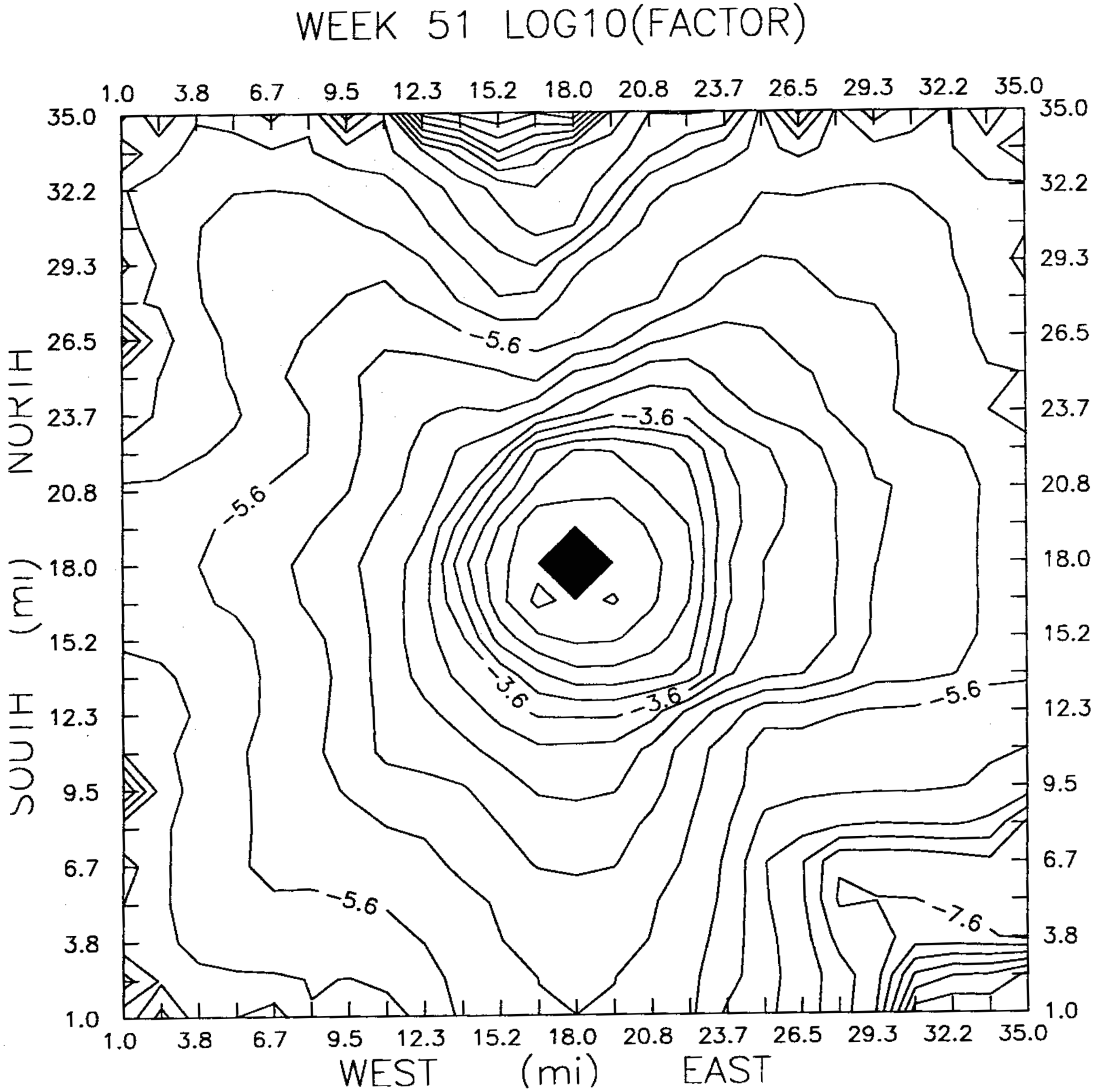


WIND DATA

DAY	DIR	SPD
351	E	4
352	SE	4
353	SW	9
354	W	13
355	N	7
356	N	6
357	NE	7


STAB CLASS = A
SPD = 1 M/S
-9.6
SOUTH (mi) NORTH
WEST (mi) EAST
1.0 2.3 3.5 4.8 6.0 7.3 8.5 9.8 11.0
21.0 19.3 17.7 16.0 14.3 12.7 11.0 9.3 7.7 6.0 4.3 2.7 1.0


GAUSSIAN PLUME FACTORS
LOG10(FACTOR)
-10.0 -9.1 -8.2 -7.2 -6.3 -5.4
21.0 17.7 14.3 11.0 7.7 4.3 1.0
NORTH (mi) SOUTH
1.0 5.2 9.3
WEST (mi) EAST



WEEK 51 LOG10(FACTOR)
1.0 3.8 6.7 9.5 12.3 15.2 18.0 20.8 23.7 26.5 29.3 32.2 35.0
35.0 32.2 29.3 26.5 23.7 20.8 18.0 15.2 12.3 9.5 6.7 3.8 1.0
-5.6
-3.6
-7.6
SOUTH (mi) NORTH
WEST (mi) EAST